Trabajo Fin de Máster
Máster Universitario en Ingeniería
de Telecomunicación

Detección de anomalías en sistemas de Smartlighting

Autor: Agustín Walabonso Lara Romero
Tutor: Rafael María Estepa Alonso

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Máster
Máster Universitario en Ingeniería de Telecomunicación

Detección de anomalías en sistemas de Smartlighting

Autor:
Agustín Walabonso Lara Romero

Tutor:
Rafael María Estepa Alonso
Profesor titular

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2023

Trabajo Fin de Máster: Detección de anomalías en sistemas de Smartlighting

Autor: Agustín Walabonso Lara Romero
Tutor: Rafael María Estepa Alonso

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

El Secretario del Tribunal

A mi familia
A mis amigos
A mis profesores
Y sobre todo a mis padres

Agradecimientos

Gracias a mis padres por educarme y permitir formarme en algo que quería ser desde pequeño, un ingeniero que pudiera entender y dar solución a la mayoría de las cosas que nos rodean en la actualidad. Ellos siempre me apoyan en todo sin tener la más mínima duda.

También gracias a toda mi familia y amigos que siempre están apoyando mis decisiones.

Por último y no menos importante, darle las gracias a mi mentor y gran profesor Rafael Estepa, que es mi tutor y también compañero de trabajo. Gracias a él, he aprendido y sigo aprendiendo mucho sobre la ciberseguridad.

Agustín Walabonso Lara Romero
Sevilla, 2023

Resumen

Cada vez es más común la monitorización y el control de todas las cosas en la vida, esto se conoce como IoT (Internet Of Things) o Internet de las cosas en español. Esta tendencia también se ve reflejada en los procesos industriales, lo que se conoce como la industria 4.0. Dentro de este ámbito nos podemos encontrar con la gestión de luminarias inteligentes (Smart-Lighting). El problema es que estos sistemas IoT presentan vulnerabilidades frente diferentes amenazas cibernéticas. Es por ello, por lo que se crea la necesidad de diseñar e implementar sistemas de ciberprotección específicos para arquitecturas Smart-Lighting.

En este trabajo se hace primero un estudio sobre las arquitecturas Smart-Lighting, analizando sus propiedades y las comunicaciones IoT internas. También se realiza un estudio sobre las posibles amenazas IoT existentes. Para dar una solución, en este trabajo se presenta un novedoso sistema de detección modular específico para el caso de uso Smart-Lighting, basado en la detección de anomalías tanto en el nivel de red, como en el nivel de aplicación. Se presentan los algoritmos internos del sistema, las pruebas realizas y por último una discusión sobre los resultados obtenidos.

Abstract

It is becoming increasingly common to monitor and control everything in life, known as IoT (Internet Of Things). This trend is also reflected in industrial processes, known as Industry 4.0. Within this field we can find the management of intelligent luminaires (Smart-Lighting). The problem is that these IoT systems are vulnerable to various cyber threats. This is why it is necessary to design and implement specific cyber-protection systems for Smart-Lighting architectures.

In this work, we first study Smart-Lighting architectures, analysing their properties and internal IoT communications. A study on the possible existing IoT threats is also carried out. To provide a solution, this paper presents a novel modular detection system specific to the Smart-Lighting use case, based on anomaly detection at both the network level and the application level. The internal algorithms of the system, the tests performed and finally a discussion of the results obtained are presented.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN Y OBJETIVOS

Para que buscar una victoria si puedes lograr un buen acuerdo
Agustín W. Lara Romero

El internet de las cosas (IoT) ha evolucionado muy rápido y cada vez es más común encontrar múltiples y diferentes tipos de dispositivos conectados a internet. El uso de IoT ha ido extendiéndose a todos los dominios como, por ejemplo, el uso en la industria 4.0 [1], casas inteligentes [2] y entornos sanitarios [3] entre otros muchos más. Del concepto industria 4.0, nace el mundo OT (Tecnología Operativa).
Otro ámbito de aplicación IoT es el de Smart City [4] donde se encuentra Smart Lighting [5].

Este trabajo se centra en la Ciberprotección dentro de un sistema de Smart Lighting o también conocido como luninarias inteligentes. A continuación, en la Figura 1 se describe un ejemplo de arquitectura Smart Lighting [5] [6].

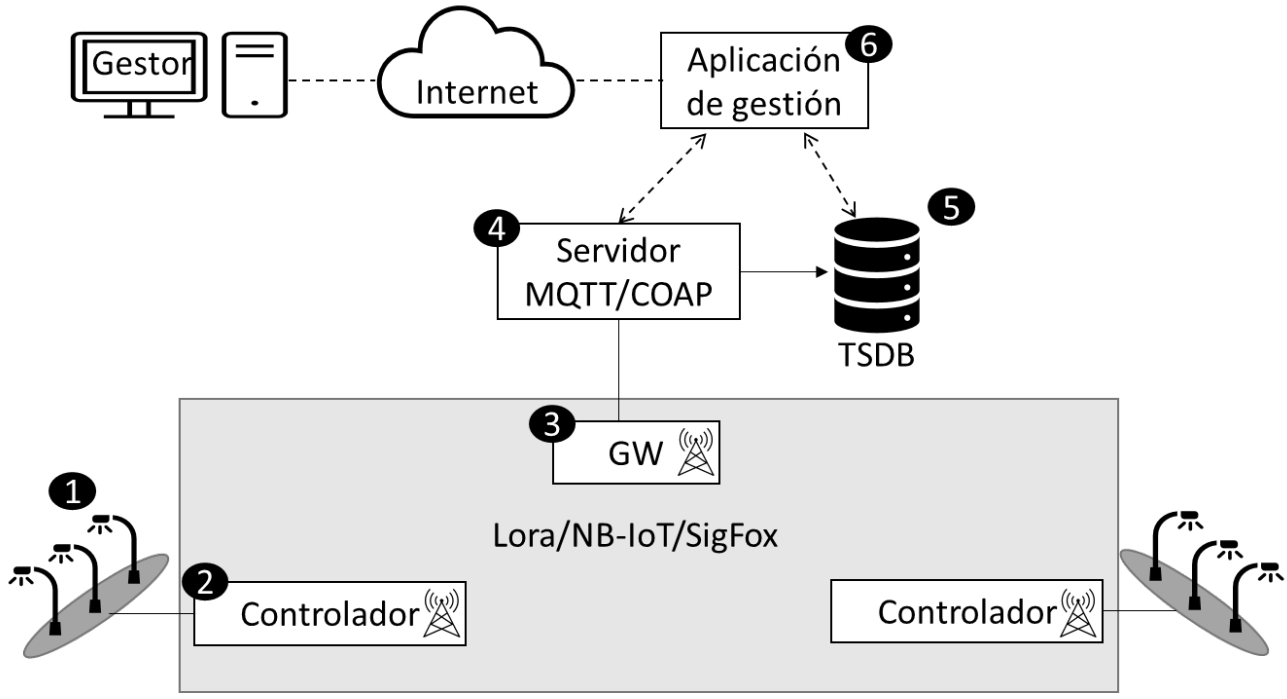

Figura 1 Arquitectura Smart-Lighting.

En la arquitectura presentada se pueden apreciar los siguientes elementos:

1. Las iluminarias inteligentes.
2. Controladores de las iluminarias.
3. Pasarela de comunicaciones.
4. Servidor MQTT o COAP.
5. Base de datos de series temporales.
6. Aplicación de gestión.

El controlador de iluminarias actúa sobre las iluminarias encendiéndola y apagándola en función a la salida y puesta del sol, o en función del nivel de luminosidad entre otros de los posibles factores. El controlador también envía datos de medidas sobre las iluminarias a un servidor a través de una pasarela. La transmisión de estos datos se puede realizar a través de diferentes redes de comunicación como pueden ser redes NB-IoT, SigFox o LoraWAn. Siendo esta última la más usada en este tipo de arquitecturas.
La red **LoraWan** (Long Range Area Network) es una red inalámbrica de baja potencia y área amplia. Esta red está diseñada especialmente para dispositivos con requisitos de bajo consumo y tienen que operar en redes de largo alcance [7]. Esta red es la más usada en las arquitecturas IoT debido a las características y requisitos mencionados. Esta red es una red de redes en estrella, en la cual, en la primera estrella se encuentran los dispositivos finales y las puertas de enlaces. En la segunda estrella, se encuentra las puertas de enlace y el servidor central de la red. Las velocidades de datos en este tipo de red se encuentran en el rango de 0.3kbps y 50 kbps.
Narrow Band IoT (NB-IoT) [8] fue introducida en las redes LTE como la principal tecnología M2M para despliegues a gran escala. Permite que un gran número de componentes, como actuadores y sensores puedan comunicarse a lo largo de un área extensa. Esto es posible gracias a que utiliza una banda estrecha de sólo 180 kHz para dar comunicación a todos los componentes. La principal ventaja del uso de este tipo de redes para aplicaciones IoT, es que también cumple con el requisito de eficiencia energética.
El servidor que usa como protocolos de aplicación COAP o MQTT, inserta los datos recibidos en una base de datos de series temporales. Además, el servidor también se encarga de enviar los comandos y operaciones a los controladores de las iluminarias. Finalmente, todo el sistema es monitorizado y configurado a través de una aplicación web de gestión, siendo configurada y monitorizada por un gestor.

Las tecnologías OT (Tecnologías de Operación) se usan para la monitorización y control de activos en procesos industriales. Dentro de estas tecnologías, se encuentran los siguientes elementos:

- **ICS** (sistema de control de industrial): monitoriza y controla los procesos industriales.
- **IACS** (sistema de control y automatización industrial): engloba el personal, software y hardware que puede llegar a afectar en la operación de un proceso industrial.
- **SCADA** (sistema de control y adquisición de datos): sistemas que obtienen datos y controlan las infraestructuras industriales.
- **HMI** (interfaz hombre-máquina): interfaz usada por los operadores para poder interactuar con los sistemas OT.
- **DCS** (sistema de control distribuido): sistema de control centralizado.
- **RTU** (unidad de terminal remota): recogen datos de los dispositivos físicos y convierten las señales analógicas en digitales para transmitiros al centro de control.
- **PLCs** (controlador lógico programable): dispositivo industrial diseñado para su uso en tiempo en real en entornos donde el control no necesita necesariamente comunicación con SCADAs o DCSs.

Para arquitecturas OT existe un grado de automatización descrito en el modelo de ISA 95 [9]. En este acuerdo se describe una pirámide de automatización dividida en 5 niveles. En cada nivel se indican los requisitos de periodicidad en el intercambio de información.

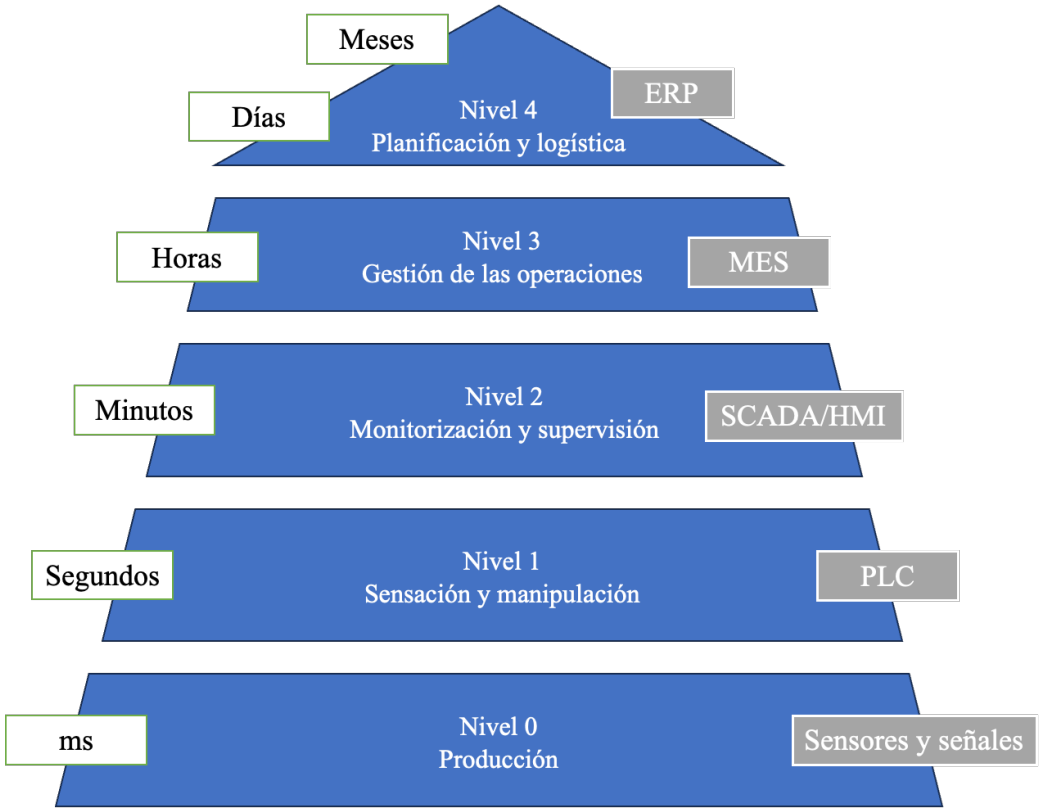

Figura 2 Pirámide de automatización según modelo ISA 95.

1.1 Amenazas y ciberprotección en Smart-Lighting

Las arquitecturas Smart-Lighting son vulnerables a múltiples amenazas como, por ejemplo:

- Amenazas OT: robo de energía, vandalismo, averías, fallos hardware.
- Amenazas IT: fallos software, ciberataques.

Debido a la existencia de múltiples amenazas, existen diferentes sistemas de protección IT como pueden ser: firewall, antivirus, network-ids, cifrado de conexiones, … etc [10]. Estos no cubren todas las necesidades, debido a que detectan ataques conocidos. Pero existen múltiples ataques no conocidos como pueden ser los ataques zero days [11] o malicious insider [12] que son difíciles de detectar por los sistemas tradicionales de protección IT. En un artículo reciente [13], se presentan diferentes amenazas categorizadas en cuatro capas:

- **Capa física:** esta capa contiene una variedad de dispositivos, como actuadores y sensores, que recopilan datos y los envían a la capa superior de la arquitectura. Es por ello por lo que los ataques en esta capa están dirigidos al hardware y podemos encontrarnos con las siguientes amenazas:
  - Sleep deprivation attack: mecanismo o inyección de código malicioso que hace que los dispositivos no entren en estado de reposo, provocando un consumo excesivo de batería, consiguiendo reducir la vida del dispositivo.
  - Capturing & Fake node injection: suplantación o reemplazo de un nodo en la red IoT, permitiendo un acceso a la red.
  - Malicious code injection attack: inyección de código en la memoria de un nodo del sistema IoT, permitiendo realizar funciones no deseadas o ganar acceso a la red.

  - Eavesdropping attack: el atacante captura los datos cuando estos son transmitidos o en la fase de autenticación.
- **Capa de red:** esta capa engloba la red de comunicación, internet y redes de sensores inalámbricos. La función principal es la de recoger los datos obtenidos por la capa física y enviarlos a capas superiores para el procesamiento de los datos. A continuación, se detallan diferentes ataques que podemos encontrar en esta capa:
  - DDOS attack: este ataque tiene el objetivo de hacer que un servicio o recurso sea inaccesible por los usuarios, provocados por ejemplo con una inhumación de peticiones múltiples dirigidas al servicio en cuestión.
  - Routing attack: este ataque consigue redirigir las rutas de transmisión de los datos, consiguiendo crear bucles o pérdidas en los datos transmitidos.
  - Sniffing attack: el atacante consigue leer los paquetes que cursan una red. Es muy común encontrarlos en redes inalámbricas donde el medio de transmisión es compartido por todos los dispositivos.
  - Traffic analysis attacks: el atacante consigue obtener de forma pasiva o activa el tráfico que cursa la red y obtiene información sensible de los usuarios. También válido cuando el atacante extrae información a través de escaneos.
  - Credential Access: el atacante consigue el acceso a un dispositivo a través de un ataque de fuerza de bruta de credenciales.

- **Capa de plataforma:** es una capa intermedia ubicada entre las capas de red y aplicación. Esta capa sirve la función de almacenamiento de los datos, cuando estos son transmitidos entre los dispositivos IoT. Esta capa presenta vulnerabilidades a una gran variedad de ataques que pueden conseguir infectar el control de la aplicación IoT.
  - Cloud malware injection: el atacante consigue introducir un Malware [1] en la cloud a través de la instalación de una máquina virtual o la instancia del módulo de un servicio malicioso.
  - SQL injection: inyección de código a base de datos SQL. El atacante consigue acceso a datos protegidos.
  - Storage attacks: los entornos IoT manejan grandes cantidades de datos que son almacenados en nubes o dispositivos, los cuales pueden ser atacados.

- **Capa de aplicación:** capa superior de la arquitectura que ofrece el servicio expuesto a los usuarios. Los ataques en esta capa dependen de la tecnología que se encuentra detrás de la aplicación. A continuación, se detalla algunos de los ataques posibles que podemos encontrar en esta capa:
  - Reprogram attack: un atacante consigue tomar el control de la red a través de la reprogramación remota de los dispositivos IoT.
  - Sniffing attack: el atacante consigue realizar lecturas de los paquetes de la capa de aplicación.
  - Service interruption attacks: ataques de disrupción en la operación del servicio.
  - False Data Injection: ataque dirigido a la manipulación de los datos enviados por los dispositivos IoT.
  - False Command Injection: ataque dirigido a la manipulación de los comandos enviados por los dispositivos IoT.

A modo de resumen, se puede observar en la Figura 3 [13] un listado de posibles amenazas IoT categorizadas por capas y sus posibles contramedidas.

