

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Electricidad y Computación**

Implementación de un algoritmo para la detección temprana de fallas en equipos inalámbricos de una empresa proveedora de servicios de Internet utilizando técnicas de aprendizaje autónomo

### **PROYECTO DE TITULACIÓN**

Previo la obtención del Título de:

### **Magister en Telecomunicaciones**

Presentado por:

Omar Arturo Haro Velasco

GUAYAQUIL - ECUADOR

Año: 2023

## DEDICATORIA

A mis padres Gladys y Samuel; porque son mi mayor motivación, ejemplo de esfuerzo y dedicación para seguir cosechando triunfos y siempre impulsarme a ser mejor.

A mis hermanos Alex y Amanda a mis sobrinos Axel y Nadia a mi cuñad@ Claudio y Erika; por su apoyo moral incondicional y constante durante todo este trabajo.

También quiero dedicar este trabajo a Sandra H. quien me apoyo incondicionalmente y alentó a seguir cuando más quise abandonar.

***Omar Haro***

## **AGRADECIMIENTOS**

Mi más sincero agradecimiento a mi Director de tesis Ronald Criollo, por brindar su apoyo en el desarrollo de este proyecto final de Maestría.

A todos los profesores de la FIEC que brindaron sus conocimientos ayudándome a fortalecer mis conocimientos.

Agradecimiento especial a toda mi familia por su apoyo durante este proceso.

***Omar Haro***

## **DECLARACIÓN EXPRESA**

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Omar Arturo Haro Velasco doy 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”

Omar Arturo  
Haro Velasco

## EVALUADORES

.....  
Ms.C. Ronald Criollo Bonilla

PROFESOR TUTOR

.....  
Ph.D. Maria Antonieta Alvarez

PROFESOR EVALUADOR

## RESUMEN

La empresa proveedora de servicio de Internet, requieren disponer de un sistema de detección temprana de fallas en equipos de inalámbricos utilizando técnicas de aprendizaje autónomo, a fin de disponer de un sistema que pueda disminuir el tiempo de inactividad de los equipos y mejor calidad de servicio.

Para resolver el presente problema se implementó un sistema de adquisición de datos históricos utilizando la herramienta Zabbix, trece equipos inalámbricos y un Punto de Acceso. La información del sistema de adquisición de datos se almacena en una base de datos dicha información se utilizó para el entrenamiento de los diferentes algoritmos de aprendizaje supervisado.

El presente trabajo inició con la descripción del problema y las necesidades de mantenimiento de los equipos inalámbricos; continuando con la investigación sobre conceptos que formaron parte de este proyecto como son: las técnicas de mantenimiento de equipos de telecomunicaciones, técnicas de aprendizaje autónomo, sistemas de adquisición de datos y métricas de evaluación de algoritmos. Se implementó el sistema de adquisición de datos utilizando la herramienta Zabbix. Finalmente se realizó el procesamiento de datos y el entrenamiento de los diferentes algoritmos de aprendizaje autónomo supervisado utilizando los datos obtenidos de los equipos inalámbricos.

Mediante el entrenamiento de los diferentes algoritmos de aprendizaje autónomo supervisado se pudo determinar que los algoritmos K vecinos más cercanos ofrecen una mejor precisión y un menor costo de validación, por lo cual es la mejor opción para la implementación del algoritmo para la detección temprana de fallas en equipos inalámbricos.

**Palabras Clave:** Aprendizaje autónomo, Zabbix, equipos inalámbricos, algoritmo

## **ABSTRACT**

*The Internet service provider company requires an early detection system for failures in wireless equipment using machine learning techniques, in order to have a system that can reduce equipment downtime and improve service quality.*

*To solve this problem, a historical data acquisition system was implemented using the Zabbix tool, thirteen wireless devices and an access point. The information of the data acquisition system is stored in a database, said information was used for the training of the different supervised learning algorithms.*

*The present work begins with the description of the problem and the maintenance needs of the wireless equipment; continuing with the research on concepts that were part of this project such as: maintenance techniques for telecommunications equipment, machine learning techniques, data acquisition systems and algorithm evaluation metrics. The data acquisition system was implemented using the Zabbix tool. Finally, the data processing and the training of the different supervised machine learning algorithms were carried out using the data obtained from the wireless equipment.*

