

PROYECTO DE TITULACIÓN
[SISTEMA DE ANÁLISIS DE TRÁFICO DE RED]

PARA OBTENER EL TÍTULO DE
*INGENIERA EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIONES.*

PRESENTA:

YATZIRI AMPARO ESQUIVEL CRUZ

ASESOR:

BENITO RODRÍGUEZ CABRERA

Mayo



ÍNDICE

PRELIMINARES.....	4
1. Agradecimientos.....	4
2. Resumen.....	5
Índice de Tablas.....	6
Índice de Figuras.....	6
CAPÍTULO 1: GENERALIDADES DEL PROYECTO.....	7
3.- Introducción	7
4. Descripción de la empresa u organización y del puesto o área del trabajo del residente.	7
4.1 Objetivos.....	8
4.2 Misión	8
4.3 Visión.....	8
4.4 Principales actividades de la empresa u organización	9
5. Problemas a resolver, priorizándolos.	12
6. Justificación	13
7. Objetivos.....	13
7.1 Objetivo general	13
7.2 Objetivos específicos.....	13
CAPÍTULO 2: MARCO TEÓRICO	15
8. Marco Teórico.	15
8.1 Red LAN	16
8.2 Red WLAN	17
8.3 Tráfico de red.....	20
8.4 Redes neuronales	21
8.5 Python.....	22
8.6 Keras	23
8.7 Tensorflow	23
CAPÍTULO 3: DESARROLLO	24
9. Procedimiento y descripción de las actividades realizadas.	24
9.1 Metodología	24
CAPÍTULO 4: RESULTADOS.....	32
10. Resultados obtenidos	32
10.1 Selección de herramientas.....	32

10.3 Refinamiento del modelo y producto de ingeniería.....	37
CAPÍTULO 5: CONCLUSIONES	40
11. Conclusiones del Proyecto.....	40
CAPÍTULO 6: COMPETENCIAS DESARROLLADAS	41
12. Competencias desarrolladas y/o aplicadas.....	41
CAPÍTULO 7: FUENTES DE INFORMACIÓN	42
13. Fuentes de información	42

PRELIMINARES

1. Agradecimientos.

A Dios, por guiarme a tomar el mejor camino, por darme salud, sabiduría, paciencia y entendimiento para lograr cada una de mis metas.

A mis padres, que son mi motor de vida, por nunca dejarme sola, brindándome su apoyo incondicional y darme ánimo para lograr mis metas, por felicitarme en cada uno de mis logros, por levantarme cuando me va mal y decirme que se puede intentar una vez más.

A mis hermanos y familia por motivarme y confiar en mí, por darme siempre palabras de aliento en los momentos más difíciles.

A mis amigos por nunca dejarme sola, motivarme siempre y ayudarme durante mi etapa estudiantil, por estar en los momentos felices y tristes de mi vida y por hacerme ver que la amistad sí existe.

Gracias a todas y cada una de las personas que directa o indirectamente estuvieron conmigo en el proceso de mi carrera universitaria.

2. Resumen.

Detectar el tráfico de red mediante monitoreo es de suma importancia, ya que se pueden detectar interrupciones que causan cuellos de botella, así como posibles problemas futuros en la red. Gracias a la detección, tales problemas se pueden corregir sin que esto afecte en mayor cantidad el flujo de la red.

El presente documento muestra el desarrollo de un modelo para la detección de tráfico en la red, basado en redes neuronales. Los datos que fueron utilizados en el entrenamiento de la red neuronal se obtuvieron de la red WLAN que forma parte de LIA. También se utilizaron librerías que ayudaron al manejo de grandes cantidades de datos obtenidos de la red, así como librerías para graficar resultados.

Con todos los elementos tecnológicos recabados se entrenó el modelo de red neuronal, realizando iteraciones hasta la obtención del producto requerido.

Los resultados obtenidos muestran la precisión de las redes neuronales en el modelamiento de análisis de tráfico de red WLAN, gracias al suficiente número de muestras de tráfico obtenidas de la red.

Índice de Tablas

Tabla 1. Tabla de clasificación de redes. Fuente: (Lederkremer Miguel, 2019).	15
Tabla 2. Protocolos de red.....	18
Tabla 3. Clasificación de redes neuronales. Fuente: (Rivas et al., 2022).....	21

Índice de Figuras

Figura 1. Infraestructura LIA. Fuente: LIA.	8
Figura 2. Organigrama. Fuente: LIA.	10
Figura 3. Metodología Prototipo y KDD. Fuente: Autor.	25
Figura 4. Datos cargados para entrenamiento de modelo. Fuente: Autor.	27
Figura 5. Selección de los datos. Fuente: Autor.	27
Figura 6. Secuencias de entrenamiento. Fuente: Autor.	28
Figura 7. Modelo secuencias. Fuente: Autor.	28
Figura 8. Código para el entrenamiento del modelo. Fuente: Autor.	29
Figura 9. Obtención de MAE. Fuente: Autor.....	30
Figura 10. Detección de anomalías. Fuente: Autor.	30
Figura 11. Importación de librerías en código.....	32
Figura 12. Datos del archivo de test. Fuente: Autor.	33
Figura 13. Datos del archivo train. Fuente: Autor.	33
Figura 14. Resumen de modelo creado. Fuente: Autor.	34
Figura 15. Entrenamiento del modelo. Fuente: Autor.	35
Figura 16. Gráfica de muestra de margen de error. Fuente: Autor.....	35
Figura 17. Resultado MAE y reconstrucción train. Fuente: Autor.	36
Figura 18. Resultado MAE y reconstrucción test. Fuente: Autor.	36
Figura 19. Resultado de detección de anomalía. Fuente: Autor.....	37
Figura 20. Cambios al modelo. Fuente: Autor.	38
Figura 21. Gráfica de muestra de margen de error con cambios. Fuente: Autor.....	38
Figura 22. Resultado de detección de anomalía mejorada. Fuente: Autor.....	39

CAPÍTULO 1: GENERALIDADES DEL PROYECTO

Para conocer la finalidad de desarrollo de este proyecto es primordial analizar acerca de sus objetivos y el área donde se lleva a cabo, es por ello que en este capítulo se dan a conocer los aspectos de entorno de desarrollo del proyecto, en los cuales se abordan temas relacionados a los datos de la empresa y el puesto que ocupa el proyecto. Se describen también puntos importantes, tales como problemas a resolver, justificación y objetivos para una mejor comprensión del proyecto y su desarrollo.

3.- Introducción

El propósito de este proyecto es poder analizar el tráfico de red para la detección de anomalías típicas y atípicas que surgen en el envío y recepción de paquetes a través de la red de internet.

La creación y uso de sistemas inteligentes para resolver problemas cotidianos, se ha convertido en una herramienta indispensable y de mucha utilidad para representar con exactitud el comportamiento de lo que analiza o procesa dicho sistema.

Las redes neuronales son utilizadas como mecanismos de aprendizajes en diferentes modelos. Así mismo, son aplicadas a la predicción de tráfico en redes, con la finalidad de poder controlar la congestión, consumo excesivo de ancho de banda y predicción de anomalías.

En la siguiente documentación se muestra la información necesaria para poder comprender el proyecto, desde marco teórico con los conceptos básicos hasta el desarrollo de la red neuronal para el sistema.

4. Descripción de la empresa u organización y del puesto o área del trabajo del residente.

Las residencias profesionales se realizaron en el Tecnológico Nacional de México campus Pabellón de Arteaga, fundado el 01 de septiembre de 2008, ubicado en Carretera a la Estación de Rincón km 1, C.P.20670 Pabellón de Arteaga, del estado de Aguascalientes; en el Laboratorio de Iluminación Artificial (LIA).

Durante el 2017 se creó el Laboratorio de Iluminación Artificial (LIA) dentro del Tecnológico Nacional de México campus Pabellón de Arteaga, gracias al apoyo recibido por parte de la Convocatoria de Infraestructura Científica y Tecnológica del CONACyT (INFRA-2016-01, Project No. 270665). Por su parte, el equipamiento en LIA (ver Figura 1) se ha consolidado a través de diversos apoyos federales y estatales (CB-2016-01, Project No. 287828, IDSCEA, SADER, por mencionar algunos).



Figura 1. Infraestructura LIA. Fuente: LIA.

4.1 Objetivos

Profundizar y clarificar los fenómenos no explicados hasta el momento sobre los efectos de la luz (continua y pulsada) en plantas y alimentos como su caracterización biológica con herramientas tecnológicas actuales LEDs, siendo un centro capaz de poseer el nivel de competitividad que otros laboratorios existentes en el mundo.

4.2 Misión

Proyectar e impulsar el desarrollo de nuevas actividades en el sector agroindustrial a través de procedimientos que ayuden a mejorar la calidad de vida alrededor del mundo, generando productos agroalimentarios con valor agregado.