[1] Software malicioso que consigue infiltrarse en un dispositivo y causa interrupciones en un servicio o robo de datos sin ser detectado.

Application layer	**Malicious scripts** **Service interruption** **Data thefts** **Sniffing attack** **Reprogram attack**	**Data security** **Anti-viruses** **Anti-spyware** **Firwalls & ACLs**
Platform layer	**Man in the middle** **Cloud malware injection** **Side-channel attacks** **Storage attacks** **SQL injection**	**Hyper safe** **Homomophic encryption** **Web application scanners** **Fragmentation** **Redundancy scaterring**
Network layer	**Wormhole attack** **Routing attack** **Trafic attack** **DDOS attack** **Jamming attack**	**Routing protocol** **Routing security** **Data privacy** **Authentication** **Hello flood detection**
Physical layer	**Malicious code injection** **Faake node injection** **Sleep Deprivation** **Eavesdropping** **Jamming attack** **Social Engineering**	**Data privacy** **Secure booting** **Data integrity** **Risk assessment** **Device authentication** **Secure physical desing**

Figura 3 Posibles amenazas y contramedidas IoT.

Como hemos visto, los entornos IoT presentan numerosas amenazas y es por ello por lo que nace la necesidad de diseñar sistemas de protección basados en la detección de anomalías, que permitan encontrar patrones de comportamiento anómalos.

1.2 Objetivo del Proyecto

Habiendo identificado las necesidades de ciberprotección en las arquitecturas IoT, el objetivo principal del trabajo propuesto trata sobre el diseño y desarrollo de un sistema de protección para Smart-Lighting basado en la detección de anomalías, buscando patrones de comportamiento anómalos. Que cumpla los requisitos de ser completamente pasivo en la red IoT y tenga un coste computacional bajo. Para ello, se busca diseñar un sistema que permita detectar anomalías tanto en el nivel de red basado en flujos IPFIX, como en el nivel de datos de aplicación haciendo uso de técnicas de análisis de datos. Una vez diseñado el sistema, se implementará y se definirán mecanismos o métricas permitan medir el rendimiento de este. Por último, se analizarán los resultados para extraer las posibles limitaciones y mejoras que presenta el sistema propuesto.

2 REVISIÓN DE LA TECNOLOGÍA Y REVISIÓN DEL CONOCIMIENTO

Cada día sabemos más y entendemos menos
Albert Einstein

En este capítulo se pretende realizar una revisión del estado del arte de los protocolos y sistemas de protección usados en las arquitecturas Smart-Lighting.

2.1 Modelo de capas

En el mundo IoT no existe un modelo de capa que defina la arquitectura aceptada universalmente. Es por ello por lo que existen diferentes arquitecturas posibles dependiendo del caso de uso particular [14], [15]. A continuación, se van a detallar el modelo 5 capas que es el usado comúnmente en la mayoría de las aplicaciones IoT.

2.1.1 Modelo de 5 capas

En la Figura 4, se presenta un modelo de 5 capas.

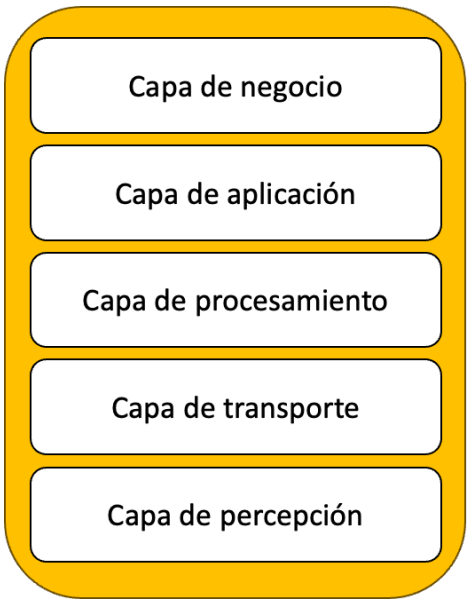

Figura 4 Modelo de 5 capas.

- **La capa de percepción** está orientada a todos los dispositivos hardware como sensores y actuadores encargados de obtener y procesar los datos que cursan una red IoT. Si se compara esta capa con el modelo tradicional de capas OSI, la capa de procesamiento engloba las capas físicas y de enlace del modelo OSI. Si se analiza el riesgo que conlleva esta capa, nos podemos encontrar con la necesidad de seguridad en los dispositivos y los protocolos de transmisión de datos inalámbricos.

- **La capa de transporte** incluye la transmisión de datos en una red IoT desde dispositivos finales hasta las aplicaciones que procesan y consumen los datos IoT. El alcance es tan extenso que hace que la seguridad en esta capa sea muy compleja y variada. Existen tantos dispositivos y tecnología distintas en esta capa, que, desde el punto de vista de ciberseguridad, se definen soluciones simples que en conjunto aporten una solución completa a la red. Las principales amenazas que se plantean en esta capa son: Sniffing, Man-in-the-middle, denegación de servicio, escaneos, entre otros posibles ataques. Todas estas posibles amenazas se pueden mitigar a través de análisis de anomalías, haciendo uso de técnicas ML basadas en la creación de modelos comportamientos, permitiendo detectar desviaciones en el comportamiento de la red.

- **La capa de procesamiento** trasporta, almacena y analiza los datos de los dispositivos finales y la aplicación IoT. En esta capa se pueden encontrar diferentes tecnologías como el procesamiento de eventos complejos, computación en la nube y la optimización de bases de datos no relacionales.

- **La capa de aplicación** incluye la solución final esperada por los usuarios, así como la operación y negocio del servicio IoT que se ofrece. En esta capa es muy importante asegurar la autentcidad, confidencialidad e integridad de los datos, debido a que esta capa es muy sensible a ataques dirigidos al acceso de datos no permitidos.

- **La capa de negocio** está orientada a la gestión completa del sistema IoT. Esto incluye las aplicaciones, el modelo de negocio y la privacidad de los usuarios.

Si se analiza el modelo de 5 capas, es una arquitectura completa que define la lógica del modelo IoT, donde los dispositivos físicos obtienen datos y estos son transmitidos hacia otros dispositivos o elementos de la red IoT, para que finalmente sean procesados por aplicaciones finales que ofrecen un servicio concreto. El modelo de 5 capas aporta un alto nivel de detalle a la hora de desarrollar aplicaciones reales, definiendo una arquitectura completa.

2.2 Comunicaciones IoT

Los protocolos y comunicaciones IoT deben cumplir con los siguientes requisitos [14]:
- Gran cantidad de dispositivos conectados a internet.
- Bajo consumo de energía.
- Alta fiabilidad.

En este apartado se presentan las características de las comunicaciones que suelen encontrarse en las arquitecturas IoT. En la Figura 5 se muestra una comparativa de las tecnologías existentes de comunicación inalámbrica [16], teniendo en cuenta el ancho de banda y el alcance de transmisión. Estos dos factores son claves a la hora de diseñar soluciones IoT.

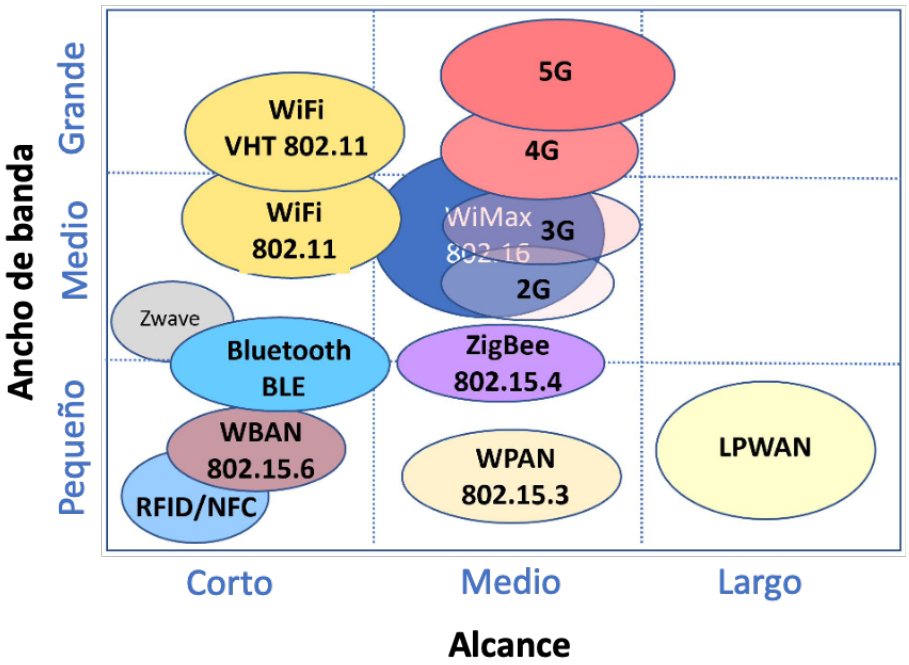

Figura 5 Comparación de tecnologías de comunicación inalámbrica.

LoRaWAN es un protocolo ubicado en la capa de control de acceso al medio (MAC) sobre una modulación LoRa (Long Range). Una característica de LoRaWAN es que todos los mensajes transmitidos por un nodo son recibidos por todos los gateways que se encuentren dentro del alcance de la comunicación. En la Tabla 1se detallan las características principales de LoRaWAN.
Las principales aplicaciones en las que podemos encontrarnos con esta tecnología son:

- Monitorización en tiempo real de redes de energía, gas y agua.
- Aplicaciones IoT en agricultura.
- Gestión y seguimiento de logística.
- Monitorización de sensores en edificios inteligentes.

Características principales de LoRaWAN	
Ancho de Banda	125, 250 o 500 kHz
Modulación	CSS
Máxima tasa de datos	50 Kbps
Tamaño máximo de payload	243 bytes
Alcance	5 km (urbano), 20 km (rural)
Cifrado	AES-128
Referencia	[17]

Tabla 1 Características principales de LoRaWAN

Sigfox es un protocolo de comunicación de banda UNB (Ultra Narrow-Band) que ofrece conectividad de extremo a extremo en dispositivos IoT. Se encuentra presente en la mayoría de las aplicaciones IoT que tengan una baja tasa de datos y cobertura del operador.
Es muy común encontrar esta tecnología en aplicaciones que requieran el rastreo de activos, como por ejemplo en la monitorización en tiempo real de instalaciones temporales debido a una obra, comúnmente conocido como mobiliario inteligente. En la Tabla 2 se presentan las características principales de esta tecnología.

Características principales de Sigfox	
Ancho de Banda	100 Hz
Modulación	BPSK
Máxima tasa de datos	100 bps
Tamaño máximo de payload	12 bytes (subida), 8 bytes (bajada)
Alcance	10 km (urbano), 40 km (rural)
Cifrado	No
Referencia	[18]

Tabla 2 Características principales de Sigfox.

NB-IoT es una tecnología de comunicación celular de banda estrecha. Se basa en las especificaciones de LTE, pero reduciendo las funcionalidades al mínimo e introduciendo mejoras específicas para entornos IoT. La idea principal de esta tecnología es la de optimizar los recursos y energía consumida por los dispositivos, permitiendo tener baterías más pequeñas en los dispositivos IoT. En la Tabla 3 se presentan las características principales de esta tecnología. Podemos encontrar esta tecnología en las diferentes aplicaciones IoT:

- Monitorización en tiempo real de redes de agua, gas y energía.
- Monitorización de maquinaria industrial para un mantenimiento predictivo.
- Monitorización de edificios inteligentes (temperatura, humedad, …, etc).

Características principales de NB-IoT	
Ancho de Banda	200 KHz
Modulación	QPSK
Máxima tasa de datos	200 kbps
Tamaño máximo de payload	1600 bytes
Alcance	1 km (urbano), 10 km (rural)
Cifrado	LTE
Referencia	[18]

Tabla 3 Características principales de NB-IoT.

A continuación, en la Figura 6, se detalla un resumen de otros protocolos de comunicación agrupados y clasificados por capa.

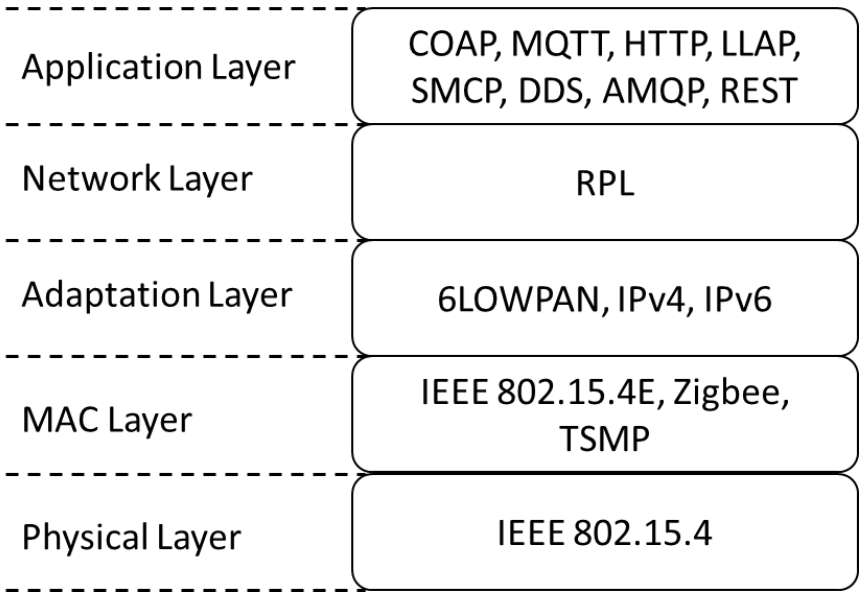

Figura 6 Protocolos Iot.

- **Physical Layer:**
  - IEEE 802.15.4: este protocolo está diseñado para dispositivos de bajo consumo en la capa física. Define la capa física y MAC para el funcionamiento de las redes LR-WPAN (Redes de Área personal Inalámbricas de Baja Velocidad). Es compatible con muchos protocolos de capa superior, por ejemplo, ZigBee. La velocidad de trasferencia se encuentra alrededor de los 250 KB/s para un rango de comunicación de entre 10 y 100 metros. También presenta múltiples características como el acceso al medio con CSMA/CD para evitar colisiones, reserva de intervalos de tiempo garantizados para admitir comunicaciones en tiempo real y aspectos de seguridad.
- **MAC Layer:**
  - Zigbee es un protocolo de comunicación inalámbrica desarrollado para crear redes WPAN (Redes Inalámbricas de Área Personal). Está diseñado para aplicaciones que requieren una baja tasa de transmisión y un bajo coste energético [19].

- **Adaptation Layer:**
  - 6LOWPAN permite enviar y recibir paquetes IPv6 a través de redes basadas en el estándar 802.15.4. Una de las principales ventajas de este protocolo, es el poder usar direcciones IPv6. Lo cual permite que el mundo IoT avance a la próxima generación de internet, permitiendo integrar IPv6 en LR-WPAN [20].
- **Network Layer:**
  - RPL es un protocolo usado para enrutamiento en redes inalámbricas con un consumo bajo de energía. Se basa en vectores de distancia y trabaja sobre el estándar IEEE 802.15.4 [21].

- **Aplication Layer:**

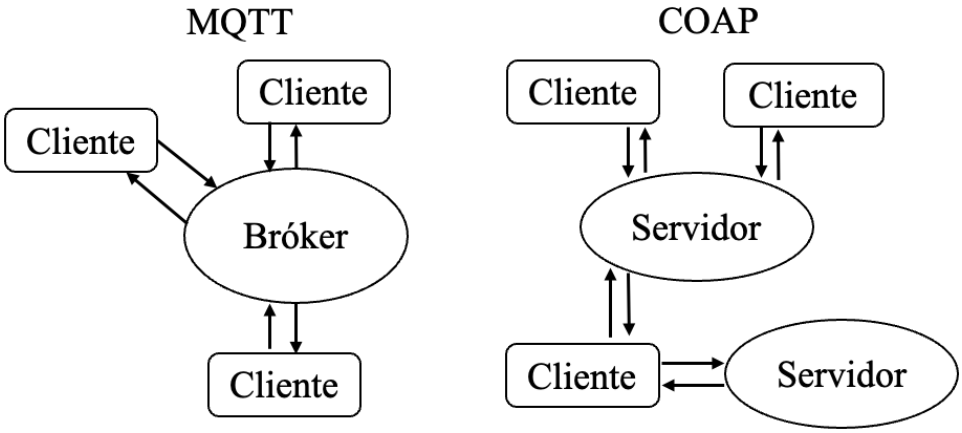

Tabla 4 Arquitectura de MQTT y COAP

  - MQTT es un servicio de mensajería que implementa el modelo publicador/suscriptor. Los dispositivos intercambian información a través de un servidor centralizado llamado bróker. La recepción o envío de los datos se realizan a través de asuntos llamados “topics”. Este protocolo trabaja sobre el protocolo de comunicación TCP y a diferencia de HTTP, cada conexión se mantiene abierta y se reutiliza cada vez que se quiere realizar una comunicación [22]. MQTT define 3 niveles de calidad de servicio: nivel 0 donde no hay garantía en la entrega de los mensajes, nivel 1 donde si hay garantía de entrega, pero puede existir entregas duplicadas y nivel 2 donde se garantiza la entrega sin la existencia de duplicados.

  - COAP es un protocolo de transferencia de ficheros diseñado para poder ser usados por dispositivos con un bajo coste energético. Con limitaciones de memoria y procesamiento computacional. Las comunicaciones son 1 a 1 y como protocolo de transporte usa UDP. Los mensajes ocupan menos bytes y no está garantizada la entrega de los mensajes. Permite la interconexión con un servidor HTTP a través de un proxy que cruce ambos protocolos [22]. Este protocolo se encuentra frecuentemente en escenarios donde hay muchos nodos finales, con pocos recursos y que no están enviando continuamente.

