ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de un sistema inteligente IDS/IPS/IRS usando ML en una infraestructura IoT para agricultura de precisión.

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingenieros en Telemática

Presentado por:

Tomás Javier Guijo Farías
Gabriel Stéfano Villanueva Rosero

GUAYAQUIL - ECUADOR

Año: 2023

DECLARACIÓN EXPRESA

”Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Gabriel Villanueva y Tomas Guijo, damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Gabriel Stéfano Villanueva Rosero

Gabriel Stéfano Villanueva Rosero

Tomas Javier Guijo Farias

Tomas Javier Guijo Farias

EVALUADORES

Firmado electrónicamente por:
IGNACIO MARIN GARCIA

Ing. Ignacio Marin Garcia

PROFESOR DE LA MATERIA

Firmado electrónicamente por:
NESTOR XAVIER ARREAGA ALVARADO

Ing. Nestor Xavier Arreaga Alvarado

PROFESOR TUTOR

TABLA DE ABREVIATURAS

Acrónimo	Significado en Ingles	Significado en Español
ESPOL	–	Escuela Superior Politécnica del Litoral
NACE	National Association of Corrosion Engineer	Asociación Nacional de Ingenieros de Corrosion
SSC	Silver Electrode Silver Chloride	Electrodo de Plata Cloruro de Plata
CSE	Copper Sulfate Copper Electrode	Electrodo de Cobre Sulfato de Cobre
IoT	Internet of Things	Internet de las Cosas
IP	Internet Protocol	Protocolo de Internet
DDoS	Distributed Denial of Service	Denegación de Servicio Distribuida
ML	Machine Learning	Aprendizaje Automático
IPS	Intrusion Prevention System	Sistema de Prevención de Intrusos
IRS	Intrusion Restriction System	Sistema de Restricción de Intrusos
IDS	Intrusion Detection System	Sistema de Detección de Intrusos
GPS	Global Positioning System	Sistema Posicionamiento Global
IICA	–	Instituto Interamericano de Cooperación para la Agricultura
WAN	Wide Area Network	Red de Area Amplia
WiFi	Wireless Fidelity	Fidelidad Inalámbrica
LoRaWAN	Low Power Wide Area Network	Red de Area Amplia de Baja Potencia
PAN	Personal Area Network	Red de Area Personal
DoS	Denial of Service	Denegacion de Servicio
COTS	Commercial Off-The-Shelf	–

RESUMEN

La implementación de infraestructura del Internet de las Cosas (IoT por sus siglas en ingles Internet of Things) en la agricultura ha crecido en los últimos años como una opción viable para poder optimizar los recursos del cultivo. Mientras crece la popularidad de estos sistemas, también se vuelve popular explotar las vulnerabilidades de estos sistemas para extraer o modificar la información recolectada. En el presente proyecto se busca mejorar la infraestructura IoT para agricultura de precisión mediante la implementación de un sistema Prevención y Restricción de Intrusos que sea capaz de tomar acciones contra tráfico malicioso detectado en el nodo central de la red IoT. También se busca implementar la automatización de la actualización de los datos los cuales son alimentados al modelo de aprendizaje automático que detecta anomalías en los datos. Finalmente, para cumplir con los objetivos propuestos se realizó un programa en Python. Este programa, mediante el análisis de tráfico de red, es capaz de realizar bloqueo de paquetes detectados como maliciosos. También se desarrolló un programa en Python que actualice los datos para el modelo de anomalías. Este último programa se ejecutaría automáticamente de forma periódica mediante la configuración de un crontab.

Palabras Clave: Internet de las cosas, Agricultura, IPS, IRS, Python

ABSTRACT

The implementation of Internet of Things infrastructure (IoT, for its acronym in English, Internet of Things) in agriculture has grown in recent years as a viable option to optimize crop resources. As the popularity of these systems grows, it is also becoming popular to exploit vulnerabilities in these systems to extract or modify the collected information. This project seeks to improve the IoT infrastructure for precision agriculture by implementing an intruder prevention and restriction system that is capable of taking action against malicious traffic detected in the central node of the IoT network. It also seeks to implement the automation of data updates, which are fed to the machine learning model that detects anomalies in the data. Finally, to achieve the proposed objectives, a program was created in Python. This script, by analyzing network traffic, is capable of dropping packets detected as malicious. A Python program was also developed to update the data for the anomaly detection model. This last program would be automatically executed periodically by configuring a crontab.

Keywords: IoT, Agriculture, IPS, IRS, Python

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

La agricultura de precisión es definida como el uso adecuado de recursos en el momento y localidad necesarios. Esto se logra mediante la recolección de datos del cultivo en tiempo real mediante el uso de tecnologías como: Sistema de Posicionamiento Global (o sus siglas en ingles GPS), sensores e inteligencia artificial [1]. Sin embargo, por regla general todo dispositivo que se encuentre conectado a Internet, es sujeto a tener vulnerabilidades. En consecuencia de esto, surge la necesidad de implementar un sistema inteligente de detección, prevención y restricción de red. Este sistema utilizaría Machine Learning para poder detectar posibles ataques y anomalías en la red de la infraestructura utilizada para la agricultura de precisión.

1.1 Definición de Problemática

Actualmente existe una infraestructura para agricultura de precisión implementada en los laboratorios de sistemas telemáticos en la Escuela Superior Politécnica del Litoral (ESPOL) [2]. Esta infraestructura consta de dos nodos sensores capaces de medir temperatura, humedad y presión; y un nodo base capaz de enviar la data recolectada a una plataforma web. Sin embargo, se necesitan realizar algunas mejoras dentro de esta infraestructura. Para la implementaciónón actual se necesita eliminar el uso de cableado para la comunicación de datos entre los sensores y los nodos padres de la infraestructura Internet de las Cosas. También se precisa modificar el dataset usado por el algoritmo implementado de Machine Learning para el sistema de anomalías. La modificación consistiría en automatizar la actualización del dataset durante periodos de tiempos determinados y alimentar el algoritmo con esta información. Finalmente, se requiere la implementación de sistemas IPS/IRS para que se puedan realizar acciones

preventivas y post detección de un ataque a la infraestructura IoT.

1.2 Objetivos

En base a la problemática expuesta en la sección anterior, se propone el siguiente objetivo general: **Realizar una red en el cual se detecte, se prevenga y se restrinja el acceso de los usuarios de manera inteligente dentro de un sistema de agricultura de presición mediante el uso de sensores inteligentes con la aplicación de un dataset, el cual viene implementado con Machine Learning dentro de los Laboratorios de Telemática de ESPOL.**

Partiendo del objetivo general propuesto, se identifican los siguientes objetivos específicos:

- Automatizar la actualización del dataset utilizado por Machine Learning mediante un proceso que se ejecutará periódicamente, una vez por semana, en el nodo central para poder detectar anomalías correctamente en los datos censados por los nodos hijos.

- Implementar un sistema IPS/IRS en el microcontrolador usado como nodo central que sea capaz de tomar acciones en base a la detección de patrones y anomalías en el trafico de red entrante para la creación de medidas de restricción y bloqueo en tiempo real para evitar ataques como de Denegación Distribuida de Servicio (o por sus siglas en ingles DDoS).

1.3 Descripción de escenarios para la propuesta

Para implementar una mejora en la comunicación inalámbrica en la red, se propuso el uso de la tecnología Zigbee[1].El envío de la información recolectada por los nodos sensores hacia el nodo padre se propuso que se realice con una frecuencia igual con la que se envía los datos actualmente, cada 60 segundos.

Con respecto a la automatización para la actualización del dataset utilizado por el algoritmo de ML, se usó la utilidad crontab. Crontab pernite la creación de trabajos que

[1] Según [3], Zigbee es un conjunto de protocolos de comunicación inalámbrica de bajo costo que permite la creación de red de Área Personal (PAN).

se ejecutan periódicamente en sistemas Unix. La configuración de un proceso crontab en el nodo padre, fue capaz de ejecutar cada 7 días el script que genera el dataset usado por el algoritmo de ML.

Finalmente, se simuló ataques DoS [2] con la ayuda del mensaje SYN [3]para verificar el funcionamiento de los sistemas IPS e IRS a implementar. Una vez detectada la anomalía en la red, se deberá realizar la notificación de la misma. El sistema deberá ser capaz de tomar una acción para terminar el ataque. Estas acciones dependerán del tipo de anomalía detectada, pero pueden incluir bloqueo de puertos o IPs.