4.3 Visión

Ser un centro capaz de impulsar, desarrollar y fortalecer iniciativas que incentiven la productividad y rentabilidad agroalimentaria; generando en el sector agroindustrial nuevas filosofías de competitividad y calidad.

4.4 Principales actividades de la empresa u organización

Las herramientas tecnológicas encontradas en LIA permiten que el laboratorio tenga impacto a nivel internacional. Se cuentan con cuatro proyectos de investigación y desarrollo tecnológico enfocados a la línea de investigación de Biosistemas Mecatrónicos y Ambientes Artificiales.

- Diseño, construcción e implementación de sistemas de producción multinivel en espacios cerrados con luz artificial LED, que permiten potenciar el contenido nutrimental de cultivos como microgreens, lechuga, lenteja, espinaca, albahaca, alfalfa, entre otras, a través de recetas de luz (diferentes combinaciones de color).
- Desarrollo de Sistemas de desinfección de alimentos con radiación ultravioleta LED tipo A, B y C aplicados a productos agroindustriales y en fresco.
- Implementar estrategias tecnológicas con luz artificial tipo LED para preservar, extender y/o acelerar el tiempo de vida en almacén de frutas y verduras, además de analizar el efecto sobre la biosíntesis de compuestos (licopeno, capsaicina, entre otros.) durante su estancia en anaquel.
- Investigación, integración y aplicación de la agricultura de precisión con el uso de tecnología aérea no tripulada (VANTs) para el monitoreo y detección de plagas en diferentes cultivos con el objetivo de evitar pérdidas en la producción.

El equipo de trabajo que conforma al Laboratorio LIA se conforma por seis áreas las cuales son coordinadores, investigadores, doctorado, maestría, tesis y auxiliares; a continuación, se muestra el organigrama con las personas pertenecientes a cada área.

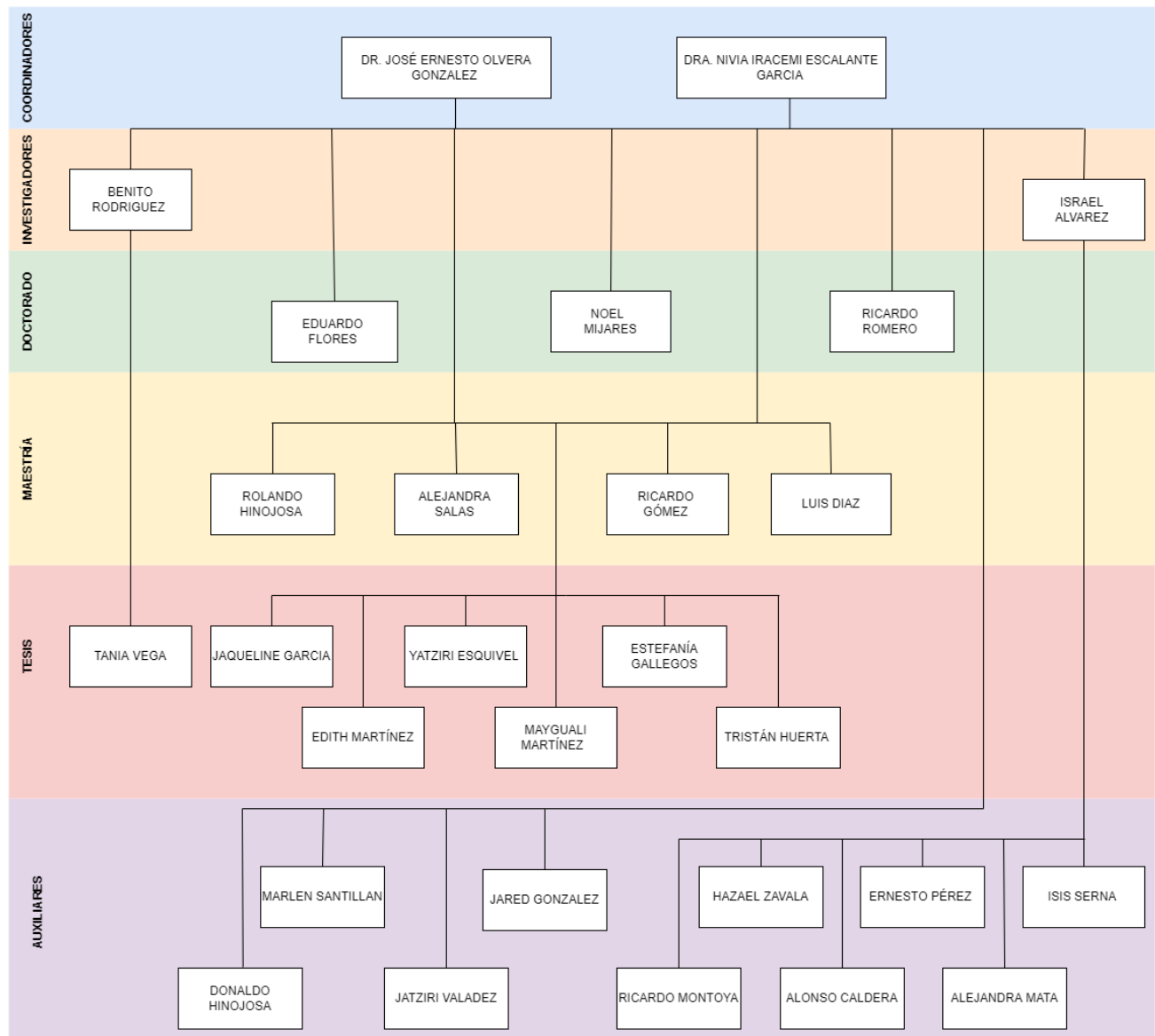


Figura 2. Organigrama. Fuente: LIA.

Cabe destacar que LIA es el único Laboratorio en el País que se ha estado especializando en el uso de la luz artificial, esto para el desarrollo de proyectos e investigaciones de gran impacto en el sector agroindustrial. Su infraestructura se compone de la siguiente manera:

- En LIA se cuenta con 3 cámaras de crecimiento (descripción Temperatura 12°C a 30°C. Iluminación de área de trabajo. Humedad: Ambiente +5% a 90%). Programación día/noche de temperatura.
- Racks y accesorios para producción de cultivos comerciales, sistema de caracterización de la luz (reflexión, absorción y transmisión) que incluye dos espectrofotómetros Ocean, esfera de medición de esparcimiento de luz e integradora (0.5 a 1 m) con accesorios para la manipulación de objetos en el interior.
- Un espectro radiómetro ILT950, flurómetro WALZ, SPAD 502, sensores de radiación fotosintéticamente activa, sensores climatológicos (humedad, temperatura, CO₂ y radiación), Controladores Lógicos Programables, Sistema de radiación artificial tipo LED RGBW (rojo, azul, verde y blanco) de 0 a 1000 µmol (Micromol) de intensidad, con canal programable independiente para el color.
- Frecuencia de pulsado de 1 a 500 kHz. Con software y control independiente.
- Sistemas de radiación artificial equipados con UV-LED e Infrarrojo.

En LIA es de suma importancia la implementación y uso de las TICS debido a la orientación de los proyectos desarrollados. Es por ello que el puesto que ocupó el residente se encuentra en el área de Desarrollo Tecnológico del laboratorio principalmente para el área de red.

5. Problemas a resolver, priorizándolos.

El Laboratorio de Iluminación Artificial (LIA) cuenta con un extenso equipo de investigadores y desarrolladores, los proyectos que allí se realizan tales como, sistemas de producción multinivel en espacios cerrados con luz artificial LED, sistemas de desinfección de alimentos con radiación ultravioleta LED, tecnologías con luz artificial tipo LED para preservar, extender y/o acelerar el tiempo de vida en almacén de frutas y verduras, investigación, integración y aplicación de la agricultura de precisión con el uso de tecnología aérea no tripulada (VANTs); necesitan principalmente el servicio de red para su correcto funcionamiento.

Debido a la cantidad de personas que hacen uso del servicio y el tipo de proyectos que se realizan, se requiere una red estable y con disponibilidad durante las jornadas laborales, lo que ocasiona problemas en el funcionamiento de la misma. El sobre envío de datos por parte del equipo de trabajo genera picos de tráfico de red considerables, causando problemas como pérdidas de tiempo en las actividades que se desarrollan en el momento, tales como, descargas de archivos incompletos, respuesta tardía a las peticiones en la web, cuellos de botella y sobre envío de paquetes. Es por ello que se requiere un sistema para analizar los problemas que suceden en la red en instantes de tiempo específicos, para que con los análisis se pueda determinar una solución óptima y así poder evitar estancamientos en el desarrollo de los proyectos.

6. Justificación

El análisis del tráfico de red permite visualizar en tiempo real lo que sucede durante el uso de la red, así mismo, permite mostrar el comportamiento de los patrones de acuerdo al flujo de los paquetes. La serie de problemas que se presentan en la red de LIA requieren de un análisis en tiempo real, debido a la disponibilidad que se necesita para el desarrollo de los proyectos.