2.3 Sistemas de protección en IoT

Los sistemas de protección IoT basados en firmas más comunes son los IDS basados en firmas y los firewalls de aplicación Web.
Los IDS basados en firmas examinan el tráfico de red y detectan patrones que corresponden con algunas de las firmas conocidas almacenadas en su base de datos. Los IDS basados en firmas más comunes son: Snort, Suricata y Bro [23].
Estos IDS tienen reglas por defecto que permiten detectar las principales amenazas conocidas. Cuando se quieren detectar ataques específicos para entornos IoT la forma de trabajar es distinta. La mayoría de los trabajos que presentan IDS basados en firmas para entornos IoT, extraen características de dataset públicos como pueden ser direcciones IP, número de puertos usados en las comunicaciones, tamaño medio de los paquetes, etc. Estos dataset suele estar formado por tráfico IoT en formato IPFIX o la propia captura del tráfico de red. Este tráfico suele estar mezclado con ataques, lo cual permite generar firmas a partir de las características seleccionadas y así poder detectar dichos ataques.
En Tabla 5 se muestra a modo de ejemplo algunos de los trabajos más recientes que han generado firmas para IDS orientados a entornos IoT a partir de dataset con tráfico de ataque.

Trabajo	Dataset usado
[24]	NSL-KDD
[25]	ISCX-UNB
[26]	Bot-IoT

Tabla 5 principales trabajos sobre IDS basados en firmas.

Los WAF (Web Application Firewall) protegen y detectan amenazas en la capa de aplicación, en concreto para aplicaciones web sobre el protocolo HTTP/S. Este mecanismo de seguridad se ubica entre la aplicación web e internet, permitiendo así monitorizar y proteger el servidor. Esto se consigue examinando las solicitudes HTTP/S que se realizan al servidor y detecta o bloquea aquellas solicitudes que coincidan con una firma conocida que se encuentre en la base de datos de firmas almacenadas como maliciosas.
Existen muchas aplicaciones WAF como pueden ser [27]: Barracura, Bee Ware, Citrix, F5, Fortinet, Imperva, Breach Security. Pero la más famosa es ModSecurity [28] debido a que se trata de una aplicación de código libre, estable y muy flexible. ModSecurity es un módulo de ciberseguridad que se instala en dispositivos que realizan la función de un proxy inverso [29]. Modsecurity proporciona reglas de configuración llamadas SecRules. Estas reglas son flexibles por lo que se pueden configurar de acuerdo con las necesidades de seguridad de la aplicación Web que se desee proteger. Uno de los proyectos más usado como fuente de reglas para ModSecurity es OWASP® ModSecurity Core Rule Set (CRS) [30] donde se pueden descargar reglas ya definidas para poder detectar los ataques más comunes dirigidos a servidores Web.

También existen los honeypots, el cual es un mecanismo de ciberprotección que se usa para detectar y recopilar información sobre las posibles amenazas y ataques de ciberseguridad. En el contexto de la iluminación inteligente (Smart-Lighting), los honeypots pueden desempeñar un papel importante en la protección de la infraestructura de iluminación inteligente contra amenazas cibernéticas. Los honeypots simulan dispositivos IoT de iluminación inteligente que presentan alguna vulnerabilidad o puntos de acceso a la red, esto atrae a posibles atacantes. Cuando un atacante intenta obtener acceso con el honeypot, esta iteración se detecta y es registrada. Esto proporciona a los administradores de seguridad información valiosa sobre las tácticas y técnicas utilizadas por los atacantes, permitiendo definir nuevas políticas o tácticas para evitar ser atacados.

A continuación, se detallan algunas de las soluciones honeypots existentes:

- Argos [31] es un honeypot demoninado como alta interacción. Estos tipos de honeypots son los más avanzados, debido a que permiten la interacción completa con el atacante. Esto es debido a que los honeypots de este tipo emulan sistemas completos reales. Argos en concreto está basado en Qemu [32], un emulador de código libre que permita simular sistemas IoT completos.
- HoneyD [33] es un honeypot de baja interacción que expone servicios en la red emulados para detectar cuando atacante intenta acceder a ellos. Este honeypot permite también la simulación de redes, engañando al atacante de la existencia de una red que realmente es simlada.
- Cowrie [34] es un honeypot de interacción híbrida. Este tipo de honeypot combina las características de las dos anteiores, permitiendo crear una red simulada y un emular un sistema completo. Una vez el atacante accede al honeypot, su actividad queda registrada en una base de datos y muestra un nivel alto de detalle de los pasos seguidos por el atacante.

Los honeypots están muy bien como capa de protección adicional, pero no aseguran la detección completa de un intruso en la red, debido a que sólo se detectará si el atacante intenta conectarse al honeypot.

Existen otros tipos de soluciones de ciberseguridad IoT, como es por ejemplo el caso de Nozomi Networks [35]. Se trata de una plataforma que permite a través de la recolección del tráfico IoT, realizar un descubrimiento de todos los activos y servicios que cursan la red IoT, creando un inventariado. Una vez crea la base de datos de lo que es normal, entra en el modo de detección donde detecta cuando existe alguna anomalía que se salga de lo normal aprendido. Esta solución es bastante completa, pero es una solución particular de una empresa y tiene un coste alto. Además, es una solución cerrada que no permite añadir o editar sus módulos con soluciones de código libre.

Como hemos visto en el punto 1.1, hay multitud de amenazadas posibles en redes IoT. También existen muchas soluciones específicas para cada una de ellas, el problema es que no existen soluciones de bajo coste que puedan ofrecer ciberseguridad en las diferentes capas, que sea modular y permita añadir, editar o eliminar módulos de detección de forma sencilla con herramientas o implementaciones de código libre, además, también se observa la necesidad de un sistema que pueda combinar la detección de ataques conocidos, con la detección de ataques no conocidos o más bien conocido como los ataque de tipo de zero-day. Debido a todas estas limitaciones que presentan las técnicas actuales, vemos la necesidad de un sistema complementario de detección basado en anomalías, que permita la detección de ellas en las diferentes capas.

3 ARQUITECTURA PROPUESTA

Hay dos cosas que son infinitas: el universo y la estupidez humana; de la primera no estoy muy seguro.
Albert Einstein

En este punto se presenta una arquitectura de ciberseguridad orientada al caso de uso Smart-Lighting. En él se presenta las amenazas que se desean detectar, los requisitos que necesitan cumplir la arquitectura propuesta y finalmente, se detalla el diseño llevado a cabo, realizando una descripción de cada uno de los módulos que conforman el sistema completo.

La arquitectura de ciberprotección presentada está orientada a la detección de amenazas específica para entornos Smart-Lighting y es un primer paso de prueba de concepto, por lo que este trabajo no protege frente a todas las amenazadas descritas en el punto 1.1 Amenazas y ciberprotección en Smart-Lighting. En la Tabla 6 se muestra las principales amenazas IoT y sus posibles contramedidas o mecanismos de detección posibles.

Ataque	Posibles contramedidas/detección
Sleep deprivation	Detección de anomalías, a través del uso de la red IoT por parte de los dispositivos conectados a ella.
Capturing & Fake node injection	Control de acceso a la red.
Malicious code injection	Detección de anomalías en los comandos recibidos/enviados por los dispositivos IoT.
Eavesdropping	Cifrado de los datos trasmitidos.
DDOS	Límite del uso de la red por parte de los dispositivos IoT. Detección de comportamientos anómalos en la periodicidad de las comunicaciones.
Sniffing	Cifrado de las comunicaciones IoT.
Traffic analysis	Honeypots que detecten cuando un atacante realiza un escaneo en la red. Detección de anomalías en el uso de los puertos usados en las comunicaciones.
Credential Access	Monitorización de los logs del servicio que ofrece acceso a través de credenciales. Uso de Honeypots.
SQL injection	Inclusión de un WAF en la red IoT.
Storage	Cifrado de los datos almacenados.
False Data Injection	Detección anomalías en los datos IoT.
False Command Injection	Detección anomalías en los comandos IoT.

Tabla 6 Contramedidas/detección a los ataques IoT más comunes.

Como hemos visto en la Tabla 6, la mayoría de las amenazas IoT se pueden detectar haciendo uso de técnicas basadas en la detección de anomalías. Las soluciones existentes no permiten detectar la gran mayoría de las amenazas con una única solución. El alcance de este trabajo se centra en la detección de las amenazas en la capa de red y Aplicación, basado en anomalías, donde dentro de estas capas, el sistema detectará las siguientes amenazas:

- **Capa de red:** DDOS attack, Traffic analysis attacks y Credential Access.
- **Capa de aplicación:** False Data Injection y False Command Injection.

El sistema se centra en la detección de anomalías porque permite la detección de la mayoría de las amenazas presentadas. Además, es un sistema modular y complementario con otros de las soluciones presentadas anteriormente.

De cada uno de los tipos de ataques, luego se realizan diferentes pruebas y variantes. Todas ellas se encuentran descritas en la Tabla 9.

A continuación, se detallan los requisitos que debe cumplir la arquitectura propuesta:

- Diseño e implementación de un sistema detección de intrusos (IDS) que permita detectar anomalías tanto del mundo IT como el OT.
- Bajo consumo de recursos debido a las limitaciones de los dispositivos IoT.
- Solución completamente pasiva que no interfiera en la red IoT.
- Compatible con sistemas basados en firmas como IDS o WAF.

3.1 Diseño

En la Figura 7 se presenta la arquitectura propuesta y se detalla el diseño de cada uno de los bloques que la conforman.

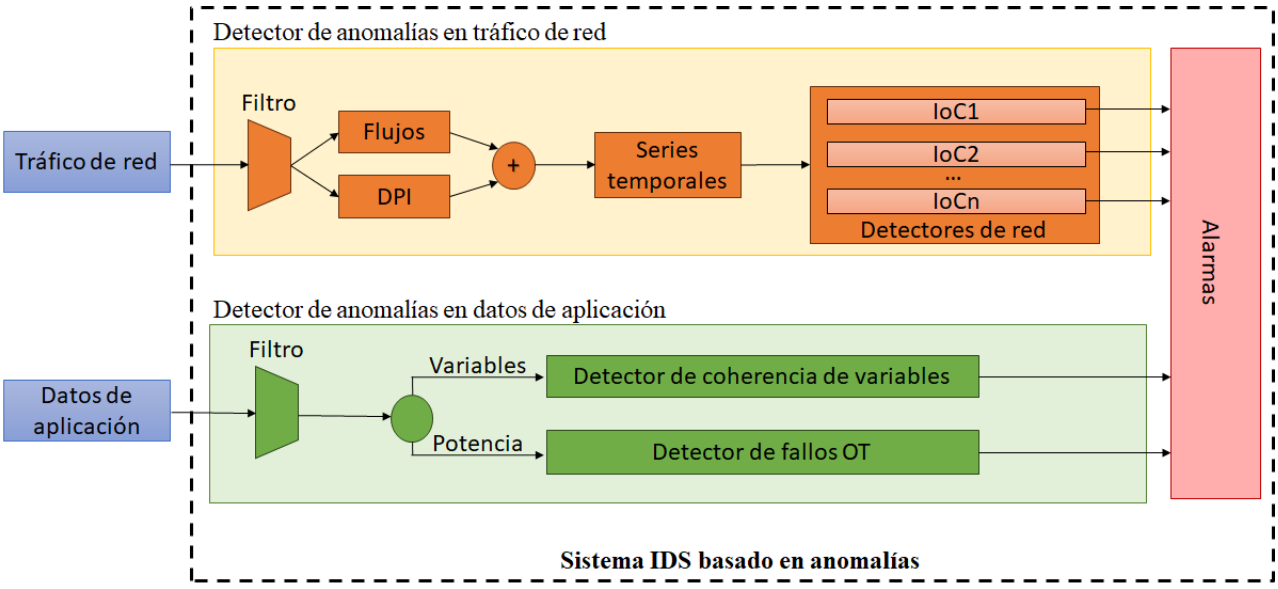

Figura 7 Sistema IDS propuesto.

El sistema propuesto detecta anomalías en las capas OSI 3-4 a través de un módulo de detección de anomalías en tráfico de red y también en la capa 7 del modelo OSI (capa de aplicación) a través de un módulo de detección de anomalías en datos de aplicación. A continuación, vamos a detallar cada uno de los componentes del sistema propuesto.

3.1.1 Datos de entrada y salida del sistema

Existen dos posibles fuentes de entrada al sistema IDS propuesto:

a. **Los paquetes de tráfico de red** (en formato PCAP) que se puede obtener realizando un port-mirror en algunos de los router o conmutadores que forman la red IP. El port-mirror [36] es una función que posee la mayoría de los conmutadores de red que permite copiar el tráfico que cursa un puerto en otro puerto distinto. Esto permite escuchar el tráfico de la red IoT de forma pasiva y sin que la red se pueda ver afectada.
b. **Los datos de aplicación** intercambiados entre los dispositivos IoT y el servidor MQTT/COAP que pueden obtenerse accediendo al archivo de registro del propio servidor o a la base de datos local.

Adicionalmente, se añade un filtro a la entrada de cada uno de los módulos de detección, esto permite analizar únicamente el tráfico o los datos de aplicación de interés, evitando insertar ruido a los detectores propuestos. En el caso del módulo de red, se permite filtrar las tramas que se desean analizar. Esto permite por ejemplo que se pueda analizar el tráfico que sea exclusivamente el generado por cada una de las UCA.

Las salidas posibles del sistema están relacionadas con las amenazas que se desean detectar, a continuación, se detallan las posibles salidas del sistema:

- DDOS attacks: estos ataques dejan muchos rastros en la red y obtendremos salidas relacionadas con un aumento en la cantidad de paquetes de red, un aumento en el número de dispositivos conectados a la red y un cambio en la periodicidad en las comunicaciones.
- Traffic analysis attacks: estos ataques tienen salidas similares al anterior, pero dejan también rastros sobre el uso de los puertos usados en las comunicaciones, porque un atacante podría realizar un sondeo a número de puertos específicos para extraer información sobre los dispositivos conectados a la red.
- Credential Access: este ataque también dejaría rastro en la red, es por ello por lo que es necesario la implementación de detectores específicos que alerten de un uso anormal a accesos a servicios con uso de credenciales.
- False Data Injection: si un dato ha sido manipulado, el sistema detectará y notificará con un mensaje sobre la coherencia de las variables.
- False Command Injection: cuando un atacante introduce comandos anómalos en los dispositivos, estos generan errores en la operación. Este comportamiento es detectado como un fallo OT y es notificado.

3.1.2 Módulo 1: detector de anomalías en tráfico de red

Los paquetes del tráfico de red capturados se procesan de forma periódica para extraer los flujos de tráfico, que incluyen información relevante de la red y de los encabezados de transporte de los paquetes. La ventana de tiempo para los procesos de medición y exportación es configurable. Al utilizar flujos de tráfico en lugar del tráfico en bruto (los paquetes) sin procesar, se consigue reducir significativamente el tamaño del conjunto de datos que tiene que procesar el detector. También es posible realizar una inspección profunda de paquetes (DPI) antes de generar los flujos, para así conseguir enriquecerlos con información extra de interés (por ejemplo: sistema operativo, geolocalización, aplicación usada en la comunicación, etc.) Esto sólo es posible siempre que el tráfico no se encuentre cifrado.
Una vez exportado los flujos, se generan series temporales para obtener estadísticos relacionadas con el tráfico de la red en general (por ejemplo, número total de mensajes ARP en la subred, número total de conexiones, etc) o específicos del dispositivo (p. ej., número de IP externas que solicitaron una conexión con un dispositivo IoT en particular).
Finalmente, de cada una de las series temporales se genera un indicador de compromiso (IoC), el cual permite detectar y generar alarmas basadas en anomalías. Para ello, se utilizan técnicas estadísticas debido a que proporcionan un buen resultado, son simples y permiten adaptarse en entornos Smartlighting.
A continuación, se detallan los detectores implementados para los IoC y en Tabla 7 Definición de IoC's para el tráfico de red.Tabla 7 se detallan los IoC y los detectores asociados a cada uno de ellos.

3.1.2.1 Detector D1: Media Móvil Simple (SMA)

Este detector se basa en el cálculo de la media móvil simple de una serie temporal de valores. Dada una serie temporal de valores X, para calcular la media móvil simple de una muestra X_n, se calcula la media de los
w valores valores anteriores a X_n. Siendo w un parámetro que define el tamaño de la ventana para el cálculo de media móvil.
El cálculo de la SMA para un instante n se define como $\overline{X_{SMA_n}} = \frac{X_n + X_{n-1} + \cdots + X_{n-w-1}}{w}$. Una vez calculada la media móvil simple, se comprueba para cada uno de los valores dados si se cumple alguna de las siguientes condiciones y se considera anomalía cuando alguna de las expresiones es verdadera.

- En función del parámetro $cond$, se evalúa una condición u otra:
  - $LCL|_{cond=0}: X_n < \overline{X_{SMA_n}} - k \times \sigma_{n-1}$
  - $UCL|_{cond=1}: X_n > \overline{X_{SMA_n}} + k \times \sigma_{n-1}$
  - Siendo σ_{n-1} la desviación estándar de los w valores anteriores a X_n y k un factor multiplicador.

En la siguiente figura se muestra un ejemplo de este detector, la serie azul corresponde con el número total de flujos vistos en una red cada 10 minutos. Tomando como parámetros $k = 10$ y $w = 3$ y aplicando SMA se obtiene la serie verde. Finalmente, se obtiene el límite (para $cond = 1$), la serie punteada en negro. En rojo, se puede apreciar como una de las muestras ha sido detectada como anómala al superar el límite UCL calculado por el detector.