*Through the training of the different supervised machine learning algorithms, it was possible to determine that the K nearest neighbor algorithms offer better precision and a lower validation cost, which is why it is the best option for the implementation of the algorithm for the early detection of failures. on wireless equipment.*

*Keywords: Machine Learning, Zabbix, wireless equipment, algorithm*

# ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	IX
CAPÍTULO 1.....	1
1.    Introducción.....	1
1.1    Descripción del problema.....	1
1.2    Justificación del problema.....	2
1.3    Objetivos.....	3
1.3.1    Objetivo General.....	3
1.3.2    Objetivos Específicos.....	3
1.4    Propuesta de la solución.....	3
1.5    Alcance del trabajo.....	6
CAPÍTULO 2.....	7
2.    Marco teórico.....	7
2.1    Técnicas de mantenimiento de equipos de telecomunicaciones.....	7
2.1.1    Mantenimiento correctivo.....	7
2.1.2    Mantenimiento preventivo.....	9



2.1.3	Mantenimiento predictivo .....	10
2.2	Técnicas de aprendizaje autónomo .....	12
2.2.1	Aprendizaje supervisado .....	13
2.2.2	Aprendizaje no supervisado .....	20
2.3	Métricas de evaluación de los algoritmos de clasificación .....	22
2.3.1	Matriz de confusión .....	23
2.3.2	Curva ROC.....	24
2.4	Sistemas de adquisición de datos .....	25
2.4.1	Sistemas de monitoreo de red .....	25
2.4.2	Protocolo Simple de Administración de Red .....	26
2.4.3	Herramientas para monitoreo de redes de telecomunicaciones .....	30
2.4.4	Base de datos relacional .....	32
CAPÍTULO 3.....		34
3.	Implementación del algoritmo para la detección de fallas.....	34
3.1	Adquisición de datos.....	34
3.1.1	Identificación del escenario de estudio.....	34
3.1.2	Implementación de sistema de adquisición de datos .....	37
3.1.3	Recreación de los distintos escenarios de funcionamiento del sistema .	40
3.2	Procesamiento de datos .....	50
3.2.1	Exploración de datos.....	51

3.2.2	Preprocesar datos .....	52
3.2.3	Identificación de indicadores .....	54
3.2.4	Análisis de componentes principales .....	59
CAPÍTULO 4.....		63
4.	Análisis de resultados .....	63
4.1	Evaluación y entrenamiento de algoritmos de aprendizaje autónomo.....	63
4.1.1	Entrenamiento con los algoritmos de árbol de decisión .....	63
4.1.2	Entrenamiento con los algoritmos de análisis discriminante .....	65
4.1.3	Entrenamiento con los algoritmos Naive Bayes .....	66
4.1.4	Entrenamiento con los algoritmos SVM .....	67
4.1.5	Entrenamiento con los algoritmos KNN.....	69
4.2	Comparativo de eficiencia por algoritmo de predicción .....	71
4.3	Comparativo de eficiencia por estado de funcionamiento.....	72
CAPITULO 5.....		78
5.	Conclusiones y Recomendaciones .....	78
5.1	Conclusiones .....	78
5.2	Recomendaciones .....	78
BIBLIOGRAFÍA.....		80
Anexo 1.....		87

## ÍNDICE DE FIGURAS

Figura 1.1 Esquema de la solución .....	4
Figura 2.1 Mantenimiento correctivo .....	8
Figura 2.2 Mantenimiento preventivo .....	9
Figura 2.3 Mantenimiento predictivo .....	11
Figura 2.4 Técnicas de aprendizaje autónomo .....	13
Figura 2.5 Estructura del algoritmo Support Vector Machines .....	15
Figura 2.6 Estructura del algoritmo de árbol de decisión .....	16
Figura 2.7 Estructura de una red neuronal .....	19
Figura 2.8 Agrupación por clusters .....	20
Figura 2.9 Matriz de confusión .....	23
Figura 2.10 Arquitectura de SNMP .....	27
Figura 2.11 Comunicación SNMP .....	28
Figura 2.12 Intercambio de mensajes SNMP .....	30
Figura 3.1 Escenario de estudio .....	34
Figura 3.2 Distribución de CPE .....	37
Figura 3.3 Sistema de adquisición de datos .....	38
Figura 3.4 Parámetro de enlace inalámbrico .....	39
Figura 3.5 Alineación correcta .....	40
Figura 3.6 Error de alineación horizontal .....	42