Para los administradores de red resulta un tanto lento solucionar el tráfico que la red presenta a consecuencia del sobre envío de paquetes generado en distintos espacios físicos, además de que no cuentan con una herramienta que permita indicar con exactitud el problema. Es por ello que se requiere un sistema de análisis de tráfico de red para poder detectar en menor tiempo posible el dilema y así poder dar una óptima solución, al igual que el sistema mostrará las estadísticas del funcionamiento de la red en un tiempo específico.

7. Objetivos

7.1 Objetivo general

Desarrollar una plataforma web donde se muestren los indicadores del estado del tráfico de red por medio de Machine Learning y redes neuronales, a fin de conocer la situación que dispone la red dentro del Laboratorio de Iluminación Artificial (LIA).

7.2 Objetivos específicos

- Analizar el tráfico de red de espacios físicos.
- Concentrar la información obtenida en una base de datos para su posterior análisis.
- Analizar la información con el fin de detectar patrones y sus consecuencias para poder dar una óptima solución a problemas derivados.
- Diseñar una plataforma web que permita visualizar la información analizada de manera gráfica y tener una mejor interpretación de los datos.

- Desarrollar la plataforma web utilizando las herramientas ReactJS, NodeJs y MySQL a fin de cumplir los requerimientos establecidos para el diagnóstico de red.
- Integrar el Frontend y Backend para el despliegue y evaluación del funcionamiento de la plataforma completa.

CAPÍTULO 2: MARCO TEÓRICO

A través de este marco teórico se dan a conocer los conceptos generales requeridos para la comprensión del desarrollo del proyecto. Es importante conocer cada una de las herramientas que se utilizaron para el cumplimiento del proyecto, así como sus fundamentos técnicos que describen la relación que existe entre cada una.

8. Marco Teórico.

La digitalización y difusión del conocimiento y comunicación, tienen un gran impacto dentro de la red, así como un gran alcance. Por su parte la red es una infraestructura de aprendizaje, ya que dentro de ella se puede encontrar todo tipo de información y cualquier cosa que se desee. De acuerdo con su infraestructura, esta trabaja con un conjunto de dispositivos interconectados a través de un medio, allí surge la compartición de recursos e intercambio de información.

De acuerdo con la clasificación de redes cada una funge para un área específica en consideración a su tamaño. Tales redes se clasifican de la siguiente manera:

Tabla 1. Tabla de clasificación de redes. Fuente: (Lederkremer Miguel, 2019).

Red	Descripción
PAN (Personal Area Network).	Su estructura de alcance tiene un rango de pocos metros, debido a que se conforma solo con dispositivos de una persona.
WPAN (Wireless Personal Area Network).	Es una red inalámbrica, ya que utiliza este tipo de tecnologías como medio y es de área personal.
LAN (Local Area Network).	Es una red de área local, su rango de alcance se limita a un área pequeña debido a que no integra medios de uso público.

WLAN (Wireless Local Area Network).	Es una red LAN que maneja medios inalámbricos de comunicación. Es una de las más utilizada, ya que no requiere cableados y es más escalable.
CAN (Campus Area Network).	Es una red apta para dispositivos de alta velocidad, tiene la capacidad de poder conectar redes LAN en un área geográfica ilimitada ya que no utiliza medios públicos.
MAN (Metropolitan Area Network).	Esta red tiene una banda ancha de alta velocidad y cubre un área más extensa que una red CAN pero limitada.
WAN (Wide Area Network).	Es una red de área amplia, emplea medios de comunicación como satélites, cables interoceánicos, fibra óptica, entre otros, gracias a ello se extiende sobre un área geográfica muy extensa.
VLAN (Virtual LAN).	Es una red LAN virtual, los dispositivos que la componen se ubican en espacios geográficos alejados.

Con base en la clasificación y descripción de las redes que se muestra en la Tabla 1, la red optima a analizar dependiendo el área en el que se encuentra el Laboratorio de Iluminación Artificial (LIA), es la red de tipo LAN y WLAN ya que en la institución son las más utilizadas.

8.1 Red LAN

Una red LAN (Local Area Network) se encarga de conectar uno o más equipos informáticos, a fin de compartir recursos e intercambiar información. Por lo regular se encuentran en áreas pequeñas como oficinas, centros educativos o espacios personales,

dado a que cubren una área reducida de kilómetros de extensión, su tamaño es restringido ya que su tecnología de difusión se da mediante un cableado sencillo al cual se conectan todas las máquinas (Andres et al., 2022).

Dentro de los componentes de la red LAN se encuentran:

- Tarjetas de red.
- Cables de red.
- Concentrador de red.
- Conmutadores de red.
- Enrutadores.

8.2 Red WLAN

Wireless Local Area Network, se encarga de manejar las redes inalámbricas, a diferencia de las redes LAN su medio de transmisión es el aire, se utiliza comúnmente en espacios como hogares, centros educativos y oficinas. Gracias a esta red se puede conectar sin la necesidad de cableado, permitiendo a los usuarios conectarse en cualquier momento desde sus dispositivos móviles, logrando la movilidad dentro de los espacios en los que se usa (Cuellar Duque Yeferson, 2021).

La conexión a redes es indispensable para llevar a cabo la comunicación, así mismo para la transmisión de datos e intercambio de paquetes en redes los protocolos de red son los que ayudan a definir las características de la conexión.

Los protocolos de red tienen mecanismos para que se pueda establecer conexión e identidad entre los dispositivos. Además de reglas de formato donde se especifica la formación de los paquetes y los datos en los mensajes enviados y recibidos. Algunos de los tipos de protocolos de red que serán analizados se muestran a continuación:

Tabla 2. *Protocolos de red.*

Protocolo	Descripción
TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet)	Representa las reglas que ayudan a la conexión de computadoras de marcas y tecnologías diferentes. Por ende, define cómo se mueve la información desde el remitente hasta el destinatario (Estrada Adrián, 2022).
Protocolo IP (Protocolo de Internet)	Programa de afiliación responsable de enviar y coordinar los paquetes entre máquinas (has). No reproduce una sesión antes de intercambiar información. El avance autorizado funciona sin transmisión segura y debido a ello en la trayectoria un paquete se puede extraviar, cambiar reuniones, duplicar, conceder o incluso cortar (Guzmán Y Valle et al., 2018).
ARP (Protocolo de Resolución de Direcciones)	Forma parte de las capas de TCP/IP y se encarga de obtener las ubicaciones de los equipos de máquinas TCP/IP. ARP utiliza una comunicación cercana de la dirección IP para encontrar la dirección de la máquina o puerta de objetivo (Guzmán Y Valle et al., 2018).
TCP (Protocolo de control de transmisión)	Es utilizado como un protocolo host a host entre los miembros de redes de comunicación de computadoras por medio del intercambio de paquetes y en un

	<p>sistema interconectado en dichas redes. Está orientado a la conexión entre dos extremos especialmente trabaja en una jerarquía de capas de protocolos en aplicaciones de múltiples redes (Oviedo et al., 2019).</p>
<p>UDP (Client Datagram Protocol)</p>	<p>Es una administración de envío de datagramas sin certificación de transporte. Este pertenece a la familia de protocolos de Internet, por ello debe clasificarse en el nivel de transporte, por ende, se ubica en una capa intermedia entre la capa de red y la capa de aplicación (Guzmán Y Valle et al., 2018).</p>
<p>HTTP (Protocolo de transferencia de hipertexto)</p>	<p>Protocolo de comunicación para las transferencias de información en la World Wide Web. Este protocolo no guarda información sobre conexiones anteriores, es decir, se considera un protocolo sin estado (Oviedo et al., 2019).</p>
<p>HTTPS (Protocolo seguro de transferencia de hipertexto)</p>	<p>Es un protocolo que hace posible la comunicación entre el cliente web y el servidor web. Evita que terceros no autorizados tengan acceso a los datos (IONOS, 2022b).</p>
<p>SMTP (Protocolo simple de transferencia de correo)</p>	<p>Es un protocolo utilizado para el envío de correos electrónicos por medio de un servidor de correo (De Luz Sergio, 2022).</p>

ICMP (Internet Control Message Protocol)	Sus funciones son: la comunicación de errores en la entrega y reenvío de paquetes IP, la notificación de cambio de ruta y la medición de rendimiento. Este protocolo permite a las pasarelas y los sistemas principales enviar informes de problemas a la máquina que envía un paquete (De Jaén et al., 2020).
--	--

El análisis de protocolos permite conocer información relacionada con el tráfico que se presenta en la red de internet, ya que de acuerdo a cada una de las características de los protocolos se adecuan para la identificación de dicho problema presentado, así mismo el tráfico de red será encontrado a través del análisis de protocolos tanto de capa de red, transporte y aplicación ya que son los más encontrados por la red.