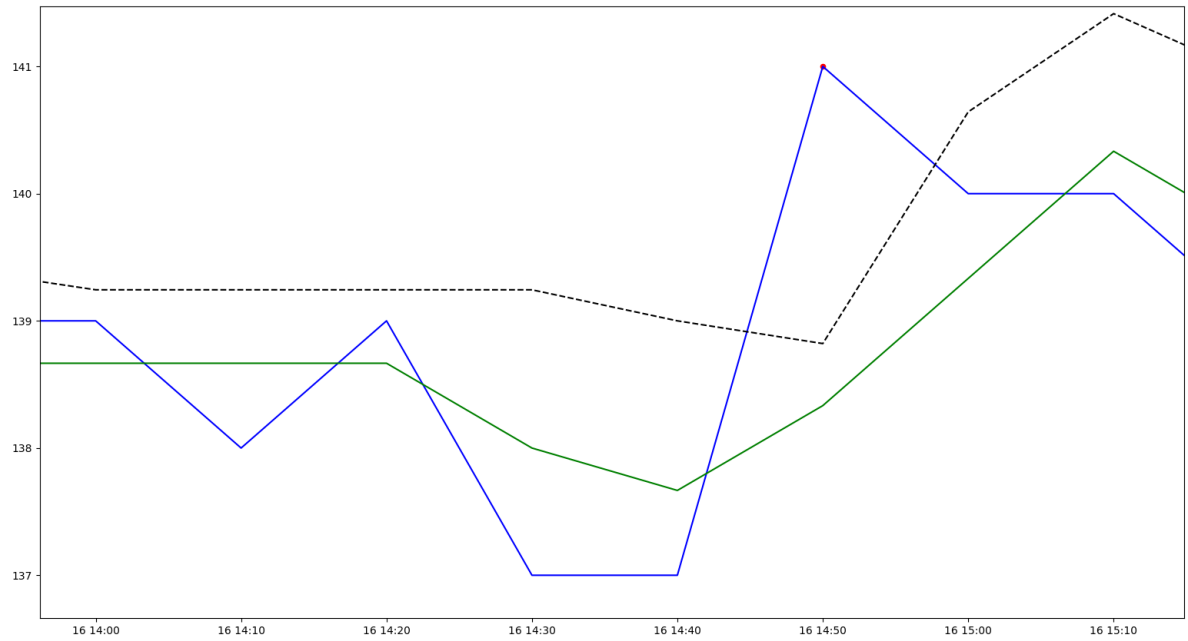

Figura 8 Ejemplo de detector D1 SMA.

3.1.2.2 Detector D2: Media Móvil Exponencial Ponderada (EWMA)

Este detector se basa en el cálculo de la media móvil exponencial ponderada (EWMA) de una serie temporal de valores. El cálculo de la EWMA se define como $\overline{X_{EWMA_n}} = \alpha \times X_n \times (1 - \alpha) \times \overline{X_{EWMA_{n-1}}}$, donde α es el parámetro de suavizado con valores comprendidos entre [0,1]. X_n el valor en el instante n de la serie de valores X y $\overline{X_{EWMA_{n-1}}}$ el valor de la media móvil exponencial ponderada en el instante $n-1$.
Una vez calculada la EWMA sobre la serie de valores X, se considera anomalía aquellos valores que cumplan la condición de $X_n < \overline{X_{EWMA_{n-1}}} - k \times \sigma_{EWMA_{n-1}}$.

- Siendo $\sigma_{EWMA_{n-1}}$ la desviación típica de la EWMA en el instante anterior $(n-1)$ a la muestra evaluada y definida como $\sigma_{EWMA_n} = \sqrt{(1-\alpha)\left(\sigma_{EWMA_{n-1}} + \alpha\left(X_n - \overline{X_{EWMA_{n-1}}}\right)^2\right)}$

ID	Descripción	Detector
IoC1	Número total de flujos IPFIX identificados con tráfico SSH	D1
IoC2	Número total de flujos IPFIX identificados con tráfico ICMP	D1
IoC3	Número total de flujos IPFIX	D1
IoC4	Número total de direcciones IP origen distintas identificadas en los flujos IPFIX	D1
IoC5	Número total de puertos destinos distintos identificadas en los flujos IPFIX	D1
IoC6	Número total de direcciones IP destinos distintas identificadas en los flujos IPFIX	D1
IoC7	Periodicidad mínima registrada en los flujos IPFIX[2]	D2

Tabla 7 Definición de IoC's para el tráfico de red.

[2] Para generar este indicador, se calcula la periodicidad mínima de cada una de las direcciones IP origen y posteriormente se selecciona la periodicidad más pequeña registrada.

3.1.3 Módulo 2: detector de anomalías en datos de aplicación

Este subsistema recibe como entrada los datos almacenados en la base de datos de series temporales (TSDB) y detecta anomalías en ellos. Para ello, define los siguientes detectores:

3.1.3.1 Detector de coherencia de variables

Este detector recibe los datos registrados por un grupo de variables (por ejemplo: timestamp, voltaje, intensidad, potencia activa, potencia reactiva, etc). Este grupo de datos se registra para cada uno de los dispositivos Smartlighting. El objetivo principal de este submódulo es el de detectar anomalías en estos grupos de variables. Para ello, hace uso del análisis de reducción de dimensiones (PCA) y el control de residuos Q, que permite detectar anomalías a través de la coherencia de variables. A continuación, se detalla la implementación de este detector:

- Dada la matriz de datos $X = [x_1^T, \dots, x_m^T]$ con n observaciones y m variables correlacionadas. Se obtiene la matriz X_s que es el resultado de normalizar la matrix de datos X con media 0 y varianza 1.
- Se realiza la descomposición en valores singulares (SVD) de la matriz X_s, aplicando la ecuación $X_s = T \cdot P^T$, siendo T la matriz $T = [T_1, \dots, T_m]$ de variables no correlacionadas, o más bien conocida como componentes principales, donde cada componente T_i es combinación lineal de las variables originales y P es la matriz que contiene los autovectores asociados con la matriz de covarianza de X_s definida en la siguiente ecuación:
  - $\Sigma = \frac{1}{n-1} \cdot X^T \cdot X = P \Lambda P^T$, siendo Λ una matriz diagonal que contiene los autovalores en orden decreciente.
- Una vez reducida las dimensiones con (PCA), seleccionamos las primeras A componentes principales de la matriz P . El objetivo es que las componentes seleccionadas describan una parte significativa de la covarianza de los datos observados. De esta manera, nos quedamos con una matriz P_A con dimensión mxA . Cuando en la fase de test tengamos una nueva observación con m variables x_n, calculamos el vector de puntuación $t_n = x_n \cdot P_a$
- Al reducir las dimensiones, la estimación P_A tendrá diferencias respecto a la observación original. Este error se define como $e_n = x_n - t_n \cdot P_A^T$. Denominamos residuo de la matriz, el error que se produce al reducir el número de componentes. Esto nos permite definir el estadístico Q, que indica la proyección de una muestra en el subespacio de residuos. Proporcionando un indicador de cómo de bien se adapta una observación al modelo PCA. El estadístico Q se define como $Q = \|I_m - P_A P_A^T\|^2$.
- Para usar un control de residuos, seleccionamos las A componentes principales y en la fase de entrenamiento se encuentra un umbral máximo para el residuo generado con los datos de entrenamiento. Para ello, se calcula el residuo generado: $Q_{train} = \left\|\left(I - \hat{P}\hat{P}^T\right) \cdot X_{s_{train}}\right\|^2$ y se encuentra un límite Q_α usado para la fase de test y definido en la siguiente ecuación: $Q_\alpha = \mu_{Q_{train}} + k \cdot \sigma_{Q_{train}}$
  - siendo $\mu_{Q_{train}}$ y $\sigma_{Q_{train}}$ la media y desviación típica del vector de residuos Q_{train}
- Una vez encontrado un límite residual Q_α. Para cada nueva observación en la fase de test, se comprueba la siguiente condición: $Q_{test} > k \cdot Q_\alpha$ y se considera que es una anomalía en caso de cumplir dicha condición.
  - Siendo Q_{test} el residuo generado al proyectar la nueva observación en el modelo PCA encontrado en la fase de entrenamiento y k un factor multiplicador del límite Q_α encontrado en la fase de aprendizaje.

3.1.3.2 Detector de fallos OT

Este detector recibe los datos registrados del consumo de potencia de las luminarias Smartlighting. Estos datos contienen una marca de tiempo y el consumo de potencia registrado en dicho instante de tiempo para cada una de las luminarias. El objetivo de este detector es el de encontrar anomalías en la serie temporal del consumo de potencia registrado. Estas anomalías pueden ser provocadas por fallos en la operación, debido a luminarias fundidas o robo de potencia en la línea de luminarias.
A continuación, se detallan los pasos llevados a cabo:

- Dada la matriz de datos $X(t,p)$ formada por dos vectores columnas: $T = [t\_1, t\_2, \dots, t\_n]$ y $P = [p\_1, p\_2, \dots, p\_n]$, siendo t_i el instante de tiempo del valor del consumo de potencia registrado p_i. Existe una nueva observación cada M minutos. Al vector de valores de potencia P se le aplica una media móvil simple (SMA) con una ventana de w muestras. Este nuevo vector lo llamaremos P' y se encuentra definido en la siguiente ecuación:
  - $P' = \frac{p_n + p_{n+1} + \cdots + p_{n+w-1}}{w}$
  - Esto se hace para realizar un suavizado sobre los valores originales y así conseguir reducir los cambios bruscos en la serie temporal.
- Teniendo el nuevo vector de valores suavizados P', calculamos la desviación típica móvil con una ventana de w muestras, obteniendo el vector P_σ definido en la siguiente ecuación:
  - $P_\sigma = \sqrt{\frac{\sum_{i=n}^{n+w-1}(p_i - \overline{P'})^2}{w}}$, sabiendo que $\overline{P'}$ es la media de las w muestras suavizadas. Al vector P_σ también se le aplica una media móvil simple con una ventana de w muestras, obteniendo el vector de valores suavizados P'_σ
- Teniendo los vectores de valores suavizados P' y P'_σ, se calcula la diferencia con la muestra registrada 24 horas antes, que corresponde con la muestra N anterior. obteniendo el vector de diferencia de valores de potencia $P^* = [p'_{n+1} - p'_1, \dots, p'_{n+1} - p'_{n-N}]$, siendo $n > N$.
- Una vez obtenido los vectores P^* y P'_σ, calculamos la media del vector P'_σ, permitiendo obtener $\mu_{P'_\sigma}$, siendo este un umbral para la desviación típica y así no detectar en las zonas de paso de encendido a apagado y viceversa. Esto permite reducir el número de falsos positivos.
  - $\mu_{P'_\sigma} = \frac{\sigma_1 + \cdots + \sigma_n}{n}$, siendo n el número de muestras seleccionadas para el entrenamiento.
- Se considera que una muestra es anómala si durante i muestras consecutivas la desviación típica suavizada se encuentra por debajo de un cierto umbral ($[p_{\sigma_n}, \dots, p_{\sigma_{n+k}}] < k \cdot \mu_{P'_\sigma}$) y, además, la diferencia de potencia supere el siguiente umbral: $[|p^*_n|, \dots, |p^*_{n+k}|] > V$, siendo k un factor multiplicador del umbral $\mu_{P'_\sigma}$ y V la diferencia máxima en valor absoluta de potencia permitida.

Para ayudar a entender al lector cómo funciona este algoritmo de forma gráfica, en la Figura 9 se muestra cada uno de los vectores calculados:

- En azul los valores de potencia sin suavizar.
- En amarillo los valores de potencia aplicando la media móvil.
- En rojo los valores de la desviación típica aplicando la media móvil.
- En verde la diferencia de los valores de potencia con la muestra registrada 24 horas antes.

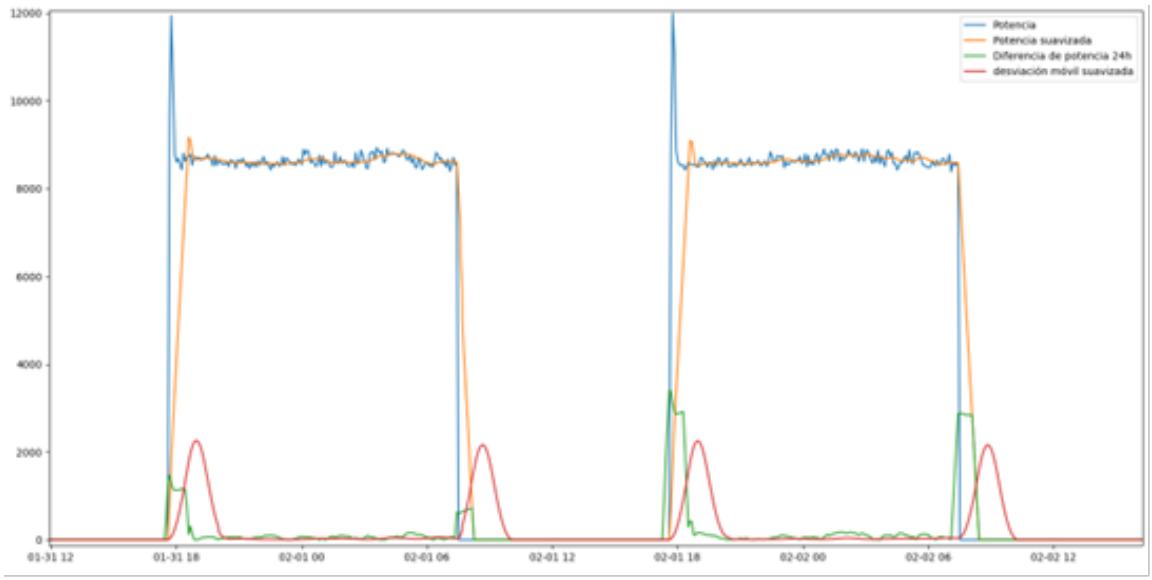

Figura 9 Ejemplo del algoritmo fallos OT.

4 IMPLEMENTACIÓN Y PRUEBAS

La verdadera sabiduría está en reconocer la propia ignorancia.
Sócrates

En este capítulo se presenta la implementación realizada del sistema propuesto en el punto anterior y las pruebas realizadas sobre el mismo.

4.1 Implementación

El sistema propuesto se ha implementado en el lenguaje de programación Python. En la siguiente figura se detalla cada uno de los componentes implementados:

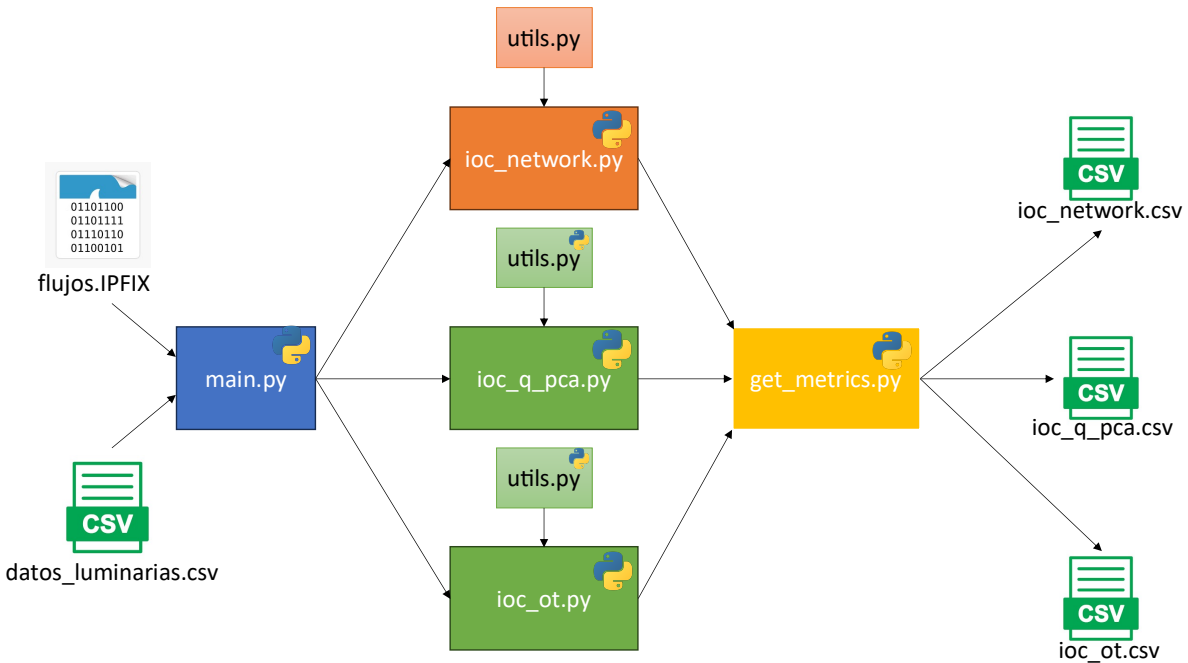

Figura 10 Implementación del sistema IDS propuesto.

4.1.1 Módulos implementados

Como vemos en la Figura anterior donde se muestra la implementación del sistema IDS propuesto, se observan diferentes módulos implementados en el lenguaje de programación Python.