Figura 3.7 Error de alineación vertical .....	43
Figura 3.9 Potencia de Ruido en funcionamiento Correcto.....	47
Figura 3.10 SNR en funcionamiento Correcto .....	47
Figura 3.13 SNR en funcionamiento Advertencia .....	49
Figura 3.16 SNR en funcionamiento Fallo .....	50
Figura 3.17 Comportamiento de las variables .....	53
Figura 3.18 Histograma de la variable PRxStrength .....	54
Figura 3.19 Histograma de la variable PRxStrengthCh0 .....	55
Figura 3.20 Histograma de la variable PRxStrengthCh1 .....	55
Figura 3.21 Histograma de la variable PTxStrength .....	56
Figura 3.22 Histograma de la variable PTxStrengthCh0.....	57
Figura 3.23 Histograma de la variable PTxStrengthCh1 .....	57
Figura 3.24 Histograma de la variable SNR.....	58
Figura 3.25 Varianza individual y acumulada explicada por PCA .....	60
Figura 3.26 Datos de PCA en escenario Correcto, Advertencia y Fallo.....	61
Figura 4.1 Matriz de confusión Fine Tree .....	64
Figura 4.2 Matriz de confusión Medium Tree.....	65
Figura 4.3 Matriz de confusión Coarse Tree .....	65
Figura 4.4 Matriz de confusión de los algoritmos de análisis discriminante .....	66
Figura 4.5 Matriz de confusión de los algoritmos Naive Bayes.....	67

Figura 4.6 Matriz de confusión SVM Lineal y Cuadrático .....	68
Figura 4.7 Matriz de confusión SVM-Cubico.....	69
Figura 4.8 Matriz de confusión Fine-Medium-Coarse KNN.....	70
Figura 4.9 Matriz de confusión Cubic-Weighted KNN.....	71
Figura 4.10 Eficiencia de algoritmos por estado Correcto .....	73
Figura 4.11 Eficiencia de algoritmos por estado AHD.....	73
Figura 4.12 Eficiencia de algoritmos por estado AHI .....	74
Figura 4.13 Eficiencia de algoritmos por estado AUP .....	74
Figura 4.14 Eficiencia de algoritmos por estado ADW .....	75
Figura 4.15 Eficiencia de algoritmos por estado FHD.....	75
Figura 4.16 Eficiencia de algoritmos por estado FHI .....	76
Figura 4.17 Eficiencia de algoritmos por estado FUP .....	76
Figura 4.18 Eficiencia de algoritmos por estado FDW .....	77

## ÍNDICE DE TABLAS

Tabla 2.1 Ventajas y desventajas del mantenimiento correctivo.....	9
Tabla 2.2 Ventajas y desventajas del mantenimiento preventivo.....	10
Tabla 2.3 Ventajas y desventajas del mantenimiento predictivo.....	12
Tabla 2.4 Versiones de SNMP.....	28
Tabla 3.1 Características de los CPE y Punto de Acceso .....	35
Tabla 3.2 Ubicación de los CPE y el Punto de Acceso.....	36
Tabla 3.3 Variable monitoreadas .....	40
Tabla 3.4 Planes de Internet.....	43
Tabla 3.5 Esquema de Codificación y Modulación .....	44
Tabla 3.6 Rangos de potencia de escenarios de funcionamiento.....	45
Tabla 3.7 Escenarios de funcionamiento .....	45
Figura 3.8 Niveles de Potencia en funcionamiento Correcto .....	46
Figura 3.11 Niveles de Potencia en funcionamiento Advertencia .....	48
Figura 3.12 Potencia de Ruido en funcionamiento Correcto.....	48
Figura 3.14 Niveles de Potencia en funcionamiento Fallo .....	49
Figura 3.15 Potencia de Ruido en funcionamiento Fallo.....	50
Tabla 3.8 Descripción del dataset.....	51
Tabla 3.9 Dataset de cada CPE.....	51
Tabla 3.10 Distribución de frecuencias de todo el dataset.....	52

Tabla 3.11 Descripción de las variables .....	59
Tabla 3.12 Matriz de correlación.....	62
Tabla 4.1 Precisión de los algoritmos de árbol de decisión .....	64
Tabla 4.2 Parámetros de entrenamiento de KNN .....	69
Tabla 4.3 Métricas de los algoritmos entrenados.....	72

# CAPÍTULO 1

## 1. INTRODUCCIÓN

### 1.1 Descripción del problema

Las redes inalámbricas se convirtieron en una herramienta indispensable para brindar servicio de acceso a Internet de alta calidad y disponibilidad a hogares y empresas (Baghaei & Hunt, 2004).