8.3 Tráfico de red

El tráfico de red se caracteriza por los datos que se desplazan a través de una red en determinado momento. Se conoce como tráfico de red o tráfico de datos, ya que durante la transmisión de los paquetes pueden surgir anomalías que irrumpen dicha acción, ocasionando pérdidas de paquetes o tiempo tardío de respuesta (Solarwinds, 2022).

Detectar el tráfico de red sirve principalmente para encontrar y solucionar problemas de manera rápida, así como también se puede tener un mayor control y visibilidad sobre el uso que le dan a la red los usuarios que se encuentran conectados.

El tráfico de red se puede detectar a través de sistemas inteligentes que determinan las anomalías típicas y atípicas que se presentan en la transmisión de los paquetes en la red de internet. Dichos sistemas trabajan con la ayuda de redes neuronales artificiales, las cuales son entrenadas con datos obtenidos del área correspondiente a analizar.

8.4 Redes neuronales

Una red neuronal es un método de la IA (Inteligencia Artificial), su principal característica es enseñar a las computadoras a procesar datos basándose en la forma en la que el cerebro humano lo hace. Es un proceso de Machine Learning conocido como aprendizaje profundo, que hace uso de las neuronas interconectadas en una estructura de capas que se asemejan al cerebro humano. Con las redes neuronales las computadoras crean un sistema el cual utilizan para aprender de los errores y mejorar continuamente (aws, 2022).

Algunas de las áreas de aplicación en las que intervienen las redes neuronales son:

- Visión artificial.
- Reconocimiento de voz.
- Procesamiento de lenguaje natural.
- Motores de recomendaciones.
- Procesamiento de imágenes.
- Detección de anomalías.

La clasificación de las redes neuronales se realiza a partir de diferentes tipos de redes, es decir, están clasificadas por el número de capas, por el tipo de conexiones y por el grado de conexiones. En la Tabla 3 se describe cada una de las redes neuronales con su respectiva clasificación.

Tabla 3. Clasificación de redes neuronales. Fuente: (Rivas et al., 2022).

Clasificación	Tipos de redes neuronales
Clasificación por el número de capas.	Redes neuronales monocapas: consta de una sola capa y su función es proyectar la capa de entrada a una capa de salida en la que se realizan los cálculos necesarios.
	Redes neuronales multicapas: es una generalización de las redes monocapas, donde existe un conjunto de capas

	intermedias entre la entrada y la salida (capas ocultas).
Clasificación por los tipos de conexiones.	Redes no recurrentes: no existe retroalimentación de ningún tipo debido a que solo se mueven hacia adelante.
	Redes neuronales recurrentes: permiten la retroalimentación a través del uso de lazos, dichos lazos pueden ser neuronas de capas distintas o de la misma.
Clasificación por grado de conexiones.	Redes neuronales totalmente conectadas: se conectan todas las neuronas con las capas anteriores o siguientes, creando así redes concurrentes o no concurrentes.
	Redes parcialmente conectadas: no se llega a una conexión total, sino parcial, ya sea en paralelo o de manera jerárquica.

Existen herramientas que ayudan a lograr el desarrollo de las redes neuronales, dentro de ellas se encuentran lenguajes de programación, bibliotecas, frameworks y librerías, por mencionar algunos.

La inclusión de estas herramientas es con el propósito de facilitar el desarrollo del Machine Learning, así como la optimización del tiempo en que son creadas.

A continuación, se describe el lenguaje de programación, biblioteca y framework utilizados en este proyecto.

8.5 Python

Es un lenguaje de programación utilizado en distintas áreas como en las aplicaciones web, desarrollo de software, ciencia de datos y Machine Learning (ML).

Dentro de las características que lo destacan son:

- Lenguaje interpretado.
- Fácil de utilizar.
- Lenguaje de alto nivel.
- Lenguaje orientado a objetos.

Este lenguaje contiene bibliotecas que ayudan a los desarrolladores a facilitar la estructura de código, ya que dentro de estas bibliotecas se encuentran colecciones de código que se utilizan en diferentes programas. Las bibliotecas más populares de Python son: Matplotlib, Pandas, Numpy, Requests, OpenCV-Python y Keras (aws, 2022).

Para consultar información sobre las bibliotecas de Python véase [Matplotlib](#), [Pandas](#), [Numpy](#), [Requests](#), [OpenCV-Python](#) y [Keras](#).

8.6 Keras

Keras es una biblioteca de código abierto que tienen como objetivo acelerar la creación de redes neuronales. Funciona como una Interfaz de Uso Intuitivo (API) que tiene acceso a diferentes frameworks de aprendizaje automático. Tales frameworks son Theano, Microsoft Cognitive Toolkit y Tensorflow (IONOS, 2022).

Para consultar información específica sobre los frameworks véase [Theano](#), [Microsoft Cognitive Toolkit](#) y [Tensorflow](#).

8.7 Tensorflow

Es una librería desarrollada por Google con la finalidad de satisfacer las necesidades a partir de redes neuronales artificiales. Está diseñada para construir y entrenar redes neuronales para detectar patrones y razonamientos usados por los humanos (Larkin, 2022).

CAPÍTULO 3: DESARROLLO

9. Procedimiento y descripción de las actividades realizadas.

9.1 Metodología

Para el desarrollo de este proyecto se empleó la metodología ágil llamada Prototipo, ya que es definida como un método de desarrollo de sistemas, en donde los prototipos llevan a cabo la construcción, prueba y reconstrucción según sea necesario. La finalidad de esta metodología es poder realizar los cambios pertinentes al sistema en una etapa temprana, evitando así pérdidas de tiempo, dinero y esfuerzo en el desarrollo. El prototipo sufre una iteración de reconstrucción considerable hasta obtener el modelo deseado, para posteriormente llevar a cabo la implementación de un sistema funcional aprobado por el cliente/usuario final.

Dentro de las fases que conforman esta metodología se encuentran: definición de los objetivos del prototipo, validación de los requerimientos a través de la funcionalidad del producto, diseñar y desarrollar el prototipo, así como evaluar su funcionamiento.

Durante la fase de diseño y desarrollo se integra la metodología Knowledge Discovery in Databases (KDD), que consiste en el proceso de buscar patrones a través de los datos y poder conocer la situación actual o futura de alguna dependencia a través de modelos matemáticos, dando paso al análisis por parte del usuario. Por lo regular se pre procesan los datos, luego se lleva a cabo minería de datos para poder mostrar resultados.

Las fases que conforman la metodología KDD son: selección, preprocesamiento/limpieza, transformación/reducción, minería de datos e interpretación/evaluación.

En la Figura 3 se muestra el flujo de las fases y la unión de las metodologías utilizadas:

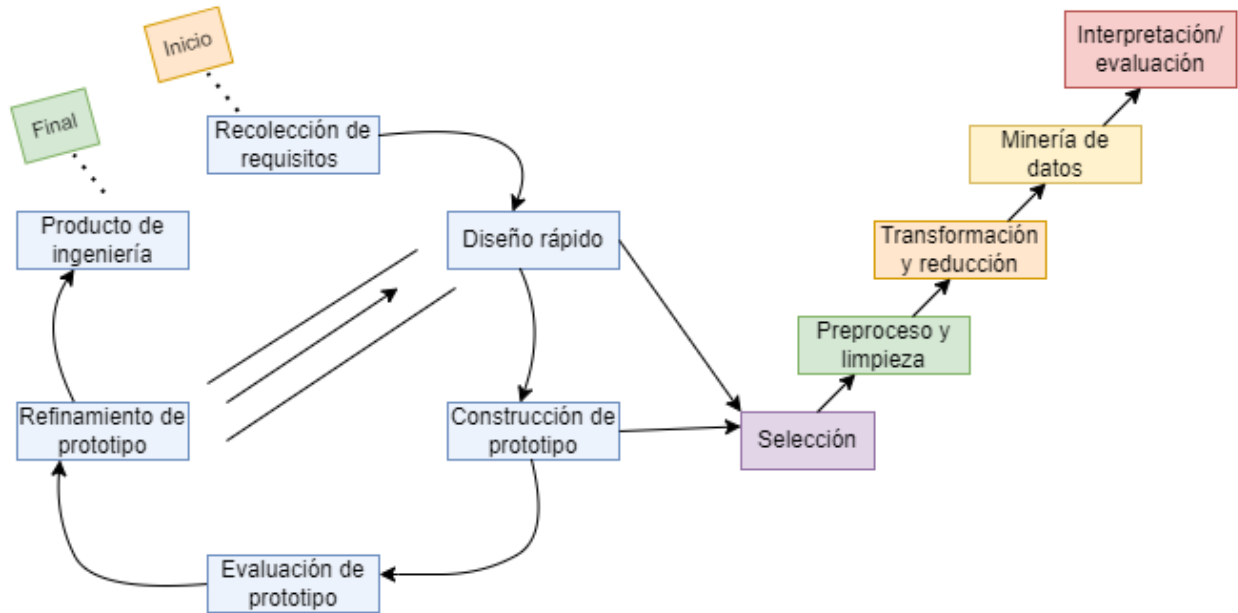


Figura 3. Metodología Prototipo y KDD. Fuente: Autor.