4.1.1.1 Módulo main.py

Primero tenemos el módulo **main.py** que es el encargado de recibir las dos fuentes de datos: los flujos IPFIX y los datos registrados por las UCA. Este primer módulo se encarga de realizar el filtrado de los datos, descartando las trazas IPFIX que no son de interés, como por ejemplo las correspondientes con el tráfico HTTP, debido a que se desea caracterizar el tráfico que genera cada una de las UCA y este tráfico corresponde con el protocolo de intercambio de datos COAP. Esto nos permite definir indicadores de compromiso basados en patrones y así poder detectar anomalías que se alejan de los patrones de comportamiento. Para el caso de los datos registrados por las UCA, este módulo permite realizar descartar registros que contienen errores, provocados por ejemplos por errores de medidas.
Una vez filtrado todos los datos, este módulo se encarga de ejecutar (con los datos ya filtrados) los módulos de tratamiento de datos y detección de anomalías: **ioc_network.py**, **ioc_q_pca.py** e **ioc_ot.py**.

4.1.1.2 Módulo ioc_network.py

Este módulo recibe las trazas IPFIX y es el encargado de generar los indicadores de compromiso de red para detectar anomalías en ellos. Adicionalmente, usa un fichero Python **utils.py** donde se encuentran definidas funciones auxiliares para el tratamiento de los datos.
Este módulo implementa el desarrollo teórico presentado en Módulo 1: detector de anomalías en tráfico de red. En la siguiente tabla se detallan los parámetros utilizados para cada uno de los indicadores de compromiso:

ID	Detector
IoC1	$D1(k = 1, w = 5)$
IoC2	$D1(k = 1, w = 5)$
IoC3	$D1(k = 10, w = 3)$
IoC4	$D1(k = 2, w = 7)$
IoC5	$D1(k = 1, w = 5)$
IoC6	$D1(k = 1, w = 5)$
IoC7	$D2(k = 3, \alpha = 0.1, cond = 0)$

Tabla 8 Parametrización del módulo ioc_network.py

4.1.1.3 Módulo ioc_q_pca.py

Este módulo recibe los datos eléctricos registrados por las luminarias descritos en la Tabla 10 y es el encargado de encontrar anomalías en la coherencia de las variables. Adicionalmente, usa un módulo de **utils.py** para realizar un tratamiento y preprocesado de los valores registrados.

4.1.1.4 Módulo ioc_ot.py

Este módulo recibe los valores del consumo de potencia para cada una de las luminarias y es el encargado de encontrar anomalías OT como son el de una luminaria fundida o un robo de potencia provocado por un consumo excesivo. Como todos los módulos anteriores, este también necesita de un módulo externo **utils.py** para realizar el tratamiento de los datos.

4.1.1.5 Módulo get_metrics.py

Este módulo recoge todos los resultados de los módulos **ioc_network.py**, **ioc_q_pca.py** e **ioc_ot.py,** calcula unas métricas de rendimiento y exporta los resultados para cada uno de los módulos en un fichero de fomato csv[3] para poder estudiar el rendimiento de cada uno de los detectores definidos.

4.2 Pruebas

En esta sección se detallan las pruebas y resultados obtenidos para el sistema propuesto. Para ello, primero se definen unos indicadores y métricas que permitan medir el rendimiento de cada uno de los componentes del sistema.
A continuación, se detallan los indicadores usados para poder medir el rendimiento:

- **Verdadero Positivo** (VP): se considera un VP cuando se detecta un ataque por algunos de los elementos (por ejemplo, IoC). Para probar el módulo a nivel de aplicación, se considera un VP si se detecta una anomalía en la muestra envenenada introducida en el conjunto de datos. Para probar el módulo a nivel de red, se insertan ataques en el archivo pcap de entrada en instantes aleatorios en el período de evaluación, verificando que la marca de tiempo del evento detectado coincida con la del ataque detectado.
- **Falso Positivo** (FP): un FP indica que se ha detectado una anomalía, pero no se insertó un ataque en el conjunto de datos original.
- **Verdadero Negativo** (VN): VN indica una muestra normal clasificada como normal.
- **Falso Negativo** (FN): se da un FN cuando una muestra contaminada no es detectada.

Con estos cuatros indicadores podemos evaluar el rendimiento de cada uno de los componentes con diferentes métricas:

- **True Positive Rate** (TPR): esta métrica, también conocida como sensibilidad, nos permite medir la capacidad de detección de anomalías y se calcula de la siguiente forma:
  - $TPR = \frac{VP}{VP+FN}$
- **True Negative Rate** (TNR): esta métrica, también conocida como especificidad y permite dar una medición sobre como de bien detecta de forma correcta un verdadero negativo nuestro sistema.
  - $TNR = \frac{VN}{VN+FP}$
- **Rendimiento:** definida las métricas TPR y TNR, necesitamos una medida global que nos indique el rendimiento general del sistema, para ello, usamos la siguiente métrica rendimiento:
  - $\eta = \sqrt{TPR \cdot TNR}$

[3] Fichero donde sus campos se encuentran separados por el carácter coma ","

4.2.1 Ataques implementados

Para poder medir el rendimiento del sistema propuesto, se han implementados ataques tanto en el nivel de red, como en el nivel de aplicación.

- **Los ataques de nivel de red** (AT1-5) se han implementado sobre una red de Docker que permite simular la red de dispositivos conectados. Además, se ha añadido un dispositivo adicional a la red que permita lanzar ataques sobre el resto de los dispositivos. El dispositivo que hace de atacante se ejecuta sobre el sistema operativo Kali Linux[4], esta distribución está preparada con múltiples herramientas diseñadas para la generación de ataques. Una vez ejecutado los ataques, se captura el tráfico de red generado y se generan trazas IPFIX con la herramienta Tranalyzer [5]. Cuando se exportan estas trazas IPFIX, se editan los campos correspondientes a la fecha y hora de inicio y fin de cada una de las trazas, para poder mezclarlas con las trazas de tráfico limpio y así poder comprobar si se detecta correctamente cada uno de los ataques implementados.

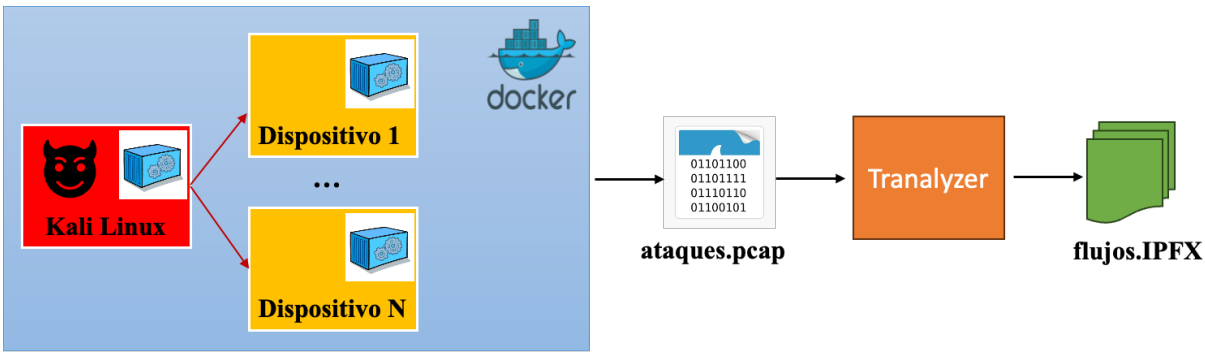

Figura 11 Arquitectura de la implementación de los ataques de red.

- **Los ataques de nivel de aplicación** (AT6-11) se han implementado contaminando las muestras originales a través de una función de Python. Esta función selecciona ***N*** muestras aleatorias según el criterio dado y modifica el valor de una variable. Esto permite simular comportamientos que se pueden ocurrir en instalaciones reales, como son por ejemplo el simular el apagado de una luminaria o el de introducir valores aleatorios como haría un ataque de manipulación de datos.

[4] https://www.kali.org/
[5] https://tranalyzer.com/

En la siguiente tabla se detallan los ataques implementados identificados también según la matriz MITRE [37]:

ID	Sub-técnica	Ataque	MITRE ID	Comando/descripción	Tipo
AT1	TCP SYN flooding	DoS	T1499.001	`hping3 -p <port> -S –flood <ip_target>`	Red
AT2	DdoS	DdoS	T1498.001	`hping3 –rand-source -d <data_size> <ip_target> -p <port> --flood`	Red
AT3	ICMP scan	Scanning	T1595.001	`nmap -sO <ip>/<mask>`	Red
AT4	Service scan	Scanning	T1595.002	`nmap -sV <ip>`	Red
AT5	Brute Force	Credential Access	T1110.001	`hydra -l <username> -P <path_to_wordlist> <IP> -t <number_of_threads> ssh`	Red
AT6	False Data Injection	Data Manipulation	T1565	Una de las variables del dispositivo (Pow), cambiado su valor mínimo al valor máximo registrado.	Aplicación
AT7	False Command Injection	Data Manipulation	T1565	Una de las variables del dispositivo (Pow), cambiado su valor máximo al valor mínimo registrado.	Aplicación
AT8	False Command Injection	Data Manipulation	T1565	Una de las variables del dispositivo (Pow), cambiado su valor mínimo al doble del valor medio registrado.	Aplicación
AT9	False Command Injection	Data Manipulation	T1565	Una de las variables del dispositivo (Pow), cambiado su valor mínimo a un valor aleatorio comprendido entre el mínimo y el máximo registrado.	Aplicación
AT10	False Command Injection	Data Manipulation	T1565	Se resta 600w al valor de la potencia (Pow) registrada por un dispositivo durante 24 horas, simulando el fundido de una luminaria.	Aplicación
AT11	False Command Injection	Data Manipulation	T1565	Se introduce consumo de potencia (Pow) cuando el dispositivo se encuentra apagado, simulando el robo de potencia.	Aplicación

Tabla 9 Ataques implementados.

4.2.2 Dataset

El sistema de alimenta con dos tipos de datos: tráfico de red y datos registrados por las luminarias. Cada 5 minutos se registra una lectura para cada una de luminarias Smartlighting. Se tienen datos de un total de 64 dispositivos. Estos datos corresponden con los siguientes intervalos de tiempo: del 9 de enero al 14 de febrero y del 12 de julio al 19 de julio, todo para el año 2021. En la Tabla 10 se detallan las variables que se registran para cada una de las luminarias. Para las variables $\boldsymbol{Pow_n}$, $\boldsymbol{APow_n}$, $\boldsymbol{RPow_n}$,$\boldsymbol{Int_n}$, $\boldsymbol{Vol_n}$ y $\boldsymbol{PF_n}$ se registran 3 valores, una para cada línea de luminarias Smartlighting. Se registran un total de 25 variables.

Variable	Descripción
$\boldsymbol{Pow_n}$	Potencia consumida
$\boldsymbol{APow_n}$	Potencia activa
$\boldsymbol{RPow_n}$	Potencia reactiva
$\boldsymbol{Int_n}$	Intensidad
$\boldsymbol{Vol_n}$	Voltaje
$\boldsymbol{PF_n}$	Factor de potencia
$\boldsymbol{PF}$	Factor de potencia total
$\boldsymbol{Freq}$	Frecuencia
$\boldsymbol{Vol}$	Voltaje total
$\boldsymbol{Pow}$	Potencia consumida total
$\boldsymbol{APow}$	Potencia activa total
$\boldsymbol{RPow}$	Potencia reactiva total
$\boldsymbol{TS}$	Marca de tiempo

Tabla 10 Variables registradas por las UCA.

Además del conjunto de datos registrados por las luminarias, se tiene un dataset de flujos de red. Este segundo dataset también tiene una periodicidad de 5 minutos para cada una de las luminarias. Se tienen registros en el intervalo del 30 de junio al 19 de julio, todo para el año 2021.

Los protocolos identificados en estos registros son:

- COAP: usado para el envío de datos registrados por las luminarias Smartlighting.
- HTTP: usado por la aplicación de gestión de las luminarias Smartlighting.

Cada flujo de red exporta un total de 45 campos, en la Tabla 11 se detallan los campos utilizados por el sistema IDS propuesto.

Campo	Descripción
timestamp	Marca de tiempo del registro IPFIX
src_ip	Dirección IP origen
dst_ip	Dirección IP destino
src_port	Número de puerto origen
dst_port	Número de puerto destino
src_mac	Dirección MAC origen
dst_mac	Dirección MAC destino
application_name	Nombre de aplicación L7

Tabla 11 Campos usados de los flujos IPFIX.

4.2.3 Resultados

En este punto se detallan los resultados obtenidos para cada uno de los módulos de detección implementados.

4.2.3.1 Resultados para el módulo de coherencia de variables

En la Tabla 12 se pueden observar los resultados medios obtenidos para el módulo de coherencia de variables, fijando el número de componentes principales a 4 y parametrizando el factor multiplicador k.

	Sin anomalías	AT6	AT7	AT8	AT9
k	FPR	η	η	η	η
4	0,029	0,96	0,93	0,94	0,71
5	0,024	0,96	0,93	0,93	0,66
6	0,021	0,96	0,92	0,92	0,61
7	0,016	0,95	0,91	0,91	0,57
8	0,009	0,93	0,91	0,9	0,53

Tabla 12 Rendimiento del módulo de coherencia de variables.

Al usar 4 componentes principales y un valor de K igual 6, se obtiene una tasa de falsos positivos del 2% y una capacidad de detección de mayor del 90% para los ataques AT6, AT7 y AT8. En la siguiente tabla se muestra los resultados de forma individual para cada una de las luminarias.

Ahora vamos a analizar el comportamiento de forma individual para cada uno de los dispositivos, para ello se fijan los valores de los parámetros k=6 y componentes principales a 4, insertando anomalías del tipo AT6, el resultado de este análisis se puede apreciar en la Tabla 13.

Luminaria	FPR	TPR	TNR	η
10.247.114.176	0	0	1	0
10.247.114.188	0,473	1	0,527	0,527
10.92.9.111	0,239	1	0,761	0,761
10.247.90.102	0,214	1	0,786	0,786
10.92.9.112	0,114	1	0,886	0,886
10.247.114.143	0,104	1	0,896	0,896
10.247.114.168	0,073	1	0,927	0,927
10.247.90.106	0,048	1	0,952	0,952
10.247.114.175	0,015	1	0,985	0,985
10.247.114.172	0,015	1	0,985	0,985
10.247.114.170	0,013	1	0,987	0,987
10.247.114.149	0,013	1	0,987	0,987
10.247.114.158	0,011	1	0,989	0,989
10.247.90.105	0,011	1	0,989	0,989
10.247.114.177	0,01	1	0,99	0,99
10.247.114.132	0,008	1	0,992	0,992

10.247.114.136	0,007	1	0,993	0,993
10.92.9.109	0,005	1	0,995	0,995
10.247.114.169	0,005	1	0,995	0,995
10.247.114.150	0,005	1	0,995	0,995
10.247.114.186	0,004	1	0,996	0,996
10.247.90.107	0,004	1	0,996	0,996
10.92.9.113	0,004	1	0,996	0,996
10.92.9.110	0,004	1	0,996	0,996
10.247.114.130	0,004	1	0,996	0,996
10.247.114.152	0,003	1	0,997	0,997
10.247.114.148	0,003	1	0,997	0,997
10.247.114.131	0,003	1	0,997	0,997
10.247.114.183	0,003	1	0,997	0,997
10.247.114.154	0,002	1	0,998	0,998
10.247.88.200	0,002	1	0,998	0,998
10.247.114.165	0,002	1	0,998	0,998
10.247.114.141	0,002	1	0,998	0,998
10.247.114.157	0,002	1	0,998	0,998
10.247.114.134	0,002	1	0,998	0,998
10.247.114.153	0,002	1	0,998	0,998
10.247.114.146	0,002	1	0,998	0,998
10.247.114.162	0,002	1	0,998	0,998
10.247.114.140	0,002	1	0,998	0,998
10.247.114.167	0,002	1	0,998	0,998
10.247.90.104	0,002	1	0,998	0,998
10.247.114.178	0,002	1	0,998	0,998
10.247.114.135	0,002	1	0,998	0,998
10.247.114.181	0,001	1	0,999	0,999
10.247.114.161	0,001	1	0,999	0,999
10.247.114.159	0,001	1	0,999	0,999
10.247.114.187	0,001	1	0,999	0,999
10.247.114.145	0,001	1	0,999	0,999
10.247.114.139	0,001	1	0,999	0,999
10.247.114.144	0,001	1	0,999	0,999
10.247.114.166	0,001	1	0,999	0,999
10.247.114.142	0,001	1	0,999	0,999
10.247.114.185	0,001	1	0,999	0,999
10.247.114.160	0,001	1	0,999	0,999
10.247.114.133	0,001	1	0,999	0,999
10.247.114.171	0,001	1	0,999	0,999
10.247.114.163	0,001	1	0,999	0,999
10.247.114.184	0,001	1	0,999	0,999
10.247.114.137	0,001	1	0,999	0,999
10.247.114.164	0,001	1	0,999	0,999
10.247.114.155	0,001	1	0,999	0,999
10.247.114.174	0	1	1	1

10.247.114.138	0	1	1	1
10.247.114.179	0	1	1	1
10.247.114.180	0	1	1	1
10.247.114.151	0	1	1	1
10.247.114.129	0	1	1	1
10.247.114.182	0	1	1	1

Tabla 13 Resultados de coherencia de variables de forma individual para k=6, CP=4 y ATP6.

Analizando los resultados obtenidos de forma individual, se pueden observar que hay ciertos dispositivos donde el algoritmo no funciona de forma correcta:

Luminaria	FP	TP	Justificación
10.247.114.176	0	0	Valores muy dispares en las variables (posibles errores de medidas). Picos en las variables de voltaje con valores de 327680. Picos en las variables de potencia reactiva con valores de 84482457. Valores de potencia a cero, teniendo valores de intensidad y voltaje no nulos.
10.247.114.188	1738	100	El número de falsos positivos es provocado cuando se registran valores de potencia no nulos. Esta zona presenta unas fluctuaciones que provoca un incremento en el residuo, detectado este comportamiento como anómalo.
10.92.9.111	874	100	El número de falsos positivos es provocado en los valores de la potencia de las últimas muestras usadas como test.
10.247.90.102	791	100	Valores muy dispares en sus variables, provocando un residuo alto. Esta UCA necesita un factor multiplicador de K igual a 8 para obtener un 95% de capacidad de detección y bajando el número de falsos positivos de 791 a 20.