Los equipos dentro de una red inalámbrica son numerosos, más aún, en un Proveedor de Servicios de Internet (ISP, por sus siglas en inglés), en este tipo de empresas se puede encontrar con cientos de equipos conectados a la red que necesitan ser monitoreados de forma continua, para poder detectar de forma rápida posibles fallas o degradaciones en su funcionamiento y recibir mantenimiento adecuado para evitar quejas de clientes.

En un ISP se realiza diferentes tipos de mantenimientos a los equipos inalámbricos, principalmente a las antenas ubicadas en los domicilios de los clientes también denominadas Equipos Locales del Cliente (CPE, por sus siglas en inglés). El no contar con sistema que permita determinar cuando el funcionamiento de un equipo se degrada implica gastos y uso de recursos por parte de la empresa. Esto debido que para solventar las fallas de los equipos se tiene que trasladar al técnico hacia el domicilio del cliente para dar un mantenimiento adecuado.

La falta de un sistema que permita predecir el momento en que se debe realizar el mantenimiento de los equipos inalámbricos dentro la red de un ISP provoca quejas continuas por parte de los clientes debido a fallas inesperadas que ocurren en los enlaces inalámbricos tanto punto a punto como punto multi punto, causando cortes o lentitud en el servicio de Internet.

Al no contar con un registro digital de fallas y parámetros de funcionamiento de los puntos de acceso y de los CPE, se causa daños en los equipos por estar operando



en condiciones anormales durante un periodo prolongado y sin recibir mantenimiento predictivo adecuado, lo que genera pérdidas económicas a la empresa pues dichos equipos deben ser reemplazados.

Las continuas fallas en de los equipos inalámbricos por falta de un sistema de detección temprana de fallas provocan que los clientes dejen de utilizar el servicio de acceso a Internet ofrecido por la empresa, dando una percepción de baja calidad del servicio prestado.

## **1.2 Justificación del problema**

Los proveedores de Internet frente a la continua exigencia de mayor ancho de banda, debido al incremento de tráfico de Internet están obligados a buscar nuevas tecnologías y aplicaciones que permitan a los equipos operar en condiciones óptimas y con un adecuado mantenimiento, de esta manera brindar un servicio de calidad y disponibilidad a los clientes.

En un estudio realizado en los Estados Unidos de América, por la Sociedad Internacional de Automatización, menciona que se encontró anomalías en el 30% de los mantenimientos periódicos realizados a los equipos de comunicaciones. En el mismo estudio menciona que al utilizar técnicas de aprendizaje autónomo para predecir el momento adecuado para realizar mantenimiento predictivo en quipos, se detectaron anomalías en el 90% de los mantenimientos realizados. (Canizo et al., 2017<sup>a</sup>, p. 1).

Según IBM, la utilización de técnicas de mantenimiento combinado con aprendizaje autónomo permite un incremento de 28.3% en la eficiencia del mantenimiento, una reducción del 20.1% en el tiempo de inactividad de equipos y un ahorro de 19.4% en el costo de equipos y materiales que son reemplazados por falta de mantenimiento adecuado (IBM Services, 2019).

La utilización de aprendizaje autónomo y análisis de datos puede ser aplicado para la creación de algoritmos que sirvan para predecir el momento de realizar el mantenimiento en equipos industriales y de telecomunicaciones. Estos algoritmos

se basan en analizar y medir el desgaste de los elementos para determinar, con alto grado de precisión, el momento en que podría ocurrir una falla (Ageeva, 2018).

Al realizar un análisis de datos de eventos y alarmas generados por los equipos inalámbricos dentro de un ISP utilizando aprendizaje autónomo, permite conocer información valiosa sobre el estado de los equipos y facilita encontrar resultados interpretativos para la toma de decisiones estratégicas, brindando ventajas como reducción de costos de mantenimiento, reducción de fallas y reducción de daños en equipos (Carvalho et al., 2019).