1.4 Estado del Arte

En la actualidad los sistemas inteligentes basados en redes IoT brindan la posibilidad de desplegar aplicaciones capaces de automatización en varias industrias. Parte de la propuesta original de este proyecto es el uso de una red de sensores IoT con la finalidad de mejorar la calidad de producción de un cultivo. El artículo ”Diseño e Implementación de un Sistema de Riego Inteligente basado en Sensores y Módulos de Radiofrecuencia para Transmisión y Sistema de Control” [4] ofrece una solución innovadora para el riego eficiente en la agricultura. Este tipo de tecnología podría ser de gran ayuda en la implementación del sistema inteligente IDS/IPS/IRS que fue desarrollada en esta tesis, ya que el uso de sensores y la transmisión de datos en tiempo real son elementos clave para mejorar la eficiencia de la agricultura de precisión. Por lo tanto, el conocimiento adquirido a partir del artículo puede ser aplicado en el desarrollo del sistema para mejorar la calidad y cantidad de los datos recolectados, lo que permitiría una mejor toma de decisiones y una mayor precisión en el monitoreo y control de la infraestructura IoT en la agricultura de precisión. Por otra parte el artículo ”Implementación de redes ad-hoc y pruebas experimentales utilizando componentes COTS y de bajo costo: un estudio de caso ecuatoriano” [5] presenta una solución económica y eficiente para la implementación de redes ad-hoc utilizando componentes de bajo costo. Este enfoque podría ser útil en la implementación del sistema inteligente IDS/IPS/IRS que se está desarrollando

[2]Por sus siglas en ingles Denial of Service o (DoS), según [6] es un tipo de ataque que genera una cantidad masiva de peticiones desde una maquina a una misma direccion IP.

[3]Segun [7] SYN, se define como un tipo de señal que se usa para la sincronización de números de secuencia, sin embargo también hace referencia a un ataque de denegación de servicio que aprovecha el mecanismo de negociación de tres vías del protocolo TCP.

en esta tesis, ya que la infraestructura IoT en la agricultura de precisión puede ser costosa. La implementación de redes ad-hoc utilizando componentes COTS y de bajo costo puede reducir significativamente los costos de la infraestructura de red y permitir una mejor conectividad y transmisión de datos en tiempo real. Por lo tanto, la información presentada en el artículo puede ser de gran utilidad en el desarrollo del sistema inteligente para mejorar la eficiencia de la infraestructura IoT en la agricultura de precisión..Los autores del articulo ”Review of agricultural iot technology” [8] clasifican a esta aplicación como Agricultura IoT. En la reseña, los autores establecen una arquitectura estándar de esta implementación. En primera instancia se tiene los sensores IoT, los cuales se implementan para poder medir diferentes factores importantes en la agricultura como: la temperatura, presión, humedad del suelo, precipitación, entre otras. La información recopilada es transmitida por la red inalámbrica establecida; generalmente se usan protocolos como Zigbee o LoRaWAN, aunque dependiendo de la implementación también es común encontrar protocolos como WiFi y redes WAN como 3G/4G. Finalmente, tenemos la capa de aplicación que suele ser una plataforma pública en la nube. En general, el uso de esta plataforma le permite al usuario ver en tiempo real las variaciones de los parámetros medidos o almacenar los mismos en una base de datos.

Como se mencionó durante la conferencia organizado por IICA [9], la combinación entre el conocimiento y las nuevas tecnologías darán paso en un contexto más amplio a la precisión. Adicionalmente, la definen como un conjunto de técnicas diseñadas para optimizar el uso de los recursos agrícolas de acuerdo a la variabilidad espacial, esto es los diferentes tipos de suelos y el tipo de clima que se tenga en el sector. Se determina que este tipo de agricultura lo que genera es en base a la información recolectada el productor deberá tomar las mejores decisiones para generar una mayor rentabilidad y contribuir a la sostenibilidad de los recursos naturales.

En el articulo antes mencionado [8] se expone que el objetivo de la recolección de información por la red IoT es poder procesarla para poder descubrir patrones. Estos patrones a su vez servirán para la mejora de calidad de producto, disminuir costos de trabajo manual, etc. La forma en la cual estos sistemas IoT pueden lograr este objetivo es por medio de la computación en la nube. Debido a que los dispositivos que conforman la red son de recursos limitados y la cantidad de información en tiempo real puede llegar a ser de gran volumen, el uso de la computación en la nube solventa estos obstáculos.

Mediante plataformas que ofertan servicios de computación en la nube, el sistema puede realizar análisis, almancenar y presentar la información recolectada.

Como se expone en el párrafo anterior, estos sistemas basados en redes IoT están relacionados íntimamente con el acceso a Internet para poder funcionar de forma óptima. La seguridad de las redes IoT es propensa a mal funcionamiento debido a que estas se componen de dispositivos con recursos limitados. También es un punto de ataque atractivo debido a que generalmente se configura información sensible en los dispositivos de estas redes como: credenciales de acceso a plataformas en la nube, o información sobre dispositivos físicos y virtuales que pueden ser accedidos por medio de Internet.

De acuerdo con el reporte de la institución Palo Alto Networks [10], la implementación de dispositivos IoT creció en una 21.50% entre 2018 y 2019. Se reporta también que el 57.00% de estos dispositivos son vulnerables a ataques. Una de las vulnerabilidades que se menciona en el reporte es el uso de estos dispositivos como un medio para poder identificar otros elementos en la red. Mediante el escaneo de direcciones IP y puertos, se puede identificar otros dispositivos que transmiten información privada. En la conferencia de ”IoT in Social, Mobile, Analytics and Cloud” [11] se categorizan los tipos de ataque IoT en: Físico, De Red, De Software y De Cifrado. Con respecto al tipo de ataque de red, en la reseña se identifica al ataque sinkhole como el más peligroso. Este tipo de ataque compromete un nodo de la red IoT logrando redirigir el tráfico hacia este. Una vez logrado esto se pueden realizar otras acciones mediante la manipulación de paquetes con la finalidad de comprometer la información de la red.

Al momento existe una gran cantidad de distintos tipos de ataques que se pueden aplicar a los dispositivos IoT, entre ellos se destacan los ataques de Denegación del servicio o por sus siglas DDoS. Adicionalmente, los cibercriminales incrementaron el uso de botnets lo cual permite ejecutar código malicioso y dañar la red de manera simultánea. Como indico el responsable de IoT en BGH Tech Partner Mario Fernandez [12], debido a la gran cantidad de dispositivos mal protegidos, las industrias se exponen a altos niveles de ser vulneradas. Complementando con la información presentada, según un informe presentado por la empresa Kaspersky [13], en el segundo trimestre de 2022 existieron una gran cantidad de ataques DDoS. Según la herramienta Kaspersky DDoS Intelligence se registro alrededor de 78558 ataques de esta naturaleza. Los mismos el 43.25% se encontraba en Estados Unidos. Como dato importante, el 62.53% fueron ataques de

desbordamiento de UDP.

Durante esta sección se ha conocido algunos términos importantes que tendrán relevancia durante el proyecto, sin embargo la pregunta mas importante a nivel general es cuanto puede costarle un ataque de este tipo de a tu empresa. Como se describe [14], los ataques DDoS interrumpen tiempo de actividad en las redes en cuales se esta provocando este tipo de problemas, lo que genera inactividad en las mismas y provoca perdidas millonarias. Dentro de las mas resaltables, se tiene que la perdida de la productividad afecta a las aplicaciones empresariales y dispositivos inteligentes ya que durante este periodo no se podrá acceder a los recursos de la aplicación, lo que reduce claramente la recolección de información y afecta a las ganancias.

Conforme a lo expuesto anteriormente, es imperativo implementar sistemas capaces de proteger la integridad de la red IoT. Es por eso que se propone el uso de sistemas IDS/IPS/IRS para poder identificar acciones maliciosas y tomar acciones adecuadas para su subyugación. De acuerdo con [15], el objetivo de un sistema IDS es detectar un ataque en la red. El sistema identifica un ataque monitoreando la red y comparando el comportamiento del tráfico contra una base de datos o contra un modelo de comportamiento. Una vez que el IDS detecta un evento, este es notificado o registrado. Los sistemas conocidos como IPS e IRS son sistemas que son encargados de realizar una acción para mitigar un ataque, una vez que haya sido detectado. Según [16], el IPS se lo asocia con acciones preventivas, mientras que el IRS con acciones para contrarrestar el ataque.

En la implementación original de este proyecto [2], el nodo central utiliza Machine Learning para detectar actividades maliciosas en la red. Una vez detectada, el sistema es capaz de realizar una notificación por correo sobre la anomalía hacia el administrador. La implementación de ML en el nodo central se la realizó mediante el uso de la tecnología TinyML.

Los autores del articulo Tiny machine learning, [17] definen TinyML como un subcampo de ML. TinyML tiene como objetivo implementar aplicativos de ML en sistemas embebidos de bajos recursos. Al igual que ML, se necesita de un dataset para poder alimentar el algoritmo de ML de la aplicación a implementar. De acuerdo con [18], el uso de TinyML supone una gran ventaja en sistemas de bajos recursos, especialmente en redes IoT. Esto se debe a que permite realizar procesamiento de datos dentro de la red y de esta

manera gana independencia del procesamiento en la nube, aumentando la seguridad de la red.