Tabla 14 Justificación de resultados de coherencia de variables.

Ahora vamos a analizar de forma individual cada una de las cuatro luminarias:

La luminaria 10.247.114.176 provoca residuos muy altos y anormales debido a valores muy dispares en sus variables, provocando un límite de Q_α alto, haciendo que la de detección sea nula. Este comportamiento se puede apreciar en la Figura 12 y luego en la Figura 13 se puede apreciar como al tener un valor tan alto de Q_α la detección es nula.

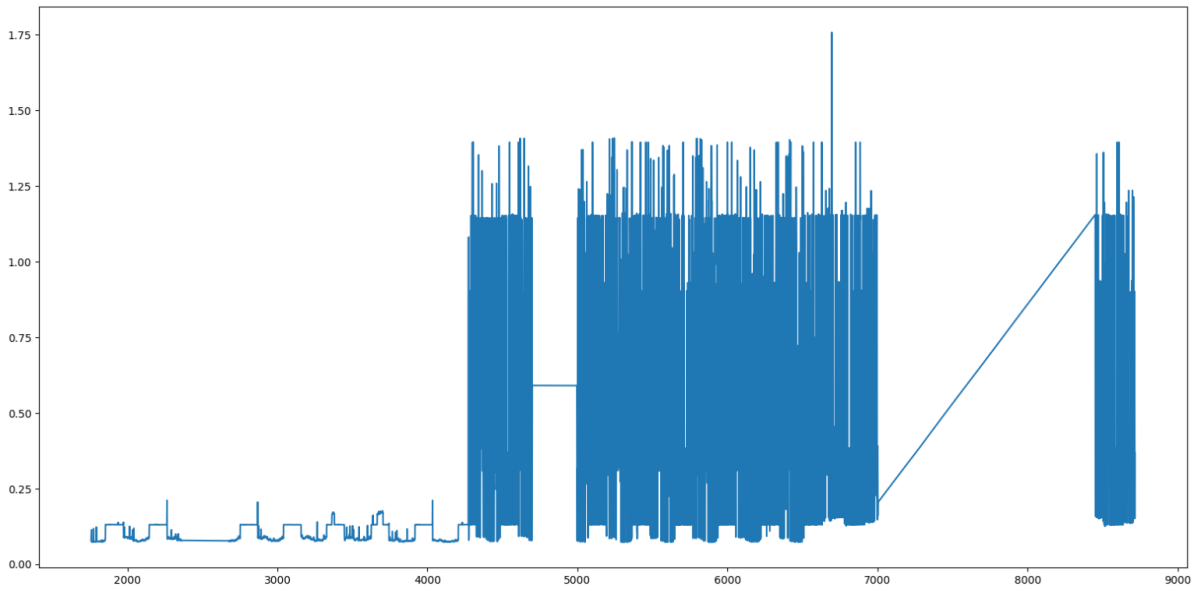

Figura 12 Residuo del conjunto de entrenamiento para la luminaria 10.247.114.176.

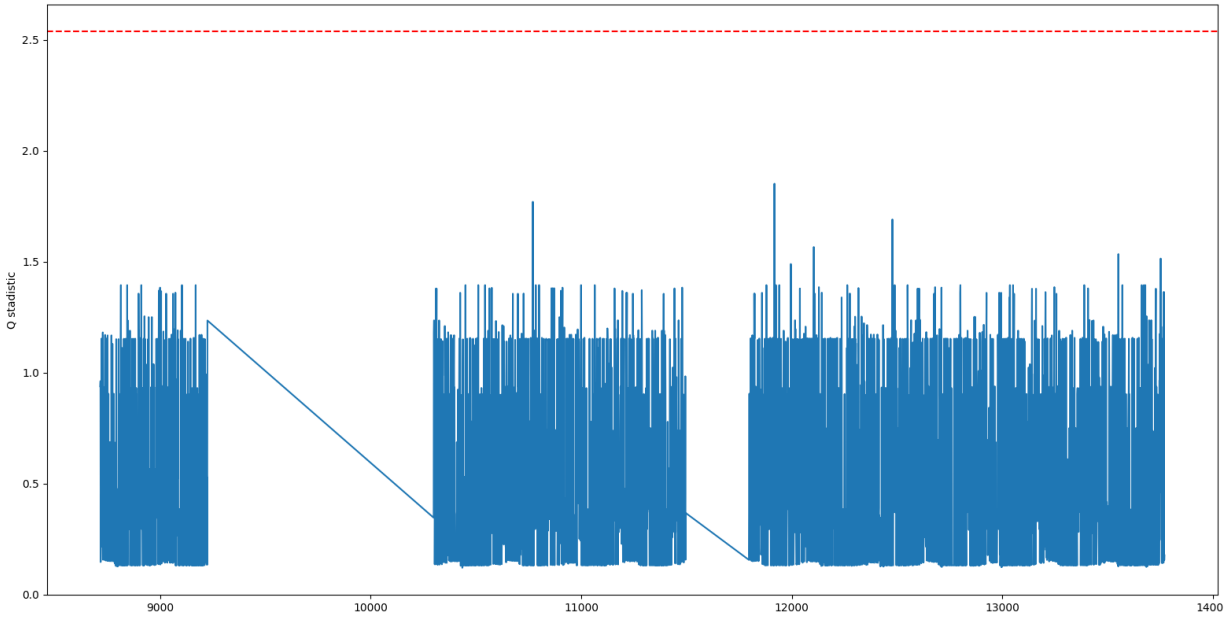

Figura 13 Residuo del conjunto de test para la luminaria 10.247.114.176.

Si ahora analizamos los valores de las variables voltaje, intensidad y potencia, se observa que no guardan relación entre ellas, lo cual nos lleva a la conclusión de que puede ser debido a errores de medidas:

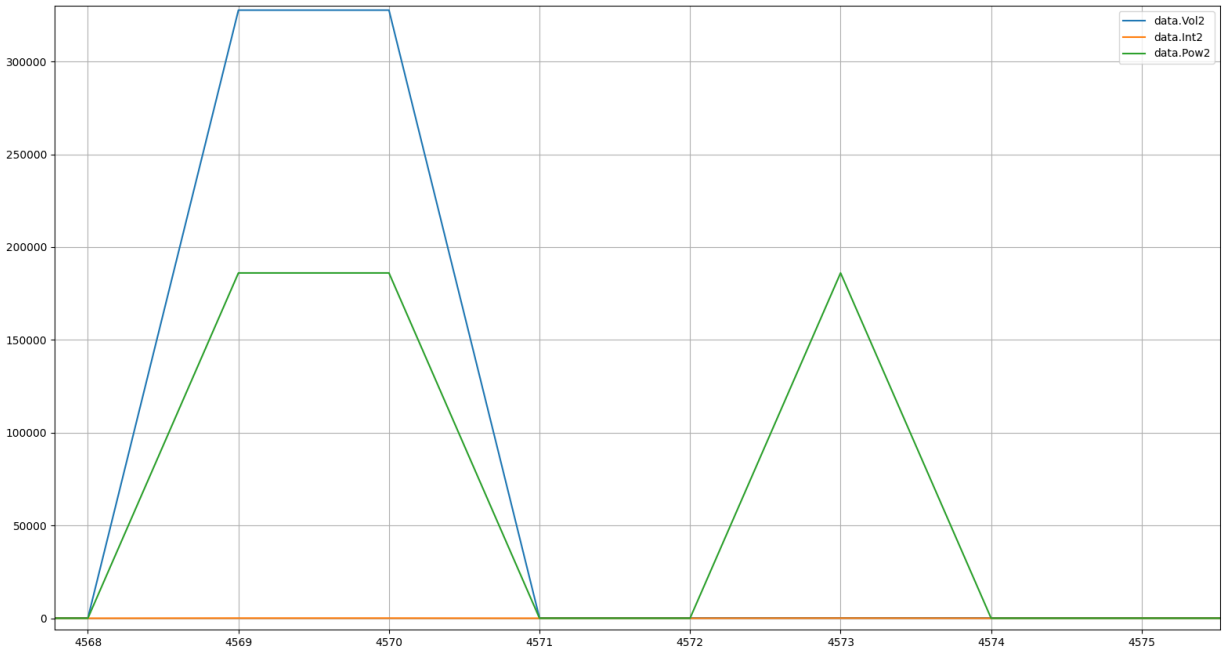

Figura 14 Relación de variables para la luminaria 10.247.114.176.

Analizando ahora los valores de potencia de la luminaria 10.247.114.188 en la Figura 15. Se observa como en la zona activa existe una fluctuación, el cual provoca en el cálculo de residuos una alta tasa de falsos positivos. Además, también se observa en la Figura 16, que los valores de potencia usados para el test son mucho más grandes que los valores usados para el entrenamiento.

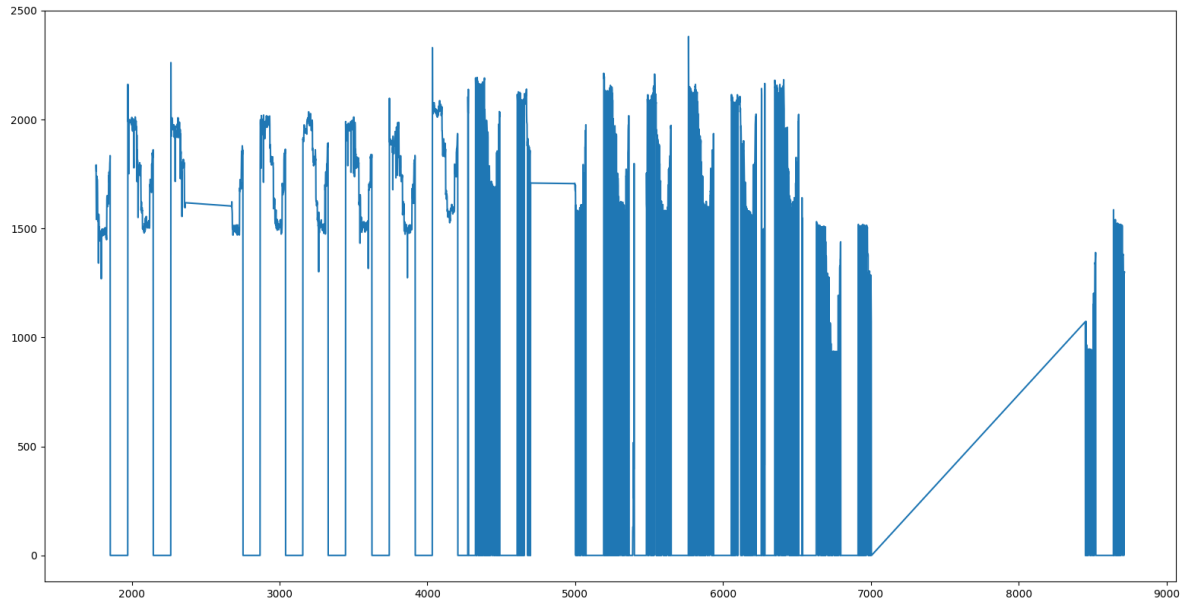

Figura 15 Valores de potencia de entrenamiento para la luminaria 10.247.114.188

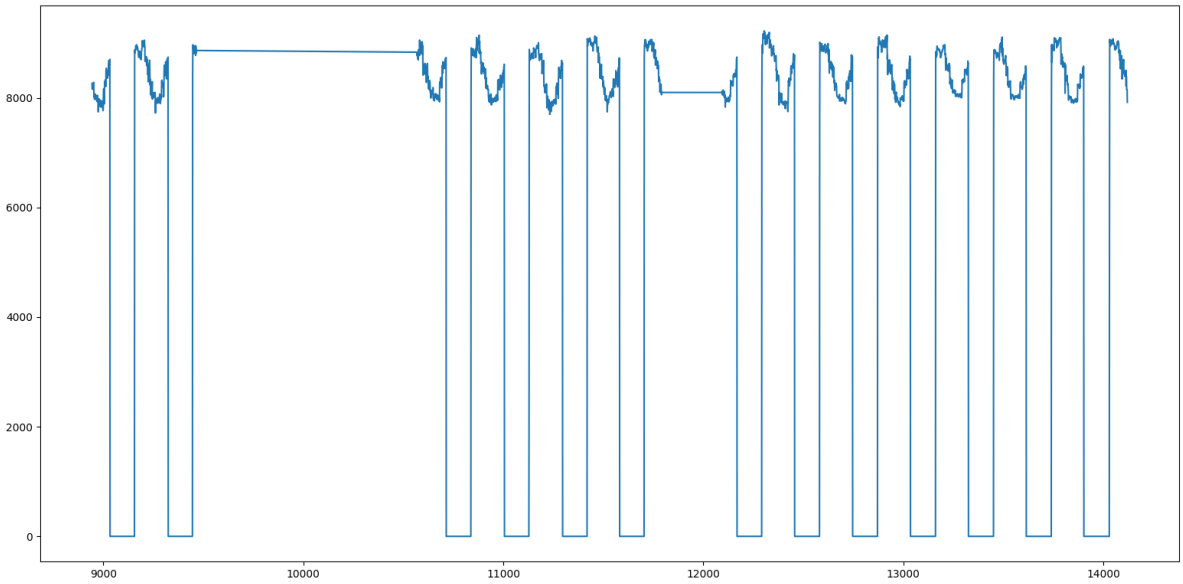

Figura 16 Valores de potencia de test para la luminaria 10.247.114.188.

Finalmente, en la Figura 17 se puede observar la gran cantidad de falsos positivos provocado por lo comentado anteriormente. Esto podría solucionarse aumentando el valor del parámetro k.

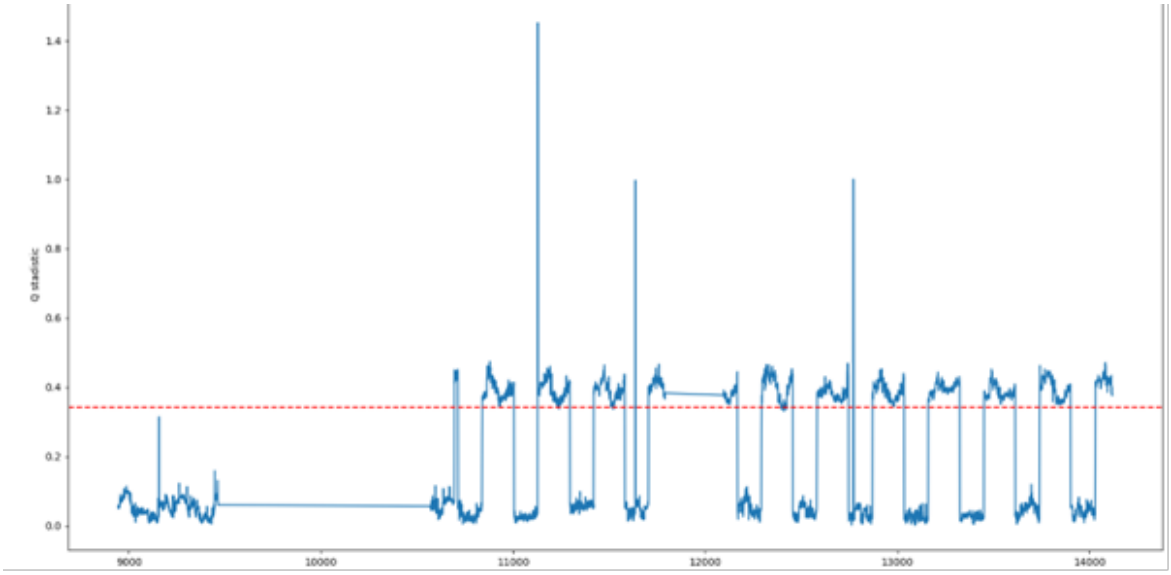

Figura 17 Residuo para los datos de test de la luminaria 10.247.114.188.