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

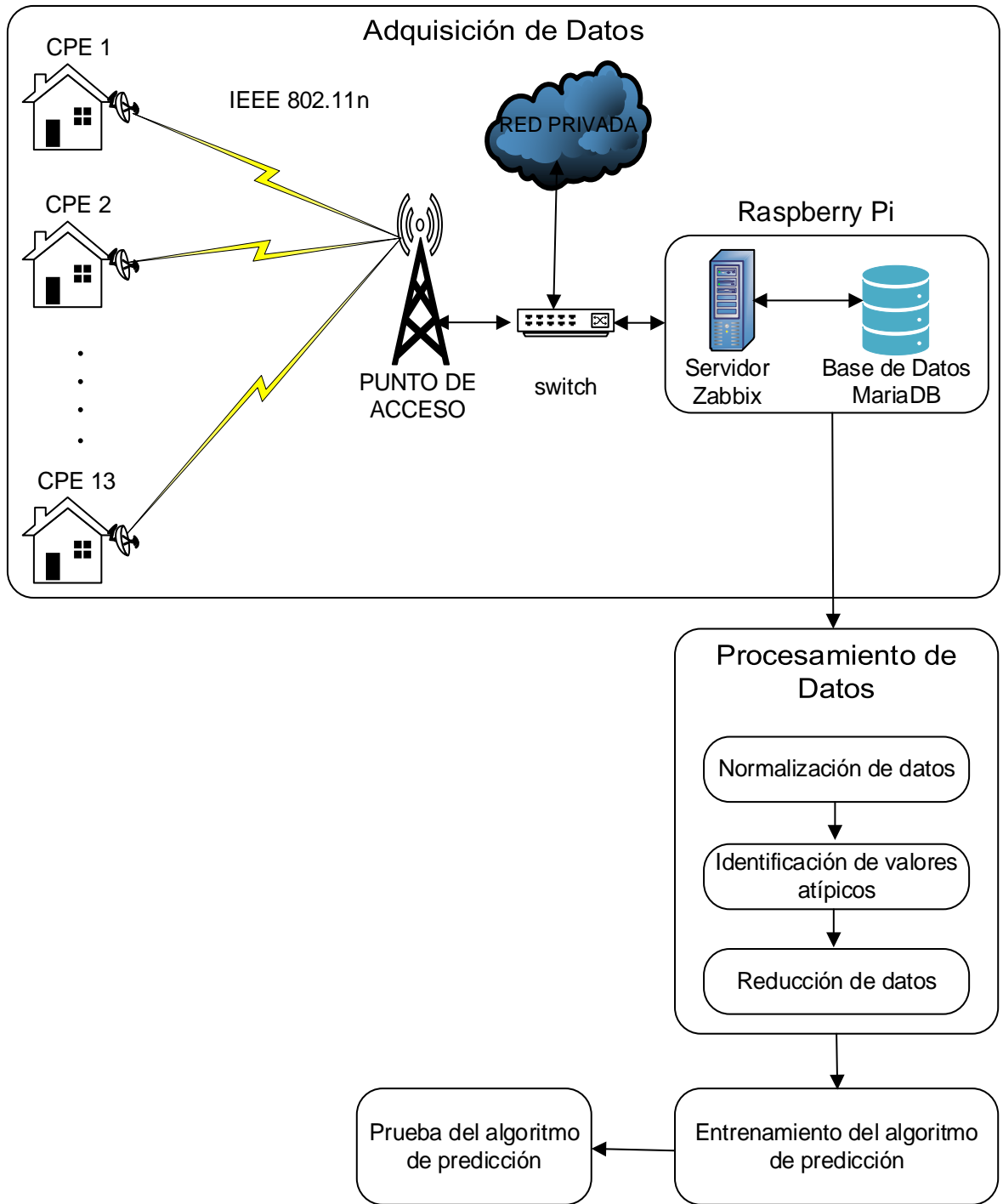
Implementar un algoritmo para la detección temprana de fallas en equipos inalámbricos de una empresa proveedora de servicios de Internet utilizando técnicas de aprendizaje autónomo.

#### **1.3.2 Objetivos Específicos**

1. Implementar un prototipo de adquisición de datos de los equipos inalámbricos de una empresa proveedora de Internet utilizando Zabbix.
2. Procesar los datos históricos obtenidos de los equipos de comunicación inalámbricos para la detección de patrones y tendencias en los datos.
3. Evaluar los diferentes algoritmos de aprendizaje autónomo para la selección del algoritmo más eficiente que se ajuste a la solución.