A pesar de que el IDS implementado en el proyecto es capaz de realizar su objetivo, no es capaz de tomar acciones sobre las anomalías detectadas. El IDS notifica al usuario, y depende del mismo decidir que acción tomar y cuándo se debería implementar. Por esta razón se propone implementar los sistemas IPS/IRS usando TinyML, para determinar los cursos de acción a seguir en base a las anomalías detectadas.

CAPÍTULO 2

2. Marco Teórico

Dentro de esta sección se tocarán temas y conceptos en los cuales se ha trabajado dentro del proyecto para su realización. De acuerdo con [19], la agricultura de precisión es el tipo de agricultura que mide las características de los cultivos, del suelo y factores climáticos para el mejoramiento a la hora de la productividad agrícola. Cabe mencionar, que esta estrategia implica tomar la mejor decisión posible, en el tiempo y lugar adecuado para generar una mayor rentabilidad.

Otro concepto que se debe conocer, es el término IoT. En palabras simples, se define como [20] la agrupación o la conexión de equipos y objetos a través de la red. Por ejemplo, un caso común puede ser un foco inteligente, el cual puede prenderse y apagarse a través de una aplicacion que a su vez se conecta a internet. También se tiene que conocer el cerebro del proyecto, el cual es un microcontrolador en cual se podrá realizar la programación de las funciones de la red inteligente.

Adicionalmente se utilizo la herramienta de Machine Learning, que según la empresa Cleverdata [21], es una de las disciplinas de la Inteligencia Artificial que crea sistemas que aprenden de manera automática. El mayor rédito de esta herramienta fue la capacidad de detectar patrones para verificar algún tipo de anomalía. Una de las partes importantes con respecto a la comunicación inalámbrica de los sensores, fue el modulo Zigbee. Según Digi[22], Es un modulo que viene presentado como una solución integrada para redes malladas eficientes en el consumo de energía y costos.

Materiales Como parte del hardware se utilizó un microcontrolador de la marca RaspberryPi, modelo 4. Este microcontrolador fue usado como nodo principal de la red, responsable de recibir la información de los nodos sensores. También tendrá el rol de enviar los datos recopilados hacia la plataforma Ubidots. La elección del modelo

RaspberryPi 4 está relacionado con la implementación de TinyML para los algoritmos de Machine Learning. Como ha sido expuesto en secciones pasadas, TinyML es una tecnología que tiene como enfoque aplicar ML en dispositivos de muy bajos recursos, tales como los microcontroladores. Sin embargo, el uso de un microcontrolador con recursos superiores, permitirá una mejor implementación de TinyML y los otros módulos que debe ejecutar el microcontrolador.

Con respecto al software, para la implementación de los sistemas IPS/IRS se utilizo TinyML. La librería a implementar es TensorFlow Lite del lenguaje de programación Python. El motivo por el cual se escogió TinyML y TensorFlow Lite fue por su facilidad de uso en microcontroladores. Como se explicó en el párrafo anterior, TinyML es de gran utilidad cuando se necesita ejecutar aplicaciones ML en sistemas de bajos recursos. El uso de Python supone también una mejor interoperabilidad con el sistema IDS que fue diseñado en el mismo idioma.

Como parte de mejoras a implementar en la estructura IoT que están fuera del alcance de los objetivos de este proyecto, se listará a continuación los materiales necesarios para estas mejoras. Para lograr la implementación de una red inalámbrica se hace uso de transceptores Zigbee. El uso del protocolo Zigbee se debe a su bajo consumo energético y su alcance entre 10 y 100 m.

2.1 Metodología

En esta sección se describe la metodología a implementar para el desarrollo de los sistemas IPS/IRS y la automatización de la actualización del dataset para el sistema IDS. La metodología que se uso dentro del proyecto es el Método Científico. Según el autor [23], el método científico permite colectar evidencia medible y empírica con la finalidad de sustentar una teoría.

Para esta metodología se identifican las siguientes etapas: Observación, Planteamiento del problema, Hipótesis, Experimentación, Análisis y Conclusión, las mismas que siguen el flujo mostrado en la Fig 2.1.

La razón por la cual se eligió esta metodología fue debido a que nos permitirá realizar experimentaciones de prueba y error. Este proceso permitirá probar la programación a realizar y comprobar los resultados obtenidos.

Figura 2.1: Metodología aplicada en el proyecto. [24]

En la etapa de Observación se pudo determinar el funcionamiento previo del proyecto ya implementado con sus sistemas, sin embargo cabe mencionar que se presentaron limitantes a la hora de la configuración de los dispositivos y se tuvo que realizar modificaciones en el código programado para que vuelva a funcionar. Adicionalmente se tuvo que cargar nuevamente el código dentro de los integrados ESP32 para su correcto funcionamiento a la hora del envió de datos. Siguiendo con la proposición, se determino cuales eran las claves para que el proyecto funcione de una manera adecuada y se adapte a las necesidades planteadas en los objetivos,

Debido a ello, en la etapa de Hipótesis; se definió una hipótesis en base a una investigación. Para esta etapa, se han realizado investigaciones de artículos científicos, en la Sección 1.4, sobre implementaciones actuales de seguridad para infraestructuras IoT en agricultura inteligente. A continuación definimos la hipótesis: al implementar políticas de bloqueo de tráfico como acción preventiva, se pueden mitigar los ataques en el tráfico de red IoT. Por ello, La implementación de un sistema de detección y prevención de intrusiones basado en dispositivos IoT y técnicas de Machine Learning, podría mejorar significativamente la seguridad en entornos de red, reduciendo el riesgo de ataques y aumentando la eficiencia en la detección y respuesta a amenazas cibernéticas. Esta

hipótesis se basa en la idea de que la combinación de dispositivos IoT (como sensores de detección de actividad en los cultivos) y técnicas de Machine Learning (como algoritmos de aprendizaje automático para la detección de patrones anómalos) podría mejorar la capacidad de un sistema de seguridad para identificar y responder a posibles amenazas. Además, la utilización de dispositivos IoT podría permitir una mejor monitorización y control de la red, lo que a su vez podría mejorar la eficiencia en la detección de incidentes y la toma de decisiones de respuesta. Es por medio de estas investigaciones que se justifica y se propone el uso de Aprendizaje Automático para el desarrollo de un sistema IPS/IRS. Finalmente, también se plantea mejorar el sistema IDS actualmente implementado.

De acuerdo con las investigaciones realizadas se propone utilizar el actual módulo sniffer implementado para obtener la información de tráfico de red necesaria para implementar políticas de bloqueo por medio del programa iptables. De esta forma, cuando el módulo IDS del nodo central detecte un posible ataque en la red, se implementará una regla en iptables para bloquear el tráfico malicioso detectado. Mediante la aplicación de estas políticas de bloqueo, se espera ralentizar el ataque de red, y disminuir la probabilidad de que el nodo central quede inoperativo e incapacitado a recibir la información de los nodos sensores.

Con respecto a la actualización de datos utilizados por el modelo de aprendizaje automático del módulo de detección de anomalías, se propone implementar un trabajo en el sistema operativo del nodo central. Para este proyecto, se plantea la creación un crontab con una frecuencia de siete días y que ejecute un script que actualice los datos medidos por los nodos sensores. Una vez que haya almacenado los datos, se pasaría a generar un nuevo modelo de aprendizaje. De esta forma, se garantizará que el modelo esté actualizado con datos recientes.

Cabe recalcar que para la programación del código fue necesario instalar varias librerías y bibliotecas que sirvieron para el funcionamiento del proyecto, dentro de las que se destacan ”os”, ”sklearn” y ”scapy”. Segun Python.Org [25] la librería ”os” permite interactuar con funciones del sistema operativo, principalmente aquellas que brindan información acerca del entorno del mismo. Por su parte la librería ”sklearn”, como menciona el economista y científico de data Andrés Torres [26], contiene muchas herramientas eficientes para el Machine Learning y modelo estadístico es por ello, que ofrece mucha flexibilidad a la hora de su importación y ejecución de ejemplos. Para la

librería ”scapy”, como lo indica Javier Gomez para la compañía Santander [27], es una librería que posee su propio interprete que permite crear, modificar, enviar y capturar paquetes de red. Una de sus principales funcionalidades es permitir escaneos y/o ataques de red.

Cabe indicar que a la hora del uso de ML existen diferentes tipos de aprendizaje que se pueden aplicar. Siguiendo la tendencia con respecto al proyecto anterior, se implementó el tipo de aprendizaje supervisado que según [28] es un método de análisis de datos que utiliza algoritmos que aprenden iterativamente de los datos para permitir que los ordenadores encuentren información escondida sin tener que programar de manera explícita dónde buscar. También, se mantuvo el algoritmo de Análisis de Discriminante Cuadrático, o por sus siglas en inglés QDA, para el entrenamiento del sistema IDS. Según [29] el QDA es un método de clasificación supervisado que se utiliza para clasificar datos en diferentes categorías o clases en función de las características o atributos de los datos de entrada.