Para la luminaria 10.92.9.111 se observan muchos falsos positivos ubicados principalmente en las últimas muestras del conjunto de test y este comportamiento podemos apreciarlo en la Figura

18

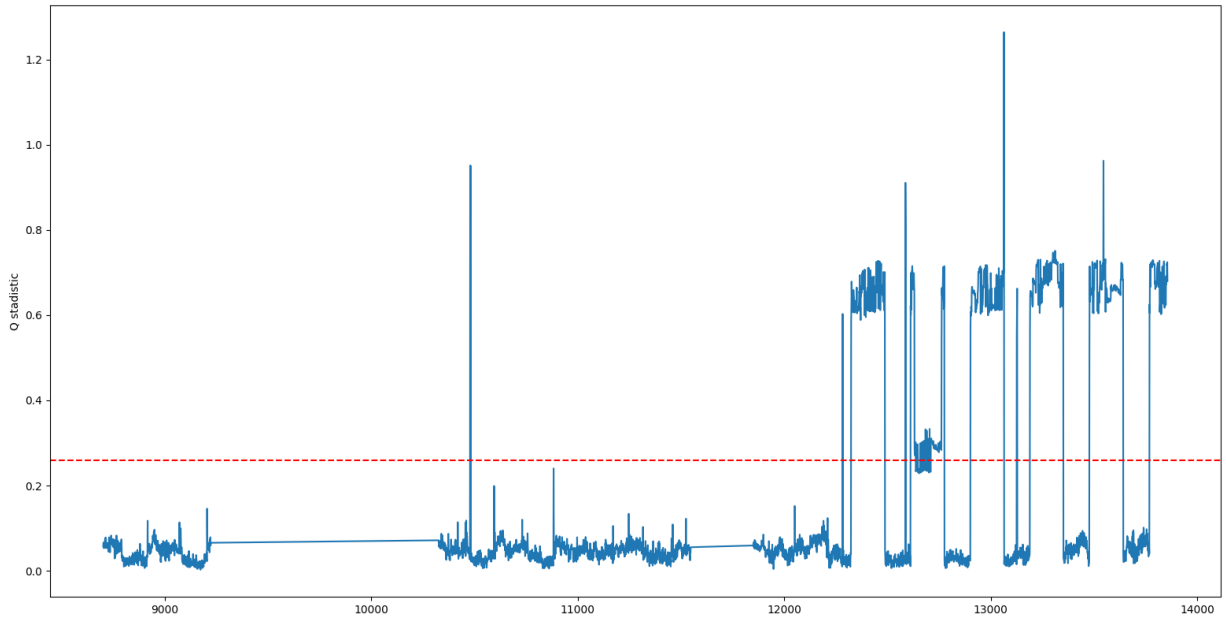

Figura 18 Residuo de test para la luminaria 10.92.9.111

Si ahora observamos los valores de potencia de la luminaria 10.92.9.111, vemos en la Figura 19 como en las últimas muestras del test, los valores de potencia son más alto que el resto, provocando esto un residuo más alto.

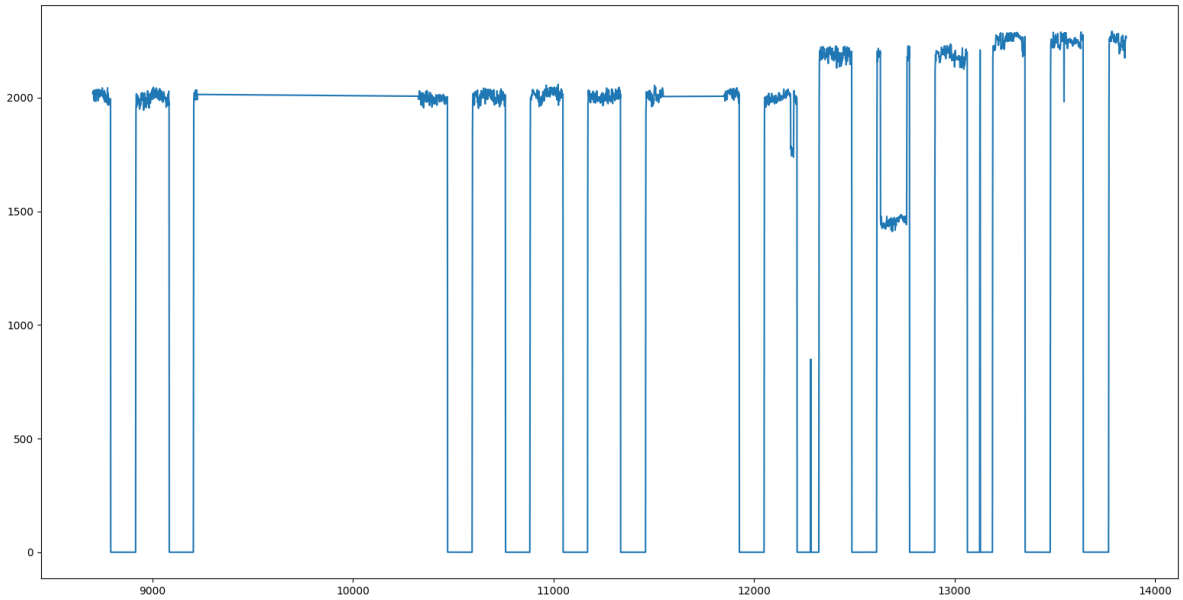

Figura 19 Valores de potencia para la luminaria 10.92.9.111.

En la luminaria 10.247.90.102 ocurre algo similar, se observan muchos falsos positivos ubicados principalmente en las últimas muestras del conjunto de test.

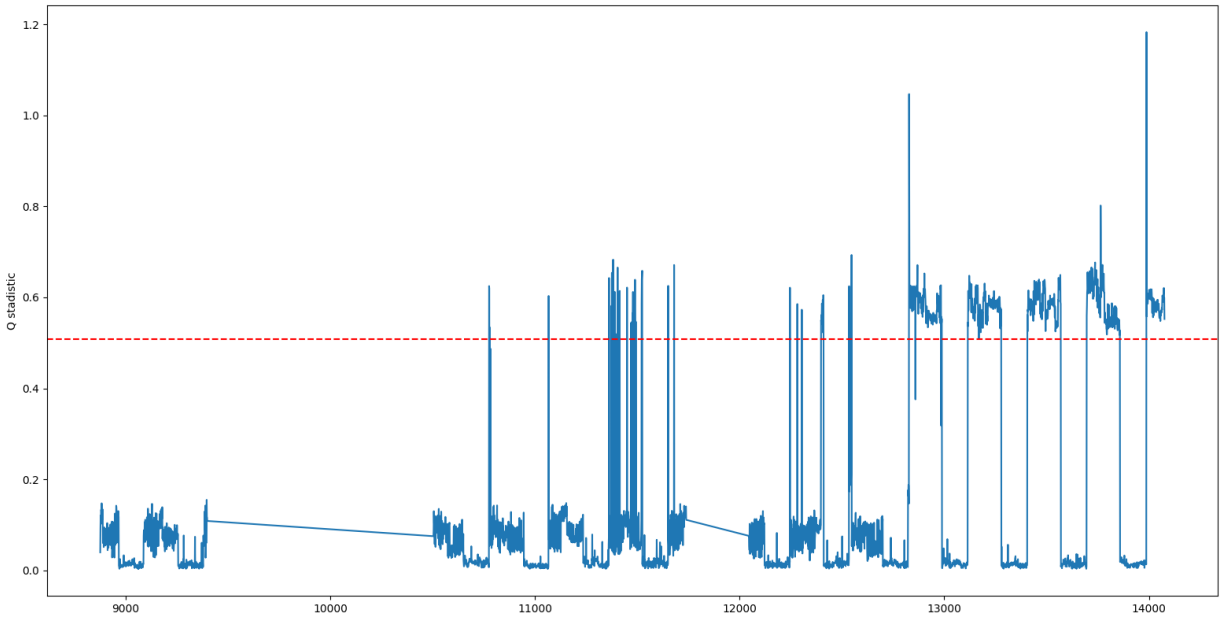

Figura 20 Residuos de test para la luminaria 10.247.90.102.

Para finalizar, se muestra un ejemplo de detección para la luminaria 10.247.114.169 incluyendo un total de 100 anomalías del tipo AT6, esta luminaria presenta una capacidad de detección del 99% cuando los valores de los parámetros son *K=6* y 4 componentes principales. En la Figura 21 se puede observar este comportamiento.

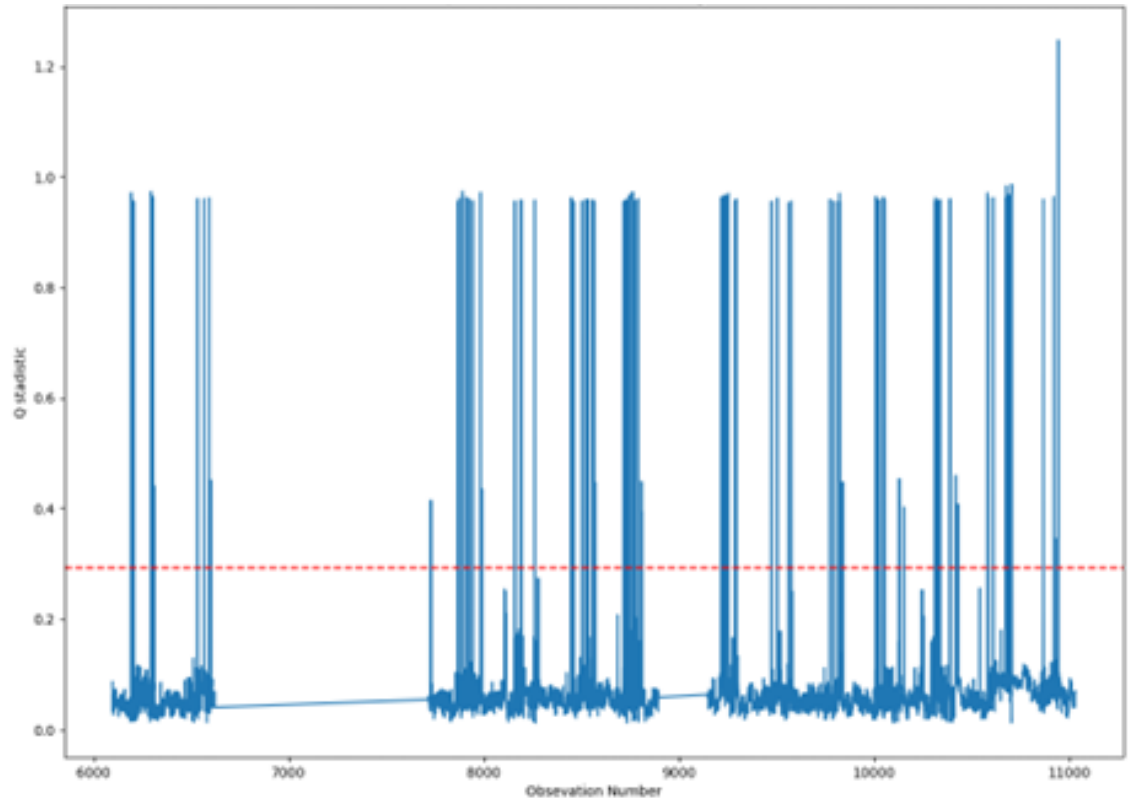

Figura 21 Residuo de test para la luminaria 10.247.114.169 con anomalía tipo AT6.

Si hacemos un zoom en uno de los picos, se puede observar cómo los datos de esta luminaria tienen un residuo aproximadamente de 0.1 y se eleva a 0.9 al contaminar las muestras. Detectando correctamente dichas anomalías.

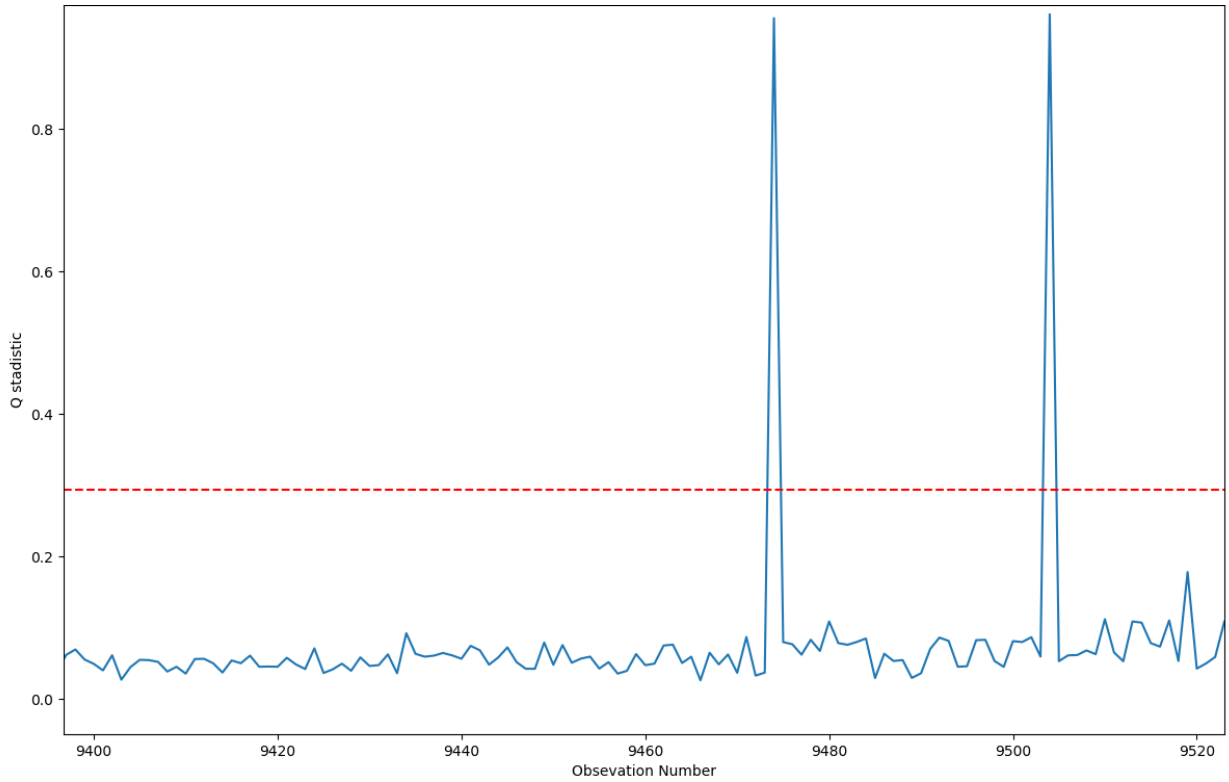

Figura 22 Zoom de datos de residuos para la luminaria 10.247.114.169.

4.2.3.2 Resultados para el módulo de coherencia de fallos OT

Para medir el rendimiento del módulo de fallos OT se ha usado la siguiente franja de tiempo: 14/01/2021-14/02/2021 y se han realizado las siguientes pruebas:

- **Luminaria fundida (AT10):** se selecciona un día y se resta a la potencia registrada un total de 600W.
- **Robo de potencia (AT11):** se selecciona un día y se suma a la potencia registrada un total de 600W.

Estas dos pruebas se han realizado contaminando las muestras de todos los días (descartando el primer día).
A continuación, se muestran los resultados para los dos tipos de pruebas, definiendo un umbral de máxima diferencia de potencia $V = 500$ y variando i (número de muestras consecutivas cumpliendo los dos umbrales) para valores de: [24, 36, 48, 60, 72] que corresponde con [2, 3, 4, 5, 6] horas.

Horas	FPR	TPR	TNR	η
2	0,062	0,865	0,938	0,892
3	0,046	0,851	0,954	0,891
4	0,034	0,818	0,966	0,877
5	0,024	0,664	0,976	0,789
6	0,013	0,448	0,987	0,616

Figura 23 Resultados para AT10.

En la Figura 24 se muestra un ejemplo de resultado contaminando las muestras del 15 de enero cuando la luminaria se encuentra encendida, restando al nivel de potencia registrado un total de 600W (AT10). En la figura se puede apreciar (marcado en rojo) cómo se detecta de forma correcta la anomalía insertada.

Figura 24 Ejemplo de detección de anomalías AT10.

Ahora se muestran los resultados obtenidos cuando se inserta anomalías del tipo AT11

Horas	FPR	TPR	TNR	η
2	0,062	0,828	0,938	0,864
3	0,046	0,833	0,954	0,871
4	0,034	0,823	0,966	0,866
5	0,024	0,816	0,976	0,859
6	0,013	0,793	0,987	0,841

Figura 25 Resultados para AT11.

En la Figura 26 se muestra un ejemplo de robo de potencia donde se contamina las muestras del día 15 de enero donde la luminaria se encuentra apagada. En dichas muestras, se ha insertado un nivel de potencia de 600W (AT11) y se ha puede apreciar cómo se ha detectado de forma correcta el robo de potencia.

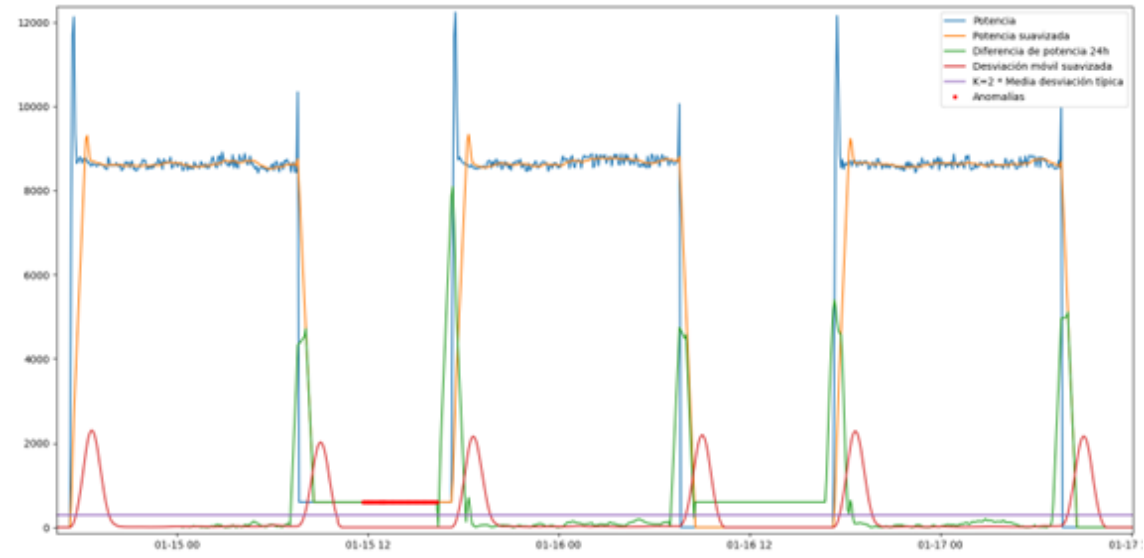

Figura 26 Ejemplo de detección de anomalías AT11.