Las actividades realizadas acorde a las fases de las metodologías se describen a continuación.

1. Fase de recolección y refinamiento de requisitos

Para la recolección y refinamiento de requisitos se llevó a cabo la investigación de los requerimientos del proyecto, esto conllevó a la investigación sobre las herramientas tecnológicas candidatas a la incorporación en el desarrollo del proyecto.

Debido a que existen diferentes tecnologías que ayudan a la creación de redes neuronales, se investigó sobre las que tienen mayor relación con el proyecto, las cuales se explican en el marco teórico, pudiendo así, hacer una selección entre las mejores encontradas para posteriormente poder utilizarlas en el desarrollo.

2. Fase de diseño

Se diseñó la estructura de la red neuronal y del modelo donde se alojan los datos obtenidos del monitoreo de la red de Internet, para que con ello la red neuronal sea entrenada y aprenda a reconocer patrones y anomalías que presentan dichos datos.

Dentro de esta fase se integra la metodología KDD. Para el procesamiento de los datos que fueron utilizados para el entrenamiento de la red neuronal fue a través de la

metodología KDD, donde se seleccionan los datos a procesar, posteriormente se procesa y limpian, luego se transforman para que se pueda llevar a cabo la minería de los datos y así poder interpretarlos y hacer su respectiva evaluación obteniendo resultados.

3. Fase de construcción del prototipo

Se construyó el modelo y la estructura de la red neuronal, de acuerdo con las características y los datos que son utilizados para el entrenamiento, se utilizaron las herramientas tecnológicas que más se alineaban a los requerimientos inicial.

Se consideró inicialmente la cantidad y tipos de datos a analizar, en este caso datos obtenidos del previo monitoreo de la red de internet.

4. Fase de evaluación del prototipo por el cliente

Se realizaron diferentes entrenamientos con datos en donde se pueda detectar la anomalía de la red de internet, para poder corroborar el correcto funcionamiento de la red neuronal entrenada. Tales datos muestran una cantidad diferente de paquetes a analizar, así como el tamaño de consumo de banda ancha por paquete.

5. Fase de refinamiento del prototipo

Se realizaron ajustes adecuados de acuerdo a las épocas de entrenamiento, el margen de error y validación, cantidad de capas y neuronas, así como tipo de conexión, con la finalidad de tener como resultado una red neuronal precisa, con un margen de error casi nulo. Dando paso a un análisis más completo y favorable para el sistema.

6. Fase de producto de ingeniería

Para esta última fase, se llevó a cabo la construcción y entrenamiento de la red neuronal completa y con los ajustes correspondientes para poder así implementarlo a la parte del sistema, siendo esta parte el backend. En esta fase se obtienen resultados a través de la red neuronal, es decir, la red logra identificar anomalías típicas y atípicas de acuerdo a los datos con los que se entrenó, así mismo queda lista para su implementación.

La estructura de la red neuronal y el modelo se diseñaron de acuerdo a las necesidades del sistema, en este caso primeramente se cargaron los datos que fueron utilizados para

el entrenamiento del modelo, dichos datos se cargaron tal como se muestra en la Figura 4.

```
#lectura de los datos de los archivos en formatos csv
df_train = pd.read_csv('mayorTrain.csv', parse_dates=True, index_col="Time")
df_test = pd.read_csv('menort_testt.csv', parse_dates=True, index_col="Time")
print(df_train.shape)
print(df_test.shape)

print(df_train.head())
print(df_test.head())
```

Figura 4. Datos cargados para entrenamiento de modelo. Fuente: Autor.

Para el train se utilizó un archivo en formato csv que contiene un total de 71,195 datos sobre el tiempo y ancho de banda consumido por paquete de red. En el test se utilizó el archivo en el mismo formato que train con la diferencia que test contiene un total de 18,519 datos.

Estos archivos contienen paquetes con anomalías los cuales ayudaron a que la red neuronal y su modelo se resultara con mayor precisión de detección.

Una vez cargados los datos se procedió a realizar la selección de los datos para el entrenamiento, para este paso se utilizó el archivo de train, ya que contiene una mayor cantidad de datos, lo cual ayuda a tener un mejor resultado. En la Figura 5 se muestra la selección de los datos.

```
#selección de datos a entrenar
training_mean = df_train.mean()
training_std = df_train.std()
df_training_value = (df_train - training_mean) / training_std
print("Número de muestras de entrenamiento:", len(df_training_value))
```

Figura 5. Selección de los datos. Fuente: Autor.

Fue necesario crear secuencias para el entrenamiento del modelo con instantes de tiempo para que las neuronas recibieran la entrada de la capa anterior. Así como su salida del instante de tiempo anterior para la generación de la salida. En la Figura 6 se muestra la estructura del código para realizar lo descrito anteriormente.

```

TIME_STEPS = 200
# Generar secuencias de entrenamiento para usar en el modelo
def create_sequences(values, time_steps=TIME_STEPS):
    output = []
    for i in range(len(values) - time_steps + 1):
        output.append(values[i : (i + time_steps)])
    return np.stack(output)

x_train = create_sequences(df_training_value.values.astype(np.float32))
print("Entrada de entrenamiento: ", x_train.shape)

```

Figura 6. Secuencias de entrenamiento. Fuente: Autor.

Posteriormente, generadas las secuencias se continuó con el diseño de la estructura del modelo secuencial, que, en conjunto con lo descrito anteriormente, quedó listo para su entrenamiento. El modelo contiene 5 capas de las cuales 1 es de entrada, una de dropout para la regularización del modelo, una de conv1Transpose para la salida de alguna convolución a alguna que tiene forma de su entrada, pero manteniendo un patrón de conectividad con la original, de igual manera cuenta con una capa de salida para resultados. A continuación, se muestra la estructura del modelo secuencial (ver Figura 7).

```

#creación del modelo secuencial
model = keras.Sequential(
    [
        layers.Input(shape=(x_train.shape[1], x_train.shape[2])),
        layers.Conv1D(
            filters=32, kernel_size=7, padding="same", strides=2, activation="relu"
        ),
        layers.Dropout(rate=0.1),
        layers.Conv1D(
            filters=16, kernel_size=7, padding="same", strides=2, activation="relu"
        ),
        layers.Conv1DTranspose(
            filters=16, kernel_size=7, padding="same", strides=2, activation="relu"
        ),
        layers.Dropout(rate=0.1),
        layers.Conv1DTranspose(
            filters=32, kernel_size=7, padding="same", strides=2, activation="sigmoid"
        ),
        layers.Conv1DTranspose(filters=1, kernel_size=7, padding="same"),
    ]
)
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.000001), loss="binary_crossentropy")
model.summary()

```

Figura 7. Modelo secuencias. Fuente: Autor.

Por último, se determina el código para el entrenamiento del modelo con un total de 50 épocas, un `batch_size` de 128 y una `validation_split` de 0.333 tal como se muestra en la Figura 8.

```
#entrenamiento del modelo
history = model.fit(
    x_train,
    x_train,
    epochs=50,
    batch_size=128,
    validation_split=0.333,
    callbacks=[
        keras.callbacks.EarlyStopping(monitor="val_loss", patience=5, mode="min")
    ],
)
```

Figura 8. Código para el entrenamiento del modelo. Fuente: Autor.

Una vez creada la estructura y código del modelo se continua con la determinación de detección de las anomalías en los datos, para ello es primordial obtener el error absoluto medio (MAE), así como la pérdida de reconstrucción. El código se muestra en la Figura 9.

```

# Obtener MAE de entrenamiento
x_train_pred = model.predict(x_train)
train_mae_loss = np.mean(np.abs(x_train_pred - x_train), axis=1)
plt.hist(train_mae_loss, bins=50)
plt.xlabel("Train MAE loss")
plt.ylabel("No of samples")
plt.show()

# Obtener umbral de pérdida de reconstrucción
threshold = np.max(train_mae_loss)
print("Reconstruction error threshold: ", threshold)
plt.plot(x_train[0])
plt.plot(x_train_pred[0])
plt.show()
df_test_value = (df_test - training_mean) / training_std
fig, ax = plt.subplots()
df_test_value.plot(legend=False, ax=ax)
plt.show()

# Crear secuencias a partir de valores de prueba
x_test = create_sequences(df_test_value.values)
print("Test input shape: ", x_test.shape)

# Obtener prueba de pérdida MAE
x_test_pred = model.predict(x_test)
test_mae_loss = np.mean(np.abs(x_test_pred - x_test), axis=1)
test_mae_loss = test_mae_loss.reshape((-1))
plt.hist(test_mae_loss, bins=40)
plt.xlabel("test MAE loss")
plt.ylabel("No of samples")
plt.show()

```

Figura 9. Obtención de MAE. Fuente: Autor.