Siguiendo con la implementación del sistema IDS actual que se basó en la detección de anomalías en la red, se utilizó el aprendizaje automático para detectar posibles ataques en el tráfico de red. Esto implicó utilizar un modelo previamente generado para detectar desviaciones en el comportamiento del tráfico de la red IoT y de esa manera predecir actividad maliciosa o sospechosa. En el presente proyecto el sistema IPS/IRS que se gestiona a través de Iptables, que es una herramienta de para aplicar políticas en el tráfico de red de un dispositivo. Adicionalmente como se baso tambien dentro de una libreria de python llamada ”sklearn”, esta libreria tambien permite la implemetacion del algoritmo de Isolation Forest para la detección de anomalías. Cabe mencionar segun [30] es un algoritmo de detección de anomalías que se basa en la construcción de árboles de decisión aleatorios para separar observaciones atípicas de las observaciones normales en un conjunto de datos.

Para poder automatizar la actualización de los datos por los cuales se crea el modelo usado en el módulo de anomalías, se hace uso de una configuración crontab. De acuerdo con [31] crontab es un comando en sistemas UNIX que permite crear un archivo crontab. Este archivo contendrá información de comandos que se deben ejecutar y la frecuencia con la cual deben ejecutarse. También se indica en [31] que un archivo crontab es un archivo que es leído por un demonio llamado cron. Este demonio es utilizado en

sistemas UNIX para programar actividades que deben ser ejecutadas una o varias veces. El demonio cron verifica cada minuto los tareas programadas y ejecuta esas tareas en segundo plano. Por medio de una configuración crontab nos aseguramos que el programa Python que actualiza la información usada para crear el modelo utilizado por el módulo de detección de anomálias, se mantenga actualizado.

En la etapa de Experimentación se emplearán simulaciones de ataques para poder medir la eficacia de la lógica implementada. Mediante los resultados de estas simulaciones se podrá obtener una retroalimentación para poder realizar ajustes a la lógica implementada, y de ser necesario, volver a ejecutar los escenarios de experimentación. Para ejecutar dichos ataques fue necesario generar una maquina virtual de Kali Linux y realizar un ataque interno dentro del segmento de red ya que actualmente el proyecto se encuentra conectado dentro de la red del Laboratorio de Telemática.

Como se indicó previamente, se realizarán pruebas de simulaciones de ataques DDoS antes y después de implementar el sistema IPS/IRS. Los resultados de estas pruebas permitirán realizar una comparación del uso de los recursos del sistema durante el ataque, y de esta forma se podrá comprobar si la implementación del sistema IPS/IRS representa una mejora contra la inanición y si permite que el sistema siga procesando los datos enviados por los nodos sensores de la red. En consiguiente se determino usar este esquemático para brindar un análisis simplificado de las funcionalidades del proyecto como se observa en la figura 2.2.

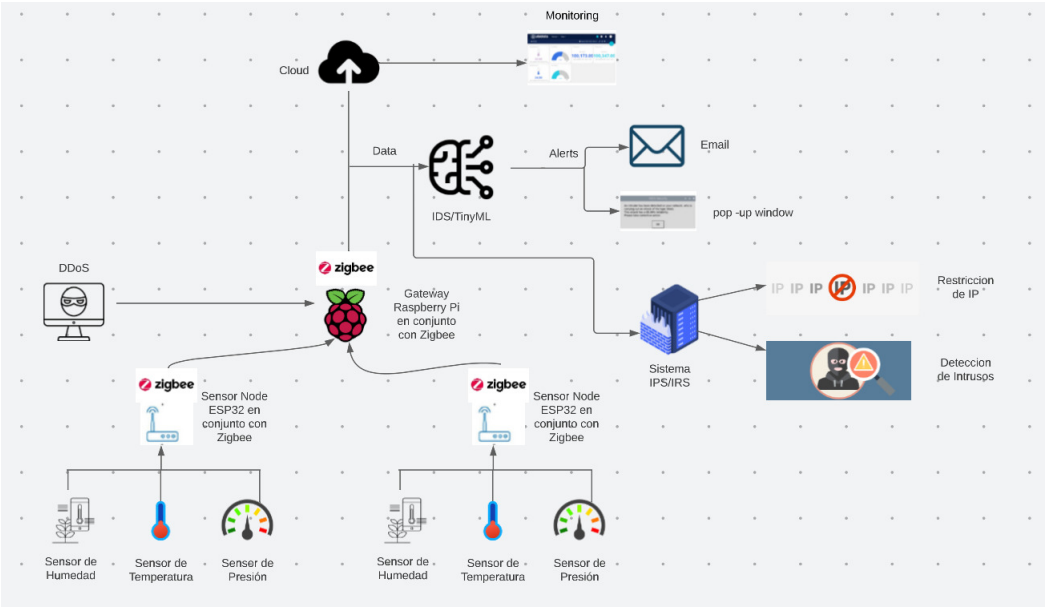

Figura 2.2: Esquemático usado para el proyecto.

2.2 Pruebas y Simulaciones

Inicialmente se determinó colocar la IP privada de manera estática para que el nodo principal, que es el microcontrolador, no tenga problemas si existiera un cambio no previsto en la configuración de la red inalámbrica a la que está conectado como se detalla en la Figura 2.3.

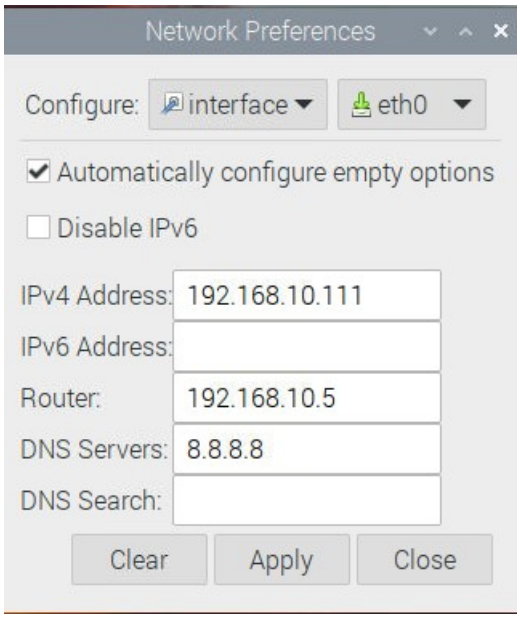

Figura 2.3: Configuración estática del microcontrolador.

Posterior a ello se realizó una prueba de conectividad para validar que la IP tuviera salida a Internet, como se puede observar en la Figura 2.3 y si se encontraba asignada la IP privada de manera correcta cuyo caso fue exitoso.Para realizar las pruebas pertinentes a la hora de la ejecución del proyecto, en primer lugar se tiene que validar la conexión de los sensores y el envió de la información hacia la plataforma de Ubidots mediante la suscripción hecha por MQTT para esto el tramado inicial se puede observar en la Figura 2.4

Adicionalmente se comprobó el envió de datos correspondientes, ejecutando el código tanto de MQTT como de Ubidots para realizar una suscripción a la plataforma y poder realizar el envió de datos censados a través de los sensores ubicados en la tierra. Cabe mencionar como se observa en la Figura 2.5 se puede verificar que los ”nodos hijos” se encuentran recolectando datos como humedad, temperatura y presión respectivamente por ello se concluyo que se encontraban funcionando de manera correcta. Dentro del código, se realizó inicialmente una suscripción en la primera instancia obteniendo

```
idscmiot@raspberrypi:~ $ ping  192.168.10.111
PING 192.168.10.111 (192.168.10.111) 56(84) bytes of data.
64 bytes from 192.168.10.111: icmp_seq=1 ttl=64 time=0.069 ms
64 bytes from 192.168.10.111: icmp_seq=2 ttl=64 time=0.071 ms
64 bytes from 192.168.10.111: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 192.168.10.111: icmp_seq=4 ttl=64 time=0.068 ms
64 bytes from 192.168.10.111: icmp_seq=5 ttl=64 time=0.067 ms
64 bytes from 192.168.10.111: icmp_seq=6 ttl=64 time=0.069 ms
64 bytes from 192.168.10.111: icmp_seq=7 ttl=64 time=0.066 ms
64 bytes from 192.168.10.111: icmp_seq=8 ttl=64 time=0.070 ms
64 bytes from 192.168.10.111: icmp_seq=9 ttl=64 time=0.069 ms
64 bytes from 192.168.10.111: icmp_seq=10 ttl=64 time=0.071 ms
64 bytes from 192.168.10.111: icmp_seq=11 ttl=64 time=0.072 ms
64 bytes from 192.168.10.111: icmp_seq=12 ttl=64 time=0.072 ms
64 bytes from 192.168.10.111: icmp_seq=13 ttl=64 time=0.072 ms
64 bytes from 192.168.10.111: icmp_seq=14 ttl=64 time=0.069 ms
64 bytes from 192.168.10.111: icmp_seq=15 ttl=64 time=0.075 ms
64 bytes from 192.168.10.111: icmp_seq=16 ttl=64 time=0.072 ms
64 bytes from 192.168.10.111: icmp_seq=17 ttl=64 time=0.072 ms
^C
--- 192.168.10.111 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16391ms
rtt min/avg/max/mdev = 0.066/0.070/0.075/0.002 ms
```

Figura 2.4: Prueba de conectividad a la IP Local Estática que se encuentra enlazada al microcontrolador.