### **1.4 Propuesta de la solución**

Una vez identificado el problema, se plantea implementar un algoritmo para la detección temprana de fallas en equipos inalámbricos de una empresa proveedora de servicios de Internet utilizando técnicas de aprendizaje autónomo.



**Figura 1.1 Esquema de la solución**

La solución se divide en cuatro etapas, como se puede observar en la Figura 1.1. Los mismos se detallan a continuación:

## **1. Adquisición de datos**

En la primera etapa se implementa el sistema para la adquisición de datos históricos de los equipos inalámbricos de un ISP. Se hace uso de la herramienta Zabbix para adquirir los datos de los equipos utilizando protocolo simple de administración de red o SNMP (Olups, 2016). Todos los datos se almacenarán en una base de datos relacional.

## **2. Procesamiento de datos**

Se identifican los valores atípicos de los datos y cierto nivel de ruido que debe ser eliminado, para obtener los datos normalizados y listos para que puedan ser procesados o utilizados por los diferentes algoritmos. También se etiqueta los datos de acuerdo con los tres estados de funcionamiento que son Correcto, Advertencia y Fallo. El conjunto de datos etiquetados se divide en dos subconjuntos, el primero se utiliza para el entrenamiento de los algoritmos de aprendizaje autónomo y el segundo grupo de datos para las pruebas de validación.

## **3. Entrenamiento del algoritmo de predicción**

Una vez procesado los datos, el próximo paso es reducirlos a un conjunto de características que se puedan utilizar como indicadores de condición del estado del equipo. Se realiza varias simulaciones para entrenar los diferentes algoritmos, utilizando el conjunto de datos para entrenamiento. Luego de haber entrenado los diferentes algoritmos, se analiza cuál tiene el mejor rendimiento y ajuste a los requerimientos de la solución.

## **4. Prueba del algoritmo de predicción**

En esta etapa se realiza la validación del algoritmo seleccionado en la etapa 3, para la predicción de fallas en los equipos de comunicación inalámbricos y se utiliza el conjunto de datos de validación.

## **1.5 Alcance del trabajo**

El presente trabajo explora las diferentes técnicas de aprendizaje autónomo para detectar las fallas en equipos inalámbricos de la empresa proveedora de servicio de Internet y predecir el momento de realizar mantenimiento a los equipos inalámbricos.

El trabajo abarca los equipos de comunicación inalámbricos de un ISP. Creando un escenario de estudio con una configuración punto a multipunto formado por un Punto de Acceso (AP, por sus siglas en inglés) y los CPE, de los cuales se obtienen los datos históricos de los parámetros de los equipos para entrenar los diferentes algoritmos de aprendizaje autónomo y seleccionar el que mejor se adapta a las necesidades de la empresa.

# CAPÍTULO 2

## 2. MARCO TEÓRICO

### 2.1 Técnicas de mantenimiento de equipos de telecomunicaciones

En las empresas de telecomunicaciones es posible la implementación de diversos tipos de mantenimiento para lograr un correcto funcionamiento de sus instalaciones, herramientas y equipos.

La definición de mantenimiento según Tomas Vázquez (Vázquez, 2014, p. 29) menciona que “Es un conjunto de acciones organizadas y dirigidas, inmediatas, ocasionales o periódicas que se ejecutan para mantener en estado óptimo la imagen y la funcionalidad de un cuerpo productivo”. Al realizar mantenimiento se tiene varios objetivos como, disponibilidad continua de equipos, mantener el correcto funcionamiento todo el tiempo, brindar un servicio de calidad, evitar pérdidas económicas a las empresas y prolongar la vida útil de los equipos. Las empresas de telecomunicaciones que pretenden ser competitivas en el mercado deben adaptar su visión a un mantenimiento moderno y eficiente (Holek, 2002).