Así mismo se generó el código que ayudó a la detección de las anomalías presentadas en los datos. Dicho código se presenta en la Figura 10.

```

# Detectar todas las muestras que son anomalías
anomalies = test_mae_loss > threshold
print("Number of anomaly samples: ", np.sum(anomalies))
print("Indices of anomaly samples: ", np.where(anomalies))

anomalous_data_indices = []
for data_idx in range(TIME_STEPS - 1, len(df_test_value) - TIME_STEPS + 1):
    if np.all(anomalies[data_idx - TIME_STEPS + 1 : data_idx]):
        anomalous_data_indices.append(data_idx)

df_subset = df_test.iloc[anomalous_data_indices]
fig, ax = plt.subplots()
df_test.plot(legend=False, ax=ax)
df_subset.plot(legend=False, ax=ax, color="r")
plt.show()

```

Figura 10. Detección de anomalías. Fuente: Autor.

Finalmente, creadas y aprobadas cada una de las partes del modelo se pudo continuar con el entrenamiento completo del modelo y su realización de pruebas.

CAPÍTULO 4: RESULTADOS

10. Resultados obtenidos

A continuación, se presentan los resultados obtenidos. Iniciando con la descripción de las herramientas seleccionadas acorde a la investigación realizada. Seguidamente, se muestran los resultados de la estructura del modelo diseñado, así como su posterior entrenamiento. De igual manera se muestran los ajustes realizados a la red para obtener una mejor precisión de la detección de las anomalías.

10.1 Selección de herramientas

De acuerdo con la selección de herramientas tecnológicas se optó por las librerías Tensorflow y Keras para el procesamiento de los datos, ya que dichas librerías trabajan con cantidades grandes de estos. También se utilizó la librería pandas, que su función es ayudar a la lectura de los datos mostrados en formato csv. Así como también la librería numpy y matplotlib para datos matriciales y creación de gráficas.

Estas herramientas en conjunto ayudan a que el desarrollo de la red neuronal sea más sencillo y rápido a diferencia del entrenamiento manualmente. En la Figura 11 se muestran las líneas de código donde se importan y utilizan las librerías mencionadas anteriormente.

```
import numpy as np #libreria para datos matriciales
import pandas as pd #libreria para leer datos en formato csv
from tensorflow import keras
from keras import layers #libreria para aprendizaje profundo en redes neuronales
from matplotlib import pyplot as plt #libreria para crear gráficas
```

Figura 11. Importación de librerías en código.

10.2 Entrenamiento del modelo

Una vez cargados los datos y seleccionados, así como la construcción del modelo, se procedió a realizar el entrenamiento de dicho modelo para comprobar su funcionamiento y poder continuar con la implementación para la detección de las anomalías de la red.

Primeramente, se muestra la gráfica de los datos del archivo de test como se puede apreciar en la Figura 12, se expone el tiempo y consumo de ancho de banda en los paquetes de tal archivo.

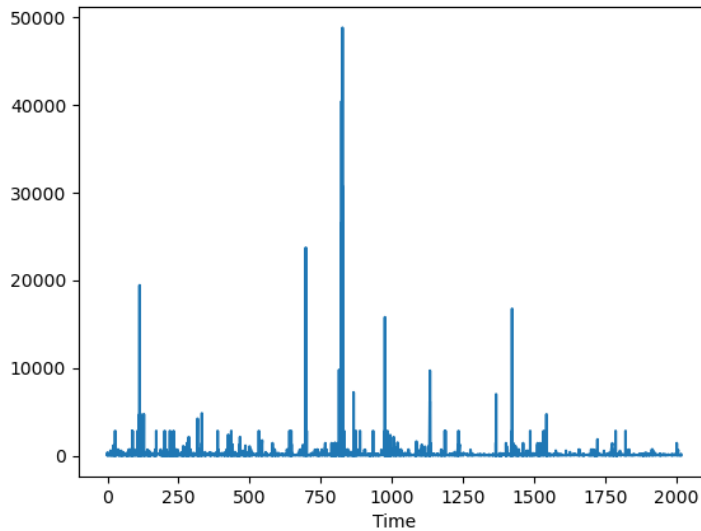


Figura 12. Datos del archivo de test. Fuente: Autor.

De igual manera se muestra la gráfica de los datos del archivo de train en la Figura 13, donde también se expone el tiempo en segundos y consumo de ancho de banda en los paquetes del archivo train.

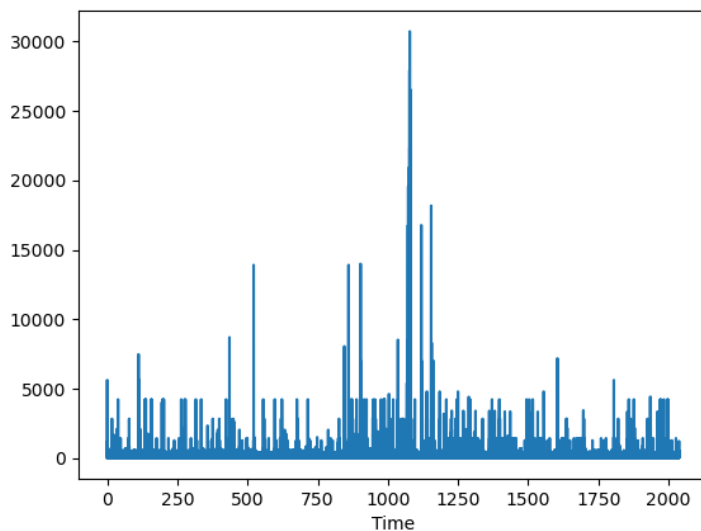


Figura 13. Datos del archivo train. Fuente: Autor.

Una vez mostradas las gráficas de los datos, se continua con el proceso de entrenamiento en la Figura 14 se muestra el resumen del modelo secuencial creado.

```
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv1d (Conv1D)              (None, 100, 32)            256
dropout (Dropout)           (None, 100, 32)            0
conv1d_1 (Conv1D)           (None, 50, 16)             3600
conv1d_transpose (Conv1DTran (None, 100, 16)            1808
nspose)
dropout_1 (Dropout)         (None, 100, 16)            0
conv1d_transpose_1 (Conv1DT (None, 200, 32)            3616
ranspose)
conv1d_transpose_2 (Conv1DT (None, 200, 1)            225
ranspose)
-----
Total params: 9,505
Trainable params: 9,505
Non-trainable params: 0
```

Figura 14. Resumen de modelo creado. Fuente: Autor.

Se comienza a entrenar el modelo, procesando cada una de las épocas establecidas en el código de entrenamiento, el resultado de esto se muestra en la Figura 15.

```

Epoch 17/50
370/370 [=====] - 7s 20ms/step - loss: 0.4391 - val_loss: -0.9507
Epoch 18/50
370/370 [=====] - 11s 29ms/step - loss: 0.4375 - val_loss: -0.9507
Epoch 19/50
370/370 [=====] - 15s 41ms/step - loss: 0.4360 - val_loss: -0.9506
Epoch 20/50
370/370 [=====] - 7s 20ms/step - loss: 0.4342 - val_loss: -0.9509
Epoch 21/50
370/370 [=====] - 9s 25ms/step - loss: 0.4318 - val_loss: -0.9510
Epoch 22/50
370/370 [=====] - 7s 18ms/step - loss: 0.4305 - val_loss: -0.9510
Epoch 23/50
370/370 [=====] - 9s 25ms/step - loss: 0.4290 - val_loss: -0.9511
Epoch 24/50
370/370 [=====] - 9s 25ms/step - loss: 0.4256 - val_loss: -0.9485
Epoch 25/50
370/370 [=====] - 10s 27ms/step - loss: 0.4233 - val_loss: -0.9515
Epoch 26/50
370/370 [=====] - 13s 34ms/step - loss: 0.4213 - val_loss: -0.9515
Epoch 27/50
370/370 [=====] - 8s 21ms/step - loss: 0.4182 - val_loss: -0.9513
Epoch 28/50
370/370 [=====] - 10s 26ms/step - loss: 0.4143 - val_loss: -0.9517
Epoch 29/50
370/370 [=====] - 7s 19ms/step - loss: 0.4114 - val_loss: -0.9519
Epoch 30/50
370/370 [=====] - 9s 25ms/step - loss: 0.4057 - val_loss: -0.9519
Epoch 31/50
370/370 [=====] - 8s 23ms/step - loss: 0.4006 - val_loss: -0.9520
Epoch 32/50
370/370 [=====] - 9s 23ms/step - loss: 0.3952 - val_loss: -0.9522
Epoch 33/50
370/370 [=====] - 13s 34ms/step - loss: 0.3846 - val_loss: -0.9524
Epoch 34/50
370/370 [=====] - 9s 25ms/step - loss: 0.3735 - val_loss: -0.9525
Epoch 35/50
370/370 [=====] - 10s 26ms/step - loss: 0.3577 - val_loss: -0.9517
Epoch 36/50
55/370 [==>.....] - ETA: 7s - loss: 0.3428

```

Figura 15. Entrenamiento del modelo. Fuente: Autor.