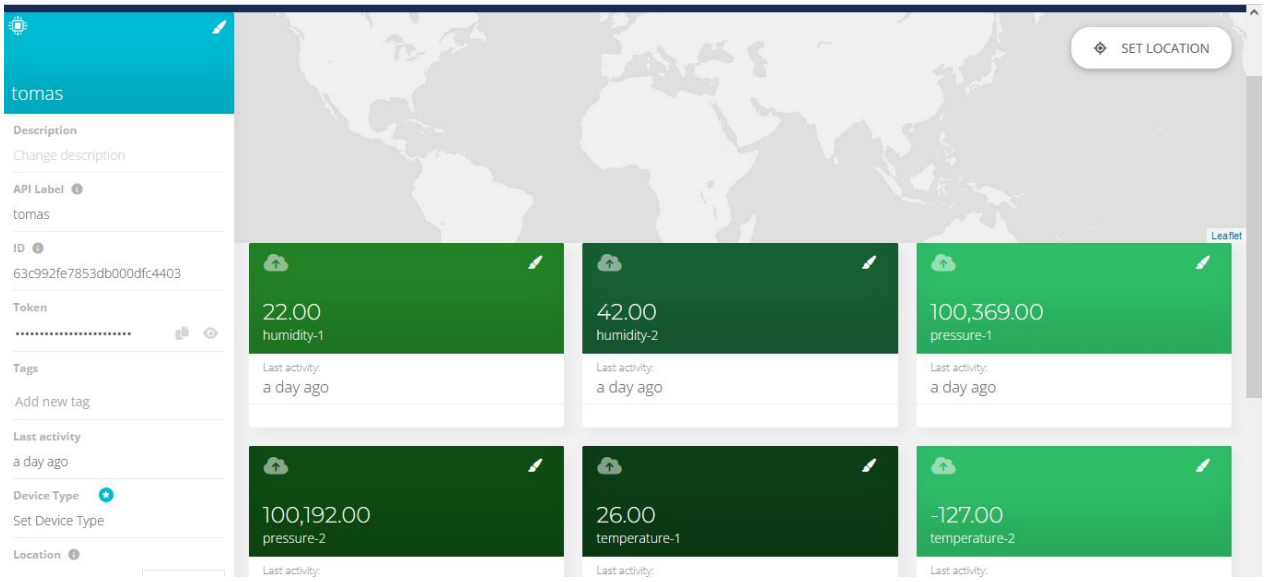

Figura 2.5: En la siguiente figura se puede observar los diferentes parámetros censados a través de los sensores enviados a Ubidots.

tanto la información del broker como de los nodos para poder registrar la información y posterior a ello recolectarla para su envió mas adelante. Siguiendo con las pruebas correspondientes, el código de Ubidots permite realizar el envió de datos a la plataforma para que se visualice en tiempo real como se muestra en la Figura 2.4 y se evidencia cualquier tipo de anomalía.

Cabe recalcar que se uso de la misma manera la suscripción con broker como se uso en MQTT con la diferencia que se realiza un envío constante de la información hacia

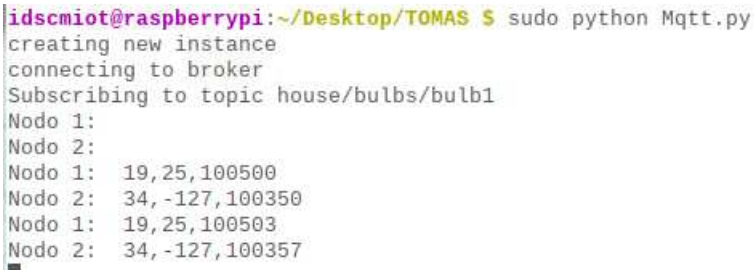

Figura 2.6: Prueba de código de MQTT.

Figura 2.7: Prueba de código de Ubidots.

la plataforma para la visualización de datos en tiempo real. Después de ello, se tuvo que programar el código de ”anomaly detector” debido a que el sensor se encuentra entregando datos aberrantes como se evidencia en la Figura 2.6 con respecto a la medición de la temperatura en el nodo N°2, se lo uso para brindar un ejemplo del mensaje de alerta que recibe el usuario a la hora que se presente algún tipo de problema dentro del sistema como refleja ya mas a detalle en la Figura 2.7. Pero para que se tenga claro que se consideren datos erróneos se realizo un ”entrenamiento previo” con una archivo de datos que se consideran normales para evitar cualquier tipo de inconveniente a la hora de ejecutar el programa por ello se genero la Figura 2.8.

En este caso debido a que se tuvo un dato aberrante dentro del sensor de Temperatura N°2 se envió un mensaje de alerta indicando que el valor de temperatura es muy bajo y no es sostenible para cultivar, esto se puede dar por problemas físicos dentro del sensor o existe un verdadero problema dentro de la zona donde se encuentra el sensor; sin embargo de igual manera se tiene una notificación constante del inconveniente presentado como se nota en la Figura 2.9. Cabe indicar que este mensaje es enviado en tiempo real, lo que significa que en caso de verificar algún problema al aplicar el

```
************************************************************
The next predicted value for the sequence Presion 2 [[[100323.]
  [100319.]
  [100319.]]] is

[[100493.08]]
************************************************************


Humedad 1     :  20
Humedad 2     :  34
Temperatura 1 :  25
Temperatura 2 :  -127
Presion 1     :  100475
Presion 2     :  100322
28
0029

An anomaly has been detected in Humedad 2
Signing in to Gmail
Sending Gmail
Mail Sent
[INFO] Attemping to send data to Ubidots
```

Figura 2.8: Prueba de código de Anomaly Detector.

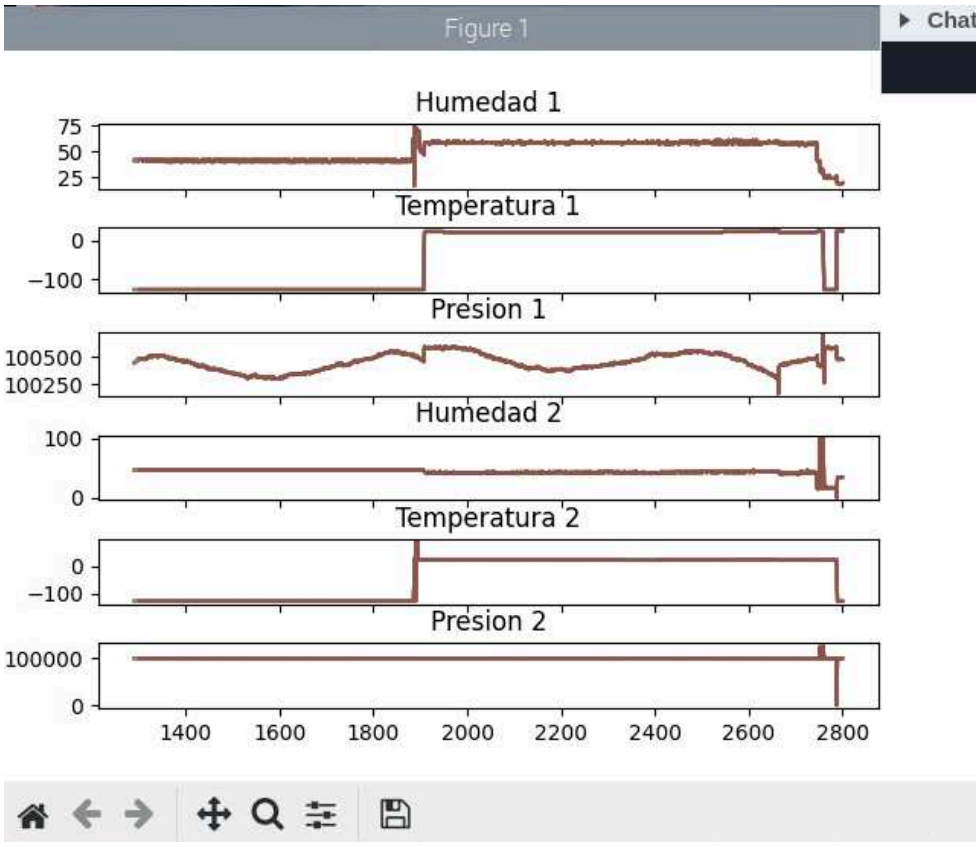

Figura 2.9: Gráfico generado al momento de realizar el entrenamiento de datos censados.

proyecto se tomaran acciones pertinentes y realizar una revisión total del funcionamiento del sensor.