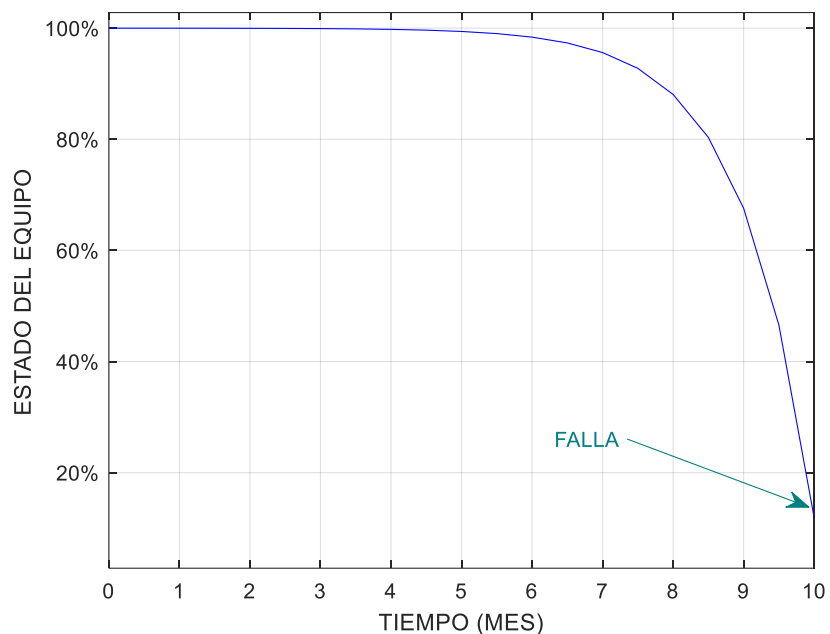
En la industria de las telecomunicaciones se suelen aplicar diversos tipos de mantenimiento, que brindan diferentes características entre los principales se mencionan a continuación:

#### 2.1.1 Mantenimiento correctivo

Al realizar este tipo de mantenimiento, los equipos se utilizan al límite y las reparaciones se realizan solo después que la máquina o equipo falla por el uso, desgaste de piezas u otros factores externos. Es el primer tipo de mantenimiento implementado en la industria, pues no necesita de ninguna clase de estudio y se lo realiza a medida que el equipo falle. Puede causar pérdidas económicas a las empresas pues su utilización conlleva costos altos de implementación (González, 2018).

Un ejemplo de esta técnica es el cambio de una bombilla pues al ser un equipo económico el enfoque remplazarlo cuando falle puede tener sentido. Pero en sistemas de telecomunicaciones se cuenta con equipos costosos que necesitan operar las 24 horas al día los 7 días de la semana, y no se puede arriesgar a que fallen de forma inesperada.

En la Figura 2.1 se puede observar la representación del ciclo de mantenimiento correctivo de un equipo. Luego de un determinado tiempo de operación falla de forma inesperada siendo el instante en el que se realiza el mantenimiento. En la actualidad el mantenimiento correctivo se recomienda utilizarlo en sistemas y equipos secundarios o de backup, que no afecten al normal funcionamiento del sistema (Primero et al., 2015).



**Figura 2.1 Mantenimiento correctivo**

El mantenimiento correctivo presenta varias ventajas y desventajas como se lista en la Tabla 2.1.

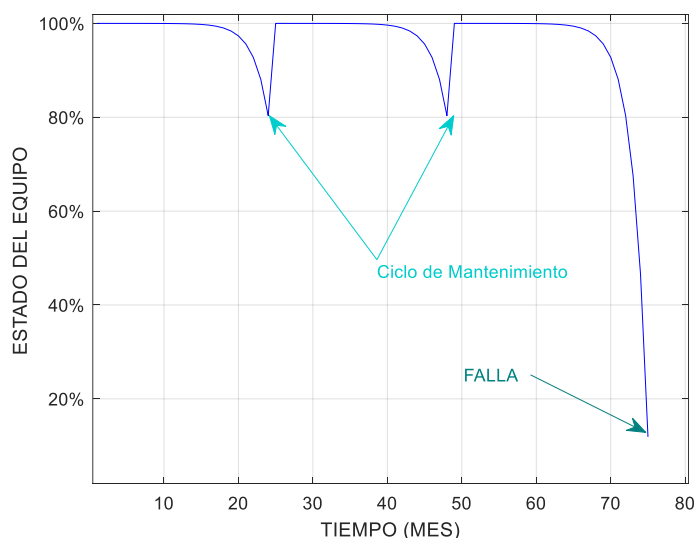
**Tabla 2.1 Ventajas y desventajas del mantenimiento correctivo**

Mantenimiento Correctivo	
Ventajas	Desventajas
Planificación mínima	Impredecible
	Equipos no maximizados
Proceso simple	Pérdida económica
	Tiempo de inactividad de equipos alta

### 2.1.2 Mantenimiento preventivo

Las empresas de telecomunicaciones intentan prevenir las fallas antes que ocurran. El mantenimiento preventivo se realiza con el fin de evitar mantenimientos no planificado concentrándose en incrementar la disponibilidad de los equipos y disminuir los costos operacionales por daños (Vidal, 2021).