En la Figura 27 se puede observar un ejemplo de falso positivo, pero si se analizan, se puede observar que hay luminarias donde los valores registrados de potencia no tienen un

comportamiento normal. Esto ocurre debido a que los propios datos contienen anomalías como pueden ser el de una luminaria fundida o un cambio en la propia línea de las luminarias que provoquen un incremento en el consumo de potencia.
Estos falsos positivos, realmente serían detecciones correctas de fallos OT.

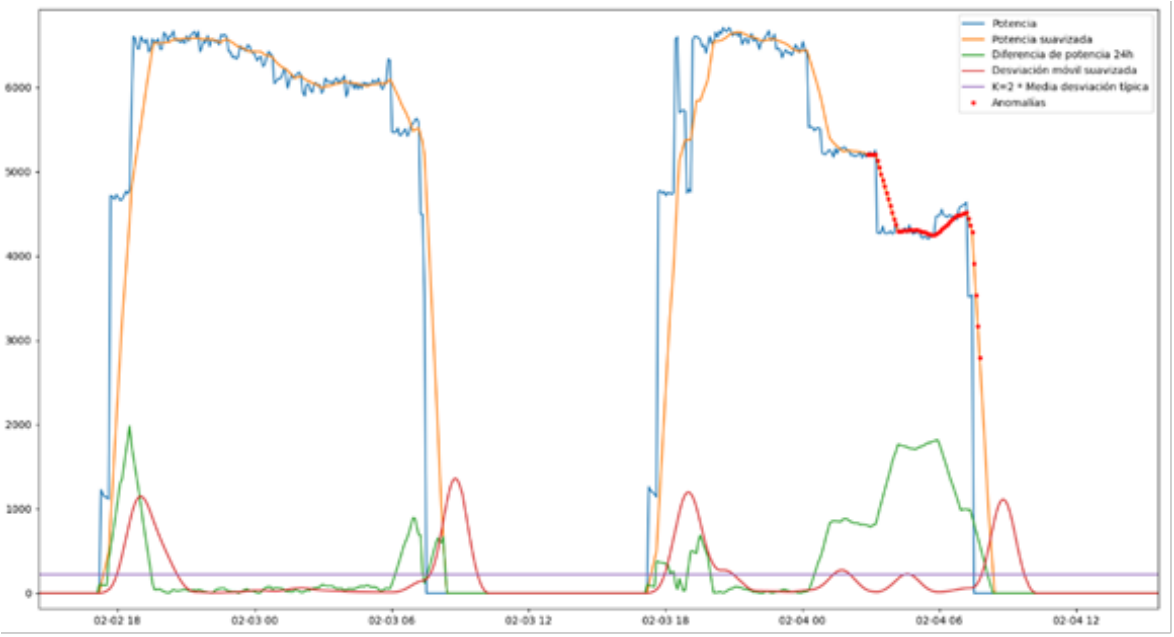

Figura 27 Ejemplo de falso positivo para el módulo de fallos OT.

4.2.3.3 Resultados para el módulo de detector de anomalías en tráfico de red

Para realizar una prueba de concepto sobre el módulo de detección de anomalías en tráfico de red, se han seleccionado aquellos registros IPFIX que corresponden con el tráfico del protocolo COAP, debido a que este es el tráfico que cursa la red IoT para transmitir los datos de aplicación obtenidos por las luminarias. Se han usado los últimos 7 días de tráfico, comprendido entre las fechas del 12 de Julio de 2021 al 19 de Julio de 2021. Teniendo un total de 153014 registros IPFIX, correspondiente a 70 luminarias diferentes.
Cada ataque de red implementado (AT1-5) se ha replicado en 100 instantes de tiempo de forma aleatoria, para comprobar si se detecta de forma correcta. En la Tabla 15 se muestra para cada uno de los ataques red implementados, las detecciones de anomalías en cada uno de los indicadores de compromiso de red.

Ataque	IoC1	IoC2	IoC3	IoC4	IoC5	IoC6	IoC7
AT1	0	0	76	0	100	100	100
AT2	0	0	76	93	100	100	100
AT3	0	100	76	60	100	100	100
AT4	0	0	76	80	100	100	100
AT5	100	0	70	60	100	100	100

Tabla 15 Detecciones correctas para cada uno de los IoC y ataques de red.

Por último, vamos se realiza una simulación sin inserción de ataques para obtener el número de falsos positivos que genera este módulo. En la Tabla 16 se puede observar el resultado de esta simulación.

ID	FP
IoC1	0
IoC2	0
IoC3	20
IoC4	8
IoC5	0
IoC6	0
IoC7	0

Tabla 16 Número de falsos positivos generados por el módulo de red.

Adicionalmente, si agrupamos los indicadores de compromiso que detectan anomalías para cada uno de los ataques, podríamos obtener unas reglas de correlación que permitiría agregar todas las alarmas en una única.

Ataque	Regla
AT1	IoC3+IoC5+IoC6+IoC7
AT2	IoC3+IoC4+IoC5+IoC6+IoC7
AT3	IoC2+IoC3+IoC4+IoC5+IoC6+IoC7
AT4	IoC3+IoC4+IoC5+IoC6+IoC7
AT5	IoC1+IoC3+IoC4+IoC5+IoC6+IoC7

Tabla 17 Posibles reglas de correlación para agrupar las alarmas de red.

Como se observa en la Tabla 17 los ataques AT2, AT3 y AT5 generan anomalías en indicadores de compromisos distintos y podríamos diferenciar perfectamente estos ataques del resto, pero los ataques AT2 y AT4 generan anomalías en los mismos indicadores. La única diferencia es que para el ataque AT2 la detección en el IoC4 funciona un 13% mejor que para el ataque AT4. Por lo que se podría generar una alarma dando una información extra donde se indicase la probabilidad de cada uno de los ataques.

5 CONCLUSIONES Y LÍNEAS DE AVANCE

La vida no es sino una continua sucesión de oportunidades para sobrevivir.
Gabriel García Márquez

En este último capítulo se detallan las conclusiones extraídas del sistema propuestos, para ello, se detalla de forma global los resultados obtenidos para cada uno de los componentes del sistema propuesto. Finalmente se extraen las posibles líneas futuras que presenta este trabajo.

Como se ha comentado al principio del documento, el objetivo era el de diseñar e implementar un sistema de detección de anomalías en entornos de SmartLighting que fuera completamente pasivo. Todas las técnicas utilizadas para la detección de anomalías cumplen con este requisito indispensable en este tipo de entornos donde no se desea interferir en la red IoT.
A continuación, se detalla las conclusiones extraídas para cada uno de los componentes del sistema:

- **Módulo de coherencia de variables:**
  - Tras realizar las pruebas, se puede afirmar que el control de residuos Q junto con la reducción de dimensiones PCA es una buena técnica para detectar anomalías en los datos registrados por las UCA. Encontrando un número óptimo de 4 componentes principales y un valor de K igual 6, obteniendo una tasa de falsos positivos del 2% y una capacidad de detección mayor del 90% para los 3 tipos de anomalías insertadas. Estos resultados se obtienen en 64 de los 68 dispositivos (94% de las UCA).
- **Módulo de fallos OT:**
  - Con los resultados obtenidos, se puede confirmar que el algoritmo propuesto obtiene un buen resultado con el parámetro de número de horas igual 3. Obteniendo un rendimiento del 89% para el caso de luminaria fundida y un 87% para el caso de robo de potencia. Con un porcentaje del menos del 4.7% de falsos positivos en ambos casos.
- **Módulo de detección de red:**
  - De todo el dataset de red generado, se ha seleccionado una fracción que contienen un total de 153014 flujos ipfix. Al realizar las pruebas se han obtenido un total de 28 falsos positivos (20 para el IoC3 y 8 para el IoC4) lo cual, corresponde con una tasa del 0.02% de falsos positivos sobre el total de los registros analizados. Si analizamos el funcionamiento de este módulo a la hora de detectar, vemos que detecta al 100% todos los ataques para los indicadores de compromiso IoC5-7 y obteniendo una tasa de mayor al 60% para el resto de los indicadores propuesto.

En base a los resultados obtenidos, en cuanto a detección y número de falsos positivos, se puede afirmar que el sistema propuesto es válido para detectar anomalías en entornos Smartlighting, tanto en la capa de red, como en la de aplicación. Además, como el sistema propuesto está diseñado en módulos independientes, sería compatible con sistemas de detección basados en firmas.
Una de las limitaciones principales que presenta este sistema, es en la parte de red, debido a que existen ataques que generan los mismos indicios y no sería posible saber a priori de que ataque se trata. Por lo que sería necesario realizar un análisis más profundo en las trazas ipfix generadas.

Este trabajo es una primera prueba de concepto y presenta diferentes líneas de futuro y mejores. A continuación, se mencionan algunas de las posibles líneas futuras:

- Implementación de más ataques tanto en el nivel de aplicación, como en el de red.
- Implementación de indicadores de red que permitan diferenciar entre todos los ataques posibles.
- Integración con un módulo de detección basado en firmas como puede ser Snort o Suricata.
- Integración con un sistema de correlación de alarmas que agrupe las alarmas generadas por el sistema y así reducir el número de falsos positivos.
- Como las luminarias tienen un comportamiento en función a las horas del sol, sería interesante diseñar e implementar un algoritmo de detección en los valores de potencia en función a un modelo solar.
- Obtener métricas de rendimiento de los algoritmos propuestos con intervalos de fechas más grande. Para ello, es necesario la obtención o generación de más datos de red y aplicación.
- Añadir un sistema de filtrado de datos, tanto en el nivel de red como en el nivel de aplicación. Esto conseguiría eliminar aquellos datos que presentan errores y hacen que los algoritmos no tengan un entrenamiento correcto.

REFERENCIAS

[1] E. Manavalan and K. Jayakrishna, "A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Comput Ind Eng*, vol. 127, no. November 2017, pp. 925–953, 2019, doi: 10.1016/j.cie.2018.11.030.

[2] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97, pp. 48–65, Nov. 2017, doi: 10.1016/J.JNCA.2017.08.017.

[3] S. A. Butt, J. L. Diaz-Martinez, T. Jamal, A. Ali, E. De-La-Hoz-Franco, and M. Shoaib, "IoT Smart Health Security Threats," *Proceedings - 2019 19th International Conference on Computational Science and Its Applications, ICCSA 2019*, pp. 26–31, 2019, doi: 10.1109/ICCSA.2019.000-8.

[4] T. hoon Kim, C. Ramos, and S. Mohammed, "Smart City and IoT," *Future Generation Computer Systems*, vol. 76, pp. 159–162, Nov. 2017, doi: 10.1016/J.FUTURE.2017.03.034.

[5] F. Hofer and B. Russo, *Architecture and Its Vulnerabilities in Smart-Lighting Systems*, vol. 1, no. 1. Association for Computing Machinery, 2023. doi: 10.1007/978-3-031-05516-4_10.

[6] H. Mora, J. Peral, A. Ferrandez, D. Gil, and J. Szymanski, "Distributed Architectures for Intensive Urban Computing: A Case Study on Smart Lighting for Sustainable Cities," *IEEE Access*, vol. 7, pp. 58449–58465, 2019, doi: 10.1109/ACCESS.2019.2914613.

[7] O. Khutsoane, B. Isong, and A. M. Abu-Mahfouz, "IoT devices and applications based on LoRa/LoRaWAN," *Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, vol. 2017-Janua, pp. 6107–6112, 2017, doi: 10.1109/IECON.2017.8217061.

[8] S. K. Routray, K. P. Sharmila, E. Akanskha, A. D. Ghosh, L. Sharma, and M. Pappa, "Narrowband IoT (NBIoT) for Smart Cities," 2021, doi: 10.1109/ICICV50876.2021.9388513.

[9] E. M. Martinez, P. Ponce, I. Macias, and A. Molina, "Automation pyramid as constructor for a complete digital twin, case study: A didactic manufacturing system," *Sensors*, vol. 21, no. 14, Jul. 2021, doi: 10.3390/s21144656.

[10] I. Lee, "Internet of Things (IoT) Cybersecurity: Literature Review and IoT Cyber Risk Management," *Future Internet 2020, Vol. 12, Page 157*, vol. 12, no. 9, p. 157, Sep. 2020, doi: 10.3390/FI12090157.

[11] A. Vetterl and R. Clayton, "Honware: A Virtual Honeypot Framework for Capturing CPE and IoT Zero Days," *eCrime Researchers Summit, eCrime*, vol. 2019-Novem, 2019, doi: 10.1109/eCrime47957.2019.9037501.

[12] A. Ahmed, R. Latif, S. Latif, H. Abbas, and F. Aslam Khan, "Malicious insiders attack in IoT based Multi-Cloud e-Healthcare environment: A Systematic Literature Review," *Multimed Tools Appl*, vol. 77, pp. 21947–21965, 2018, doi: 10.1007/s11042-017-5540-x.

[13] G. Ikrissi and T. Mazri, "IOT-BASED SMART ENVIRONMENTS: STATE of the ART, SECURITY THREATS and SOLUTIONS," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Dec. 2021, pp. 279–286. doi: 10.5194/isprs-Archives-XLVI-4-W5-2021-279-2021.

[14] C. C. Sobin, "A Survey on Architecture, Protocols and Challenges in IoT," vol. 112, pp. 1383–1429, 2020, doi: 10.1007/s11277-020-07108-5.

[15] E. Villa Crespo, *Ciberseguridad IoT y su aplicación en ciudades inteligentes*. Madrid: Ra-Ma, 2023.
[16] "An Overview and the Trends of Wireless Communications for IoT — Uniting Digital." https://www.unitingdigital.com/articles/2018/10/26/an-overview-and-the-trends-of-wireless-communications-for-iot (accessed Sep. 07, 2023).
[17] A. Osorio, M. Calle, J. D. Soto, and J. E. Candelo-Becerra, "Routing in LoRaWAN: Overview and Challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 72–76, Jun. 2020, doi: 10.1109/MCOM.001.2000053.
[18] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT".
[19] S. G. Varghese, C. P. Kurian, V. I. George, A. John, V. Nayak, and A. Upadhyay, "Comparative study of zigBee topologies for IoT-based lighting automation," *IET Wireless Sensor Systems*, vol. 9, no. 4, pp. 201–207, 2019, doi: 10.1049/iet-wss.2018.5065.
[20] A. Verma and V. Ranga, "Security of RPL Based 6LoWPAN Networks in the Internet of Things: A Review," *IEEE Sens J*, vol. 20, no. 11, pp. 5666–5690, 2020, doi: 10.1109/JSEN.2020.2973677.
[21] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, "RPL-Based Routing Protocols in IoT Applications: A Review," *IEEE Sens J*, vol. 19, no. 15, pp. 5952–5967, 2019, doi: 10.1109/JSEN.2019.2910881.
[22] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*, 2017, doi: 10.1109/SysEng.2017.8088251.
[23] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Towards a lightweight detection system for cyber attacks in the IoT environment using corresponding features," *Electronics (Switzerland)*, vol. 9, no. 1, pp. 1–19, 2020, doi: 10.3390/electronics9010144.
[24] A. N. Iman and T. Ahmad, "Improving Intrusion Detection System by Estimating Parameters of Random Forest in Boruta," *undefined*, Feb. 2020, doi: 10.1109/ICOSTA48221.2020.1570609975.
[25] M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry (Basel)*, vol. 11, no. 4, 2019, doi: 10.3390/sym11040583.
[26] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "electronics Article", doi: 10.3390/electronics8111210.
[27] V. Clincy and H. Shahriar, "Web Application Firewall: Network Security Models and Configuration," *Proceedings - International Computer Software and Applications Conference*, vol. 1, pp. 835–836, 2018, doi: 10.1109/COMPSAC.2018.00144.
[28] "GitHub - SpiderLabs/ModSecurity: ModSecurity is an open source, cross platform web application firewall (WAF) engine for Apache, IIS and Nginx that is developed by Trustwave's SpiderLabs. It has a robust event-based programming language which provides protection from a range of attacks against web applications and allows for HTTP traffic monitoring, logging and real-time analysis. With over 10,000 deployments world-wide, ModSecurity is the most widely deployed WAF in existence." https://github.com/SpiderLabs/ModSecurity (accessed Dec. 06, 2022).
[29] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," *2020 International Workshop on Big Data and Information Security, IWBIS 2020*, pp. 85–90, 2020, doi: 10.1109/IWBIS50925.2020.9255601.
[30] "OWASP ModSecurity Core Rule Set – The 1st Line of Defense Against Web Application Attacks." https://coreruleset.org/ (accessed Dec. 06, 2022).
[31] "Argos - An emulator for capturing zero-day attacks." https://www.few.vu.nl/argos/ (accessed Sep. 19, 2023).

[32] “QEMU.” https://www.qemu.org/ (accessed Sep. 19, 2023).
[33] “Honeyd.” https://www.honeyd.org/ (accessed Sep. 19, 2023).
[34] “GitHub - cowrie/cowrie: Cowrie SSH/Telnet Honeypot https://cowrie.readthedocs.io.” https://github.com/cowrie/cowrie (accessed Sep. 19, 2023).
[35] “The Leader in OT Cybersecurity Technology | Nozomi Networks.” https://www.nozominetworks.com/ (accessed Sep. 19, 2023).
[36] M. Kassim, A. R. Mahmud, M. Amirullah Ramli, and R. A. Rahman, “Network Analysis of Students’ Online Activities via Port mirroring Switch Port Analyzer,” *2022 12th IEEE Symposium on Computer Applications and Industrial Electronics, ISCAIE 2022*, pp. 49–54, 2022, doi: 10.1109/ISCAIE54458.2022.9794504.
[37] “Matrix - Enterprise | MITRE ATT&CK®.” https://attack.mitre.org/matrices/enterprise/ (accessed Sep. 13, 2023).