Adicionalmente se procedió ejecutando el código de detección de intrusos realizando un escaneo constante del trafico de red que pasa por el nodo padre que en este caso

Figura 2.10: Mensaje de alerta con respecto a la Temperatura del Suelo.

```
[MAIN MODULE]: Module Status -
        NIDS Analyser Module: Running
        NIDS Sniffing Module: Running
        Ubidot Module: Running
[NIDS ANALYSER]: Loading the network label mappings data from the file system...
[NIDS ANALYSER]: >>> Loaded the network label mappings successfully.
[NIDS ANALYSER]: Loading the network packets data from the network mpnitoring sy
stem...
[NIDS ANALYSER]: >>> No network packets monitored yet for analysis.
[NIDS ANALYSER]: Initializing the prediction.
[NIDS ANALYSER]: >>> Prediction process completed.
[NIDS ANALYSER]: No attacks found in the monitored network packets.
[NIDS ANALYSER]: Sleeping for 30 seconds

[MAIN MODULE]: Module Status -
        NIDS Analyser Module: Running
        NIDS Sniffing Module: Running
        Ubidot Module: Running

[MAIN MODULE]: Module Status -
        NIDS Analyser Module: Running
        NIDS Sniffing Module: Running
        Ubidot Module: Running
```

Figura 2.11: Prueba de código IDS.

seria el microcontrolador como se muestra en la Figura 2.10 para evidenciar algún tipo de anomalía. En consiguiente de realizar las configuraciones y validaciones anteriormente planteadas, se propuso probar un código que trabaje con las librerías scapy, que según [27] menciona que es una de las herramientas mas importantes a la hora de la manipulación de paquetes. En este caso para el sistema IPS/IRS se ejecuto el código para que se analice el trafico de red que pasa por el nodo. Adicionalmente cabe recalcar que durante todo el momento se encuentra realizando un escaneo de todo lo que ocurre

dentro de la red, para justamente validar que tipo de paquetes se están enviando y detectar cualquier tipo de anomalía.

```
idscmiot@raspberrypi:~/Desktop $ sudo python ips.py
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65137
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55784
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65138
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55785
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65139
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55786
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65140
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55787
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65141
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55788
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65142
[+] Ataque detectado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55789
```

Figura 2.12: Prueba del código planteado para realizar un IPS.

Como se puede observar en la Figura 2.11 se realizo la prueba del código para el sistema de Prevención de Intrusos el cual ejecuta un escaneo constate dentro del trafico de red determinando la IP de origen y destino con su respectivo puerto de uso, cabe mencionar que únicamente hace un escaneo ya que en este punto no se habían definido la lógica a usarse en el sistema IRS.

Para la implementación del sistema de restricción de intrusos es necesario que se trabaje a la par con la herramienta de Iptables, ya que de esa manera se va a restringir los paquetes de protocolo de control de transmisión dependiendo si contienen una bandera de sincronización y acuse de recibido, lo que en palabras simples es un escaneo de ambos indicadores mencionados como se determino en la Figura 2.12. Si se detecta un ataque se utiliza la herramienta de Iptables para realizar el bloqueo correspondiente. La lógica que se planteo para posterior a la implementación, es la que al momento de estar conectado a la red privada y ser una red perteneciente a ESPOL las IPs que no pertenecen al mismo segmento de red, se procedió a bloquear mediante Iptables como se observa en la Figura 2.13.

En vista que se observo que tanto el sistema IPS como IRS funcionan de la mano en el código propuesto, se realizo una combinación para que ambos realicen tanto el escaneo como la restricción de algún tipo de trafico malicioso dentro de la red, adicionalmente se propuso que ya que estamos realizando un trabajo dentro una red interna, todo trafico que no pertenezca dentro del segmento de red, sera bloqueado mediante Iptables.

A la hora de la visualización de resultados con respecto a la Figura 2.14 se pudo observa que el código cumple su cometido, ya que se realiza un bloque de paquete a la

```
^Cidscmiot@raspberrypi:~/Desktop $ sudo python irs.py
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65145
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55792
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65146
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55793
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65147
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55794
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65148
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55795
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65149
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55796
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.211:65150
[+] Ataque bloqueado: SYN-ACK Scan desde 192.168.10.111:1883 hasta 192.168.10.229:55797
```

Figura 2.13: Prueba del código planteado para realizar un IRS.

```
IndexError: Layer [IP] not found
idscmiot@raspberrypi:~/Desktop $ sudo python tomas2.py
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 188.172.244.149
Paquete bloqueado desde dirección no permitida: 188.172.244.149
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.211
Paquete bloqueado desde dirección no permitida: 181.175.10.225
```

Figura 2.14: Prueba del código planteado para realizar un sistema IPS/IRS.

hora de que la IP de origen no pertenece a la IP interna de red, por ello no se envía el paquete al nodo padre y a la vez se esta previniendo de algún posible ataque de red a la hora por ejemplo de un ataque de denegación de servicio.

En la Figura 2.15 se pudo visualizar que las Ips internas tiene permitido el trafico de paquetes, en cambio las externas no se permitió el trafico de paquetes hacia el nodo padre debido a la lógica usada a través del código de Python para el sistema IDS/IPS/IRS. Posterior a ellos, de acuerdo con [32] la herramienta Hping3 permite probar sistemas IDS y Firewalls para verificar las técnicas utilizadas por estos sistemas para la detección de ataques en la red.

Las pruebas realizadas para verificar la implementación del sistema IPS/IRS involucraron utilizar un sistema Kali Linux que se encontrara en el mismo segmento de red que el microcontrolador Raspberry. Desde el sistema Kali Linux se utilizó la herramienta

```
idscmiot@raspberrypi:~/Desktop $ sudo python tomas2.py
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 188.172.244.149
Paquete bloqueado desde dirección no permitida: 188.172.244.149
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete permitido desde dirección permitida: 192.168.10.211
```

Figura 2.15: Prueba del código planteado para realizar un sistema IPS/IRS.

```
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 188.172.244.149/32 -j DROP
-A INPUT -s 188.172.244.149/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 157.240.197.60/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 188.172.244.149/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 188.172.244.136/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
```

Figura 2.16: Reglas de Iptables agregadas en base al código anterior.

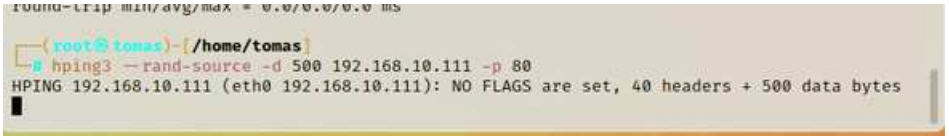

Figura 2.17: Herramienta hping3 usada para el ataque de la IP interna.

Hping3. Dentro de la misma, se realizo un ataque de denegación de servicio a la IP estática del microcontrolador, en este caso la ”192.168.10.111” para simular un ataque interno como se evidencia en la Figura 2.16.

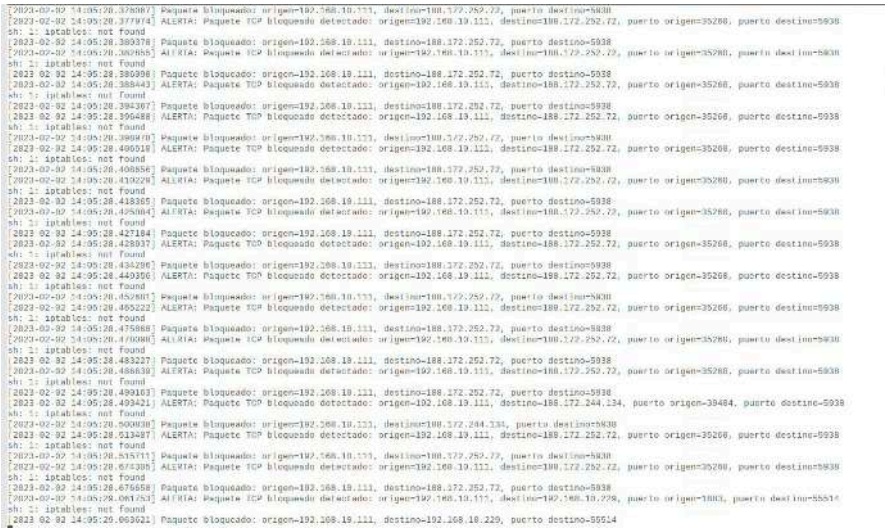

Figura 2.18: Captura del trafico de red durante el ataque.