Normalmente se lo realiza de forma periódica y sistemática en los equipos, aunque estos aun no hayan dado señales de desgaste o error. Un gran desafío de esta técnica de mantenimiento es determinar cuándo realizarlo, debido a que no es posible determinar de forma exacta cuando es probable que ocurra la falla (González, 2018).



**Figura 2.2 Mantenimiento preventivo**



En la Figura 2.2 se aprecia el comportamiento de un equipo en el tiempo aplicando mantenimiento preventivo. Luego de un determinado tiempo de operación se realiza el mantenimiento al equipo sin que el mismo haya fallado o presente algún desgaste. Se puede tener varios ciclos de mantenimiento antes que el equipo falle.

Con esta técnica las principales actividades que se realizan durante el mantenimiento son: limpieza, reajuste y reemplazo de piezas o componentes del equipo, independientemente del estado en el que se encuentre. Las desventajas y desventajas de esta técnica se listan en la Tabla 2.2.

**Tabla 2.2 Ventajas y desventajas del mantenimiento preventivo**

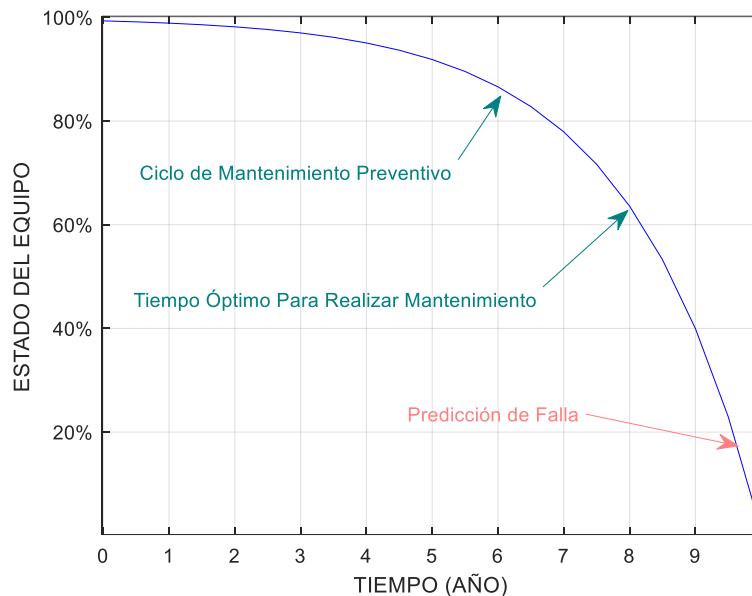
Mantenimiento Preventivo	
Ventajas	Desventajas
Reducción de costos	Dificultad para determinar el nivel de depreciación o desgaste de los equipos
Reducción de tiempo de inactividad de equipos	
Aumenta la vida útil de los equipos	Se necesita personal con experiencia para aplicar las recomendaciones de los fabricantes.
Mejora la calidad de los productos y servicios	

### 2.1.3 Mantenimiento predictivo

Se desarrolló como un complemento para los mantenimientos correctivo y preventivo. Ayuda a estimar el tiempo en que fallará un equipo y permite determinar el momento óptimo para programar el mantenimiento. No solo predice una falla futura, sino también ayuda a identificar problemas y determinar la vida útil en los equipos (Canizo et al., 2017b).

En la Figura 2.3 se muestra un gráfico de tendencia del comportamiento de mantenimiento predictivo de un equipo. En el mismo se observa que se

trata de alargar el ciclo del mantenimiento y predecir el instante óptimo para realizarlo y así evitar que el equipo falle.



**Figura 2.3 Mantenimiento predictivo**

En este tipo de mantenimiento se monitorea una serie de parámetros los cuales son analizados utilizando aprendizaje autónomo para descubrir si los parámetros cambian, es más tecnológico, pues requiere de conocimientos matemáticos, físicos y técnicos. Con esta técnica los equipos son intervenidos antes que fallen, pero no antes de que realmente se necesite el mantenimiento (Ageeva, 2018).

Al utilizar esta técnica de mantenimiento se obtiene una gran cantidad de datos que permite generar