En la Figura 16, se muestra el resultado de la validación y entrenamiento, es decir, muestra cómo es que fue disminuyendo el margen de error durante el entrenamiento. Se puede observar que aún está muy lejos de poder predecir con exactitud las anomalías, es por ello que se requirieron ajustes en el modelo.

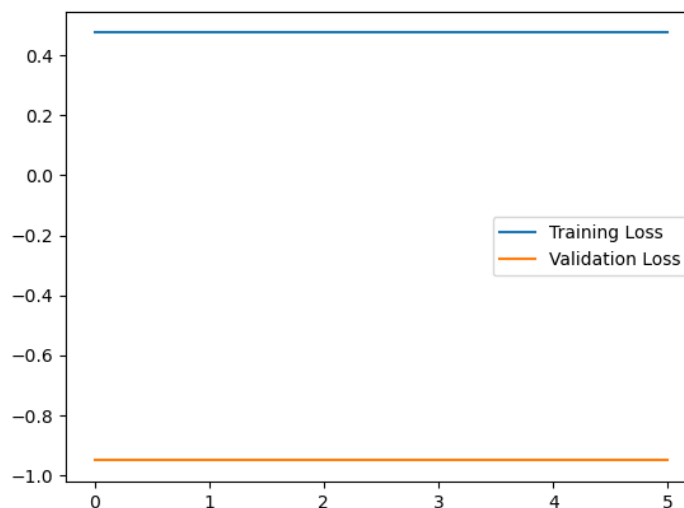


Figura 16. Gráfica de muestra de margen de error. Fuente: Autor.

En las siguientes graficas se muestra resultado del MAE del train y el umbral de pérdida de reconstrucción (ver Figura 17).

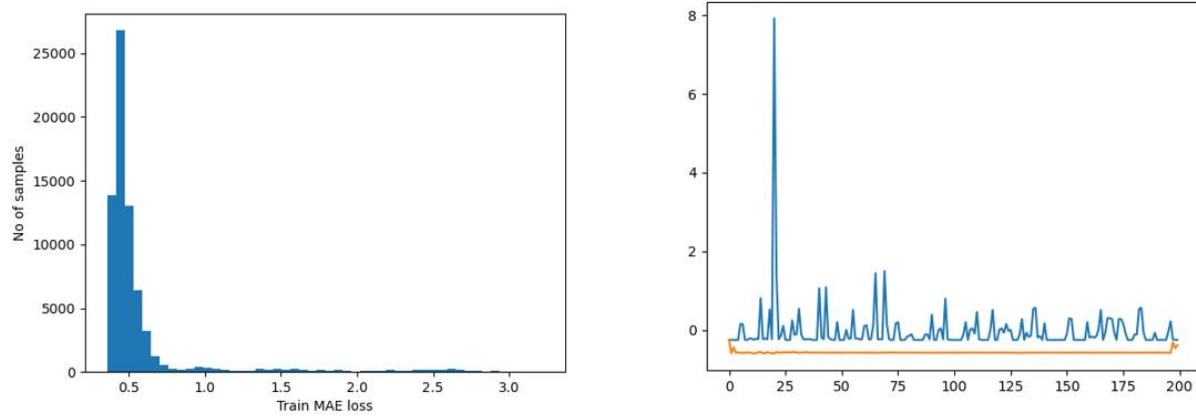


Figura 17. Resultado MAE y reconstrucción train. Fuente: Autor.

En estas graficas se muestra el resultado MAE y reconstrucción, pero sobre el archivo test (ver Figura 18).

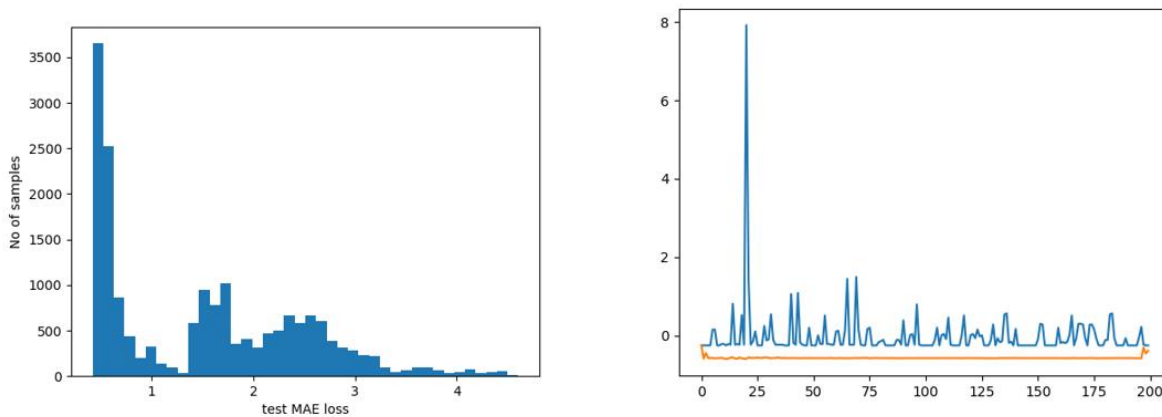


Figura 18. Resultado MAE y reconstrucción test. Fuente: Autor.

Por último, se obtiene la Figura 19, donde se muestra marcada con color rojo la detección de la anomalía, es decir señala el punto con mayor consumo de ancha banda del total de los paquetes que fueron utilizados para el entrenamiento. Se puede apreciar que no se obtuvo un resultado preciso, ya que solo marca una parte de la anomalía presentada,

debido a ello se recurrió al refinamiento y ajuste de los datos, dando paso a la siguiente fase, refinamiento del modelo.

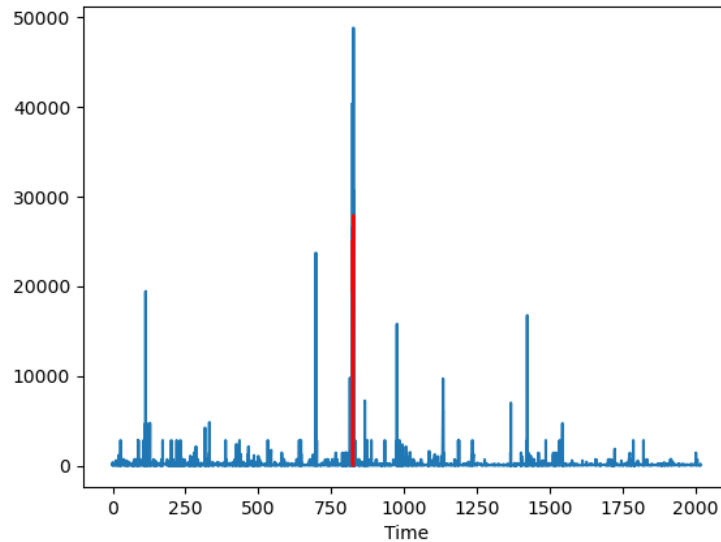


Figura 19. Resultado de detección de anomalía. Fuente: Autor.

10.3 Refinamiento del modelo y producto de ingeniería

En esta fase se modificaron valores del modelo creado, debido a que en el primer prototipo no se obtuvieron los resultados deseados.

Se modificó la parte del código del entrenamiento, inicialmente se tenía un `batch_size` de 0.333 el cual fue cambiado a 0.33, ya que se refiere a que la red tiene en memoria poca cantidad de datos, dichos cambios se muestran en la Figura 20.

Estos cambios dieron como resultado las siguientes gráficas que se muestran a continuación (ver Figura 21 y Figura 22).

Para finalizar éste fue el único ajuste realizado a la red y gracia a ello se obtuvieron resultados con mayor exactitud.

```
#entrenamiento del modelo
history = model.fit(
    x_train,
    x_train,
    epochs=50,
    batch_size=128,
    validation_split=0.33,
    callbacks=[
        keras.callbacks.EarlyStopping(monitor="val_loss", patience=5, mode="min")
    ],
)
```

Figura 20. Cambios al modelo. Fuente: Autor.

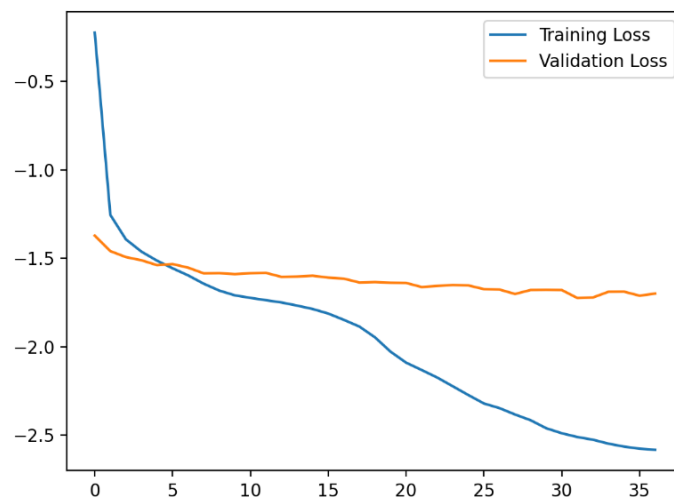


Figura 21. Gráfica de muestra de margen de error con cambios. Fuente: Autor.