Como dato importante, al momento que se realizo el ataque a través de la maquina virtual de Kali Linux por medio del código previamente empleado y en base al escaneo de trafico de red se pudo visualizar que los requerimientos y peticiones a la IP ”atacada” aumentaron considerablemente como se observa en la Figura 2.17, lo que a la larga provocaría ataques de Denegación de Servicio en caso de que no se apliquen políticas para mitigar este tipo de situaciones.

2.3 Resultados

De acuerdo con las pruebas detallas en la sección anterior, una vez implementado los sistemas IPS/IRS, los mismos fueron capaces de realizar un análisis de los paquetes detectados como tráfico malicioso. Una vez que se analizaron los paquetes, se procedió a implementar reglas de bloqueo de tráfico por medio del aplicativo Iptables. De acuerdo con las pruebas realizadas se obtuvieron los siguientes datos:

Prueba	Estado	Protocolo	Paquetes	
			Transmitidos	Rechazados
1	No Activo	TCP	106	0
2	Activo	TCP	54	54
3	Activo	TCP	73	73

Tabla 2.1: Tabla de resultados obtenidos

También se observo que a la hora de la implementación del código, todos los paquetes que no estaban dentro de la regla definida de Iptables se procedieron a bloquear para evitar cualquier tipo de trafico malicioso. Un claro ejemplo se puede observar en la Figura 2.18 la cual se detalla que las IPs que no pertenezcan dentro del mismo segmento en este caso la ”181.175.10.225” el paquete se bloqueara y guardara esa regla dentro de Iptables. Caso contrario lo que ocurre con respecto a la IP ”192.168.10.111” que el paquete si esta permitido.

```
idscmiot@raspberrypi:~/Desktop $ sudo python tomas2.py
Paquete bloqueado desde dirección no permitida: 181.175.10.225
Paquete permitido desde dirección permitida: 192.168.10.111
Paquete bloqueado desde dirección no permitida: 31.13.65.49
Paquete bloqueado desde dirección no permitida: 181.175.10.225
```

Figura 2.19: Ips bloqueadas y permitidas en base al codigo propuesto.

En este caso, se pudo observa que las reglas de Iptables se encuentran establecidas de manera correcta y el paquete que no pertenece a la regla se procede con el bloqueo del mismo esto se detalla claramente en la Figura 2.19 en los cuales se detallan todas las IPs en las cuales sus paquetes harán ”DROP”. Cabe mencionar que agregar dichas reglas a través de Iptables se ejecutaron de manera automática. Posterior a ello también

se realizo un análisis de los paquetes bloqueados y validando que tipo de IPs de origen y destino estaría orientado el envió o en un supuesto ataque como se observa en la Figura 2.20.

```
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 188.172.244.149/32 -j DROP
-A INPUT -s 188.172.244.149/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
-A INPUT -s 157.240.197.60/32 -j DROP
-A INPUT -s 181.175.10.225/32 -j DROP
-A INPUT -s 31.13.65.49/32 -j DROP
```

Figura 2.20: Paquete bloqueado proveniente de la ip 181.175.10.225.

```
target     prot opt source               destination
DROP       all  --  225.cpe-181-175-10.gye.satnet.net  anywhere
DROP       all  --  225.cpe-181-175-10.gye.satnet.net  anywhere
DROP       all  --  225.cpe-181-175-10.gye.satnet.net  anywhere
DROP       all  --  38.218.71.112        anywhere
DROP       all  --  230.63.160.39        anywhere
DROP       all  --  187.113.79.97.static.host.gvt.net.br  anywhere
DROP       all  --  whatsapp-cdn-shv-02-mia3.fbcdn.net  anywhere
DROP       all  --  cpe-71-72-205-61.cinci.res.rr.com  anywhere
DROP       all  --  210.179.218.126      anywhere
DROP       all  --  113.119.65.216       anywhere
DROP       all  --  58.101.190.219       anywhere
DROP       all  --  6.45.8.221           anywhere
DROP       all  --  IGLD-77-124-89-177.inter.net.il  anywhere
DROP       all  --  130.125.100.124      anywhere
DROP       all  --  112.14.63.248        anywhere
DROP       all  --  100.19.211.159       anywhere
DROP       all  --  209.230.29.76        anywhere
DROP       all  --  190.15.135.40        anywhere
DROP       all  --  196.79.177.212       anywhere
DROP       all  --  208.119.39.13        anywhere
DROP       all  --  149.101.68.185       anywhere
DROP       all  --  254.185.121.251      anywhere
```

Figura 2.21: Lista de paquetes bloqueados por parte de Iptables.

Por ultimo para validar las operaciones aplicadas también fue necesario aplicar un NMap, que según el blog KeepCoding [33] es un programa de código abierto que permite realizar escaneos de dispositivos conectados a la red, el servicio que ofrecen y a las vez los puertos que utilizan; cuyo objetivo dentro de este proyecto es descubrir algún tipo de vulnerabilidades. Dentro del análisis por parte de la herramienta descrita se observo que posterior a la implementación del código por el momento no se observo algún tipo de vulnerabilidades por el momento como se detalla en la Figura 2.21.

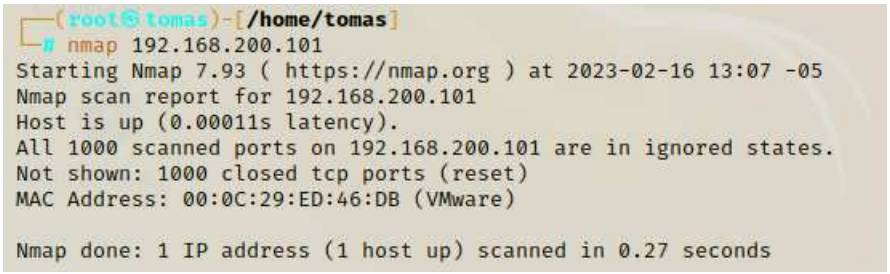

Figura 2.22: Resultado luego de validar el sistema con el NMAP.

CAPÍTULO 3

3. CONCLUSIONES Y LINEAS FUTURAS

Durante el desarrollo del proyecto se presentaron ciertos desafíos los cuales fueron exitosamente superados. Sin embargo, debido a que se trabajó dentro de un entorno protegido, ya que se usó la red de la Escuela Superior Politécnica del Litoral para llevar a cabo las pruebas pertinentes, surgieron ciertas limitantes a la hora de intentar realizar ataques externos al sistema inteligente IPS/IDS/IRS.

3.1 Conclusiones

En vista del desarrollo del proyecto se determinaron las siguientes conclusiones:

- El sistema IPS/IRS implementado es capaz de crear políticas de seguridad, de forma automática, para el bloqueo de tráfico entrante mediante la detección de anomalías usando Machine Learning, como se pueden observar en la Figura 2.19. Estas políticas permiten tomar acciones inmmediatas para mititgar posibles ataques en la red enfocas al nodo central de la infraestrucutra IoT, como se puede evidenciar en las Figuras 2.18 y 2.20.

- Segun [34], la tecnología Zigbee se presenta como una solución prometedora para el desarrollo de redes de sensores en la agricultura de precisión, ya que ofrece ventajas significativas en términos de bajo consumo de energía, alta capacidad de conectividad en malla y flexibilidad en la interoperabilidad con otros dispositivos y tecnologías. La implementación de redes de sensores basadas en Zigbee puede ayudar a mejorar la eficiencia y productividad de las actividades agrícolas, al proporcionar información precisa y en tiempo real sobre diversos parámetros del cultivo y del entorno, lo que permite una toma de decisiones más informada y precisa

como se puede observar a la hora del envió de la información a la plataforma de Ubidots con respecto a la Figura 2.4. A pesar de las ventajas mencionadas, no se pudo realizar la implementación del protocolo Zigbee en el presente proyecto. Al realizar las revisiones de la implementación actual del proyecto, se detectó un fallo del nodo central en la comunicación inalámbrica WiFi. Luego de validar que las configuraciones de red estuvieran correctas, no se pudo superar el incidente mencionado. Debido a la problemática mencionada, no se implementó el protocolo Zigbee ya que el problema se podía replicar ya que este puede radicar en el microcontrolador Raspberry.

- De acuerdo con la investigación de [35], la calidad de la información con la que se alimenta los algoritmos de Machine Learning, afecta la eficacia de estos y por lo tanto, afecta el modelo generado por ellos. Gracias a la actualización automática periódica del dataset para la detección de anomalías en la recolección de datos, se logró mejorar la exactitud de los datos que serán alimentados al algoritmo de Machine Learning. En resumen, la actualización automática periódica del dataset es crucial para asegurar la eficacia del modelo generado por el algoritmo de Machine Learning y, por lo tanto, para lograr una detección precisa de anomalías en los datos procesados por el nodo central.