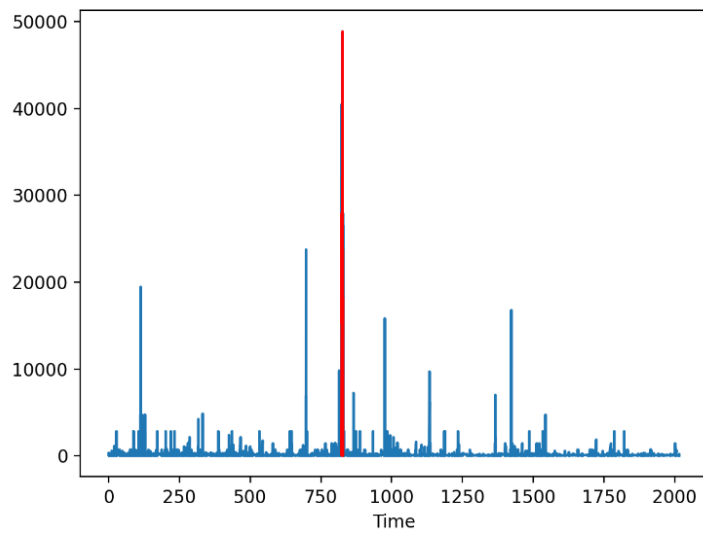


Figura 22. Resultado de detección de anomalía mejorada. Fuente: Autor.

El modelo y red neuronal quedaron listos para su posterior implementación a futuro, ya como trabajo a futuro se llevará acabo el desarrollo de una página web donde se pueda mostrar en tiempo real el tráfico y anomalías que suceden en la red de internet de LIA.

CAPÍTULO 5: CONCLUSIONES

11. Conclusiones del Proyecto

Predecir el tráfico de la red a través de redes neuronales resulta más sencillo, ya que la red neuronal se encarga de realizar todo el trabajo una vez entrenada.

Gracias a su entrenamiento y ajuste de los valores, después de realizar varias iteraciones de entrenamiento en el modelo, el margen de error tiende a ser cero, por lo que se obtienen buenos resultados.

Las redes neuronales son de gran utilidad, ya que con ellas se facilita la adaptación a diferentes entornos, por ejemplo, tráfico en general y redes de comunicación. Sin embargo, es necesario tener las herramientas tecnológicas con alta capacidad de cómputo para que el resultado se obtenga en un menor tiempo y con mayor exactitud.

Con respecto al análisis de tráfico en espacios físicos, se concluye que es importante estar monitoreando la red de cualquier área en específico, debido a los problemas que pueden presentarse en el envío y recepción de paquetes, ya que gracias a la detección de dichos problemas se puede efectuar una pronta solución.

La concentración de la información obtenida del monitoreo de red para el entrenamiento del modelo, resulta mucho más sencillo y practico guardarla en formatos csv, ya que de esta manera se pueden cargar los datos de una manera más práctica.

Respecto al análisis de la información para la detección de patrones y sus consecuencias, se concluye que mientras la cantidad de datos a analizar sea mayor, se obtiene una detección de anomalías con mayor exactitud.

De acuerdo con la creación de la página web para la muestra del análisis, se deja como trabajo a futuro ya que es importante tenerla bien estructurada por lo que se requiere más tiempo para lograrlo.

CAPÍTULO 6: COMPETENCIAS DESARROLLADAS

12. Competencias desarrolladas y/o aplicadas.

1. Adquirí habilidades en investigación y selección de herramientas tecnológicas adecuadas al desarrollo del proyecto.
2. Adquirí conocimiento en la recopilación de requerimientos que demandaba el proyecto.
3. Apliqué conocimientos en minería de datos obtenidos a través de la investigación.
4. Utilicé herramientas tecnológicas como librerías, para un mejor manejo de grandes cantidades de datos.
5. Aprendí a seleccionar el tipo de red neuronal adecuado para la detección de anomalías en la red de internet.
6. Desarrollé redes neuronales y modelos entrenados con base en los conocimientos obtenidos.
7. Aprendí a analizar graficas de datos basadas en la detección de anomalías.
8. Apliqué técnicas de manejo de datasets a través de minería de datos.
9. Diseñé e implementé posibles prototipos de redes neuronales para la detección de anomalías.
10. Desarrollé técnicas de aprendizaje de nuevos temas para su posterior aplicación en distintas áreas relacionadas.

CAPÍTULO 7: FUENTES DE INFORMACIÓN

13. Fuentes de información

Andres, G., Herrera, R., Armando, D., & Ramirez, G. (n.d.). *DISEÑO DE UNA RED LAN PARA DAVID RESTREPO ARQUITECTOS UNIVERSIDAD COOPERATIVA DE COLOMBIA FACULTAD DE INGENIERIA INGENIERIA ELECTRONICA SECCIONAL BOGOTA DC NOVIEMBRE DE 2019.*

aws. (2022a). *¿Qué es Python? | Guía de Python para principiantes de la nube | AWS.*
<https://aws.amazon.com/es/what-is/python/>

aws. (2022b). *¿Qué es una red neuronal? Guía de IA y ML - AWS.*
<https://aws.amazon.com/es/what-is/neural-network/>

Cuellar Duque Yeferson, V. S. F. (2021). *DISEÑO E INSTALACIÓN (SIMULADA) DE UNA RED WLAN QUE PERMITA LA CONEXIÓN Y AUDITORIA CON EL SERVIDOR SAMMU DE LAS MAQUINAS INDUSTRIALES DE CONFECCIÓN EN LA ESCUELA GENERAL SANTANDER DE BOGOTÁ [UNIVERSIDAD COOPERATIVA DE COLOMBIA FACULTAD DE INGENIERÍA BOGOTÁ].*
http://74.208.53.179/bitstream/20.500.12494/33785/11/2021_DISEÑO_RED_WLAN.pdf

De Jaén, U., José, A. :, Parrilla Martínez, A., & Cruz, C. M. (2020). *Trabajo Fin de Máster.*

De Luz Sergio. (2022). *Qué es el protocolo SMTP de correo saliente y cómo configurarlo.* <https://www.redeszone.net/tutoriales/internet/que-es-protocolo-smtp-email-configuracion/>

Estrada Adrián. (n.d.). *PROTOCOLOS TCP/IP DE INTERNET INTERNET: DEL USO MILITAR AL USO CIVIL.* Retrieved September 22, 2022, from <http://www.revista.unam.mx/vol.5/num8/art51/art51.htm>

Guzmán Y Valle, E., Máter, A., Magisterio, D., Facultad, N., Tecnología, D. E., Luis, S., & Marx, C. (2018). *UNIVERSIDAD NACIONAL DE EDUCACIÓN Protocolo de comunicación TCP/IP y ethernet Examen de Suficiencia Profesional Res. N° 0493-2018-D-FATEC Presentada por.*

IONOS. (2022a). *¿Qué es Keras? Introducción a la biblioteca de redes neuronales - IONOS.* <https://www.ionos.mx/digitalguide/online-marketing/marketing-para->

motores-de-busqueda/que-es-keras/

IONOS. (2022b). *HTTPS: el protocolo HTTP cifrado y más seguro* - IONOS.

<https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/que-es-https/>

Larkin, J. (2022). *¿Qué es TensorFlow y para qué sirve?* <https://www.incentro.com/es-ES/blog/que-es-tensorflow>

Lederkremer Miguel. (2019). *Redes Informáticas* (Claudio Peña Millah...).

[https://books.google.es/books?id=7frADwAAQBAJ&pg=PA3&lpg=PA1&ots=H5jHV6p_hv&focus=viewport&dq=tipos+de+redes+informáticas&lr=&hl=es#v=onepage&q=tipos de redes informáticas&f=false](https://books.google.es/books?id=7frADwAAQBAJ&pg=PA3&lpg=PA1&ots=H5jHV6p_hv&focus=viewport&dq=tipos+de+redes+informáticas&lr=&hl=es#v=onepage&q=tipos+de+redes+informáticas&f=false)

Oviedo, B., Suarez Litardo, L., Zhuma Mera, E., Puris, A., & Hernández, R. (2019).

VISUALIZADOR. <https://orcid.org/0000-0002-5366-5917>

Rivas, W., Bertha, A., & Olivo, M. (n.d.). *Redes neuronales artificiales aplicadas al reconocimiento de patrones*. Retrieved November 1, 2022, from www.utmachala.edu.ec

Solarwinds. (2022). *Monitor del tráfico de red: ver, analizar y optimizar el tráfico* |

SolarWinds. <https://www.solarwinds.com/es/network-bandwidth-analyzer-pack/use-cases/network-traffic-monitor>