3.2 Recomendaciones

Como se menciono previamente durante el desarrollo del proyecto se tuvieron ciertos desafíos a la hora de afrontar las configuraciones e implementaciones del código. En primer lugar, verificar si existe el envió de datos desde el microcontrolador hacia la plataforma de Ubidots y si los dispositivos ESP32 se encuentra recolectando los datos de manera correcta, caso contrario subir nuevamente el código a los mismos. Debido a que se trabajo dentro del entorno del Laboratorio de Telemática, validar si existe algún tipo de bloqueo con respecto a los puertos usados por Mqtt, y evitar colocar Sistema de Nombres de Dominio relacionados a ESPOL ya que no permite el envió de datos hacia Ubidots. Comprobar si todas las librerías se encuentran instalas y ejecutar los archivos de python en modo administrador a través de sudo. Cabe mencionar que Mqtt no permite realizar suscripciones en simultaneo, debido a ello ejecutar uno por uno el

código en cuestión. Trabajar los ataques a través de una maquina virtual con Kali Linux para poder realizar un ataque interno a la Ip del microcontrolador, ya que debido a que la Ip del Laboratorio pertenece a ESPOL no es posible realizar un ataque externo ya que se tendrá que vulnerar la seguridad de la institución. Al momento de descargar la libreria scapy, la cual se usa durante el sistema IPS/IRS realizar dicha descarga dentro del microcontrolador ya que se desarrolla mejor el programa y no se tiene inconvenientes a la hora de probarlo.

3.3 Líneas Futuras

Con respecto a las mejoras que pueden ser implementadas en trabajos futuros, se recomienda el uso de un token para la comunicación entre los nodos hijos y el nodo central de la red IoT. Esto con la finalidad de autenticar de forma eficiente si los datos recibidos han sido manipulados por una fuente externa. Otra implementación a realizar a futuro es el protocolo a implementar para la comunicación inalámbrica de la infraestructura. La implementación del protocolo LoRaWan implica el poder utilizar la infraestructura en un escenario físico más cercano a la realidad, en donde un cultivo puede abarcar varios kilometros de distancia. Finalmente, se sugiere implementar un repositorio donde se guarde un historial de los datos recopilados para el modelo de detección de anomalías y también las reglas de bloqueo que se van implementando por el sistema IPS, con la finalidad de poder confirmar el correcto funcionamiento de los diferentes módulos que se ejecutan en el nodo central de la red.

BIBLIOGRAFÍA

[1] D. O. Santillán and D. M. E. R. Rodríguez, "Agricultura de precisión," *INCYTU No. 015*, apr 2018.

[2] G. M. E. y Cristopher D. Vega, "Desarrollo de un sistema inteligente de detección de intrusos en una infraestructura iot para agricultura de precisión," sep 2022.

[3] fgarcia, "□ descubre qué es zigbee y para qué se utiliza." https://www.efectoled.com/blog/es/que-es-zigbee/, Apr 2018.

[4] M. Arcentales, V. Calero Bravo, and I. Marin-Garcia, *Diseño e Implementación de un Sistema de Riego Inteligente basado en Sensores y Módulos de Radiofrecuencia para Transmisión y Sistema de Control*. PhD thesis, 11 2013.

[5] P. Chavez-Burbano, I. Marin-Garcia, and A. Muñoz-Arcentales, "Ad-hoc network implementation and experimental testing using low cost and cots components: An ecuatorian case study," in *3rd IEEE International Work-Conference on Bioinspired Intelligence*, pp. 133–137, 2014.

[6] OSI, "¿qué son los ataques dos y ddos? | oficina de seguridad del internauta." https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos, Aug 2018.

[7] C. L. Jurado, "Qué es un ataque de inundación syn." https://es.ccm.net/aplicaciones-e-internet/museo-de-internet/enciclopedia/10618-que-es-un-ataque-de-inundacion-syn/, Mar 2023.

[8] J. Xu, B. Gu, and G. Tian, "Review of agricultural iot technology," *Artificial Intelligence in Agriculture*, vol. 6, pp. 10–22, 2022.

[9] IICA, “La agricultura de precisión tiene potencial para transformar el agro.” https://www.iica.int/es/prensa/noticias/la-agricultura-de-precisi%25C3%25B3n-tiene-potencial-para-transformar-el-agro, 2016.

[10] P. A. Networks, “2020 unit 42 iot threat report,” *IoT Business News*, pp. 1–22, 2020.

[11] J. Deogirikar and A. Vidhate, “Security attacks in iot: A survey,” in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 32–37, 2017.

[12] www.onussistemas.com.ar, “6 de cada 10 dispositivos de internet son vulnerables a ataques.” https://santafecanal.com.ar/4149-6-de-cada-10-dispositivos-de-internet-son-vulnerables-a-ataques, 08 2022.

[13] SecureList, “Los ataques ddos en el segundo trimestre de 2022.” https://securelist.lat/ddos-attacks-in-q2-2022/96933/, 08 2022.

[14] B. Sethunathan, “Vuelven las amenazas ddos | blog de softwareone.” https://www.softwareone.com/es-co/blog/articles/2022/05/30/vuelven-las-amenazas-ddos, 2022.

[15] M. Karanfilovska, T. Kochovska, Z. Todorov, A. Cholakoska, G. Jakimovski, and D. Efnusheva, “Analysis and modelling of a ml-based nids for iot networks,” *Procedia Computer Science*, vol. 204, pp. 187–195, 2022. International Conference on Industry Sciences and Computer Science Innovation.

[16] J. Networks, “What is ids and ips? | juniper networks.” https://www.juniper.net/us/en/research-topics/what-is-ids-ips.html, 04 2021.

[17] P. Kunz, “Tiny machine learning: A new technique for ai security.” https://ercim-news.ercim.eu/en129/special/tiny-machine-learning-a-new-technique-for-ai-security, 04 2021.

[18] J. Hertz, “What is tinyml? - technical articles.” https://www.allaboutcircuits.com/technical-articles/what-is-tinyml/, Jan 2022.

[19] BBVA, “¿qué es la agricultura de precisión? la gestión digital del campo.” https://www.bbva.com/es/sostenibilidad/que-es-la-agricultura-de-precision-la-gestion-digital-del-campo/, 08 2021.

[20] D. España, “¿qué es iot (internet of things)?.” https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html, 07 2021.

[21] Cleverdata and A. Gonzalez, “¿qué es machine learning? – cleverdata.” https://cleverdata.io/que-es-machine-learning-big-data/, 2019.

[22] DiGi, “¿qué es xbee? xbee.cl - comunicación inalámbrica para tus proyectos.” https://xbee.cl/que-es-xbee/, 08 2021.

[23] A. Bradford and A. Hamer, “What is science?.” https://www.livescience.com/20896-science-scientific-method.html, 08 2017.

[24] VIRESA, “9 beneficios que te aporta el método científico.” https://viresa.com.mx/blog_9_beneficios_metodo_cientifico, Jan 2022.

[25] P. Org, “10. pequeño paseo por la biblioteca estándar.” https://docs.python.org/es/3/tutorial/stdlib.html, 11 2021.

[26] A. Torres and freeCodeCamp, “Aprendizaje automático en python: Las principales características nuevas de scikit-learn 0.24 que debes saber..” https://www.freecodecamp.org/espanol/news/aprendizaje-automatico-en-python-las-principales-caracteristicas-nuevas-de-scikit
10 2021.

[27] J. Gomez and S. D. Services, “Guía: Uso de scapy con python.” https://santandergto.com/guia-uso-de-scapy-con-python/, 07 2020.

[28] TIBCO, “¿qué es el aprendizaje supervisado?.” https://www.tibco.com/es/reference-center/what-is-supervised-learning, Aug 2021.

[29] C. Gil Martínez, “AnÁlisis discriminante lineal y cuadrÁtico.” https://rstudio-pubs-static.s3.amazonaws.com/389151_3cfc2588daff4989b0ce8da8b3d5ab01.html, May 2018.

[30] J. A. Rodrigo, “Detección de anomalías: Isolation forest.” https://www.cienciadedatos.net/documentos/66_deteccion_anomalias_isolationforest.html, May 2020.

[31] U. I. T. S. . I. University, “What are cron and crontab, and how do i use them?,” jan 2018.

[32] A. Techradix, “What is hping3 and how to use it -techradix technology,” apr 2019.

[33] R. KeepCoding, “¿para qué sirve nmap? | keepcoding tech school.” https://keepcoding.io/blog/para-que-sirve-nmap/, Sep 2022.

[34] A. Rodríguez, M. Santander, and P. Pizarro, “Diseño e implementación de una red de sensores inalámbricos para monitoreo de variables ambientales en agricultura de precisión.” https://repository.unicatolica.edu.co/bitstream/handle/20.500.12237/806/FUCLG0016629.pdf?sequence=1, 2018.

[35] V. Sessions and M. Valtorta, “The effects of data quality on machine learning algorithms.,” pp. 485–498, 01 2006.

APÉNDICES

Apéndice A: Repositorios

- Repoaitorio del Proyecto: https://github.com/tguijo/IDS-IPS-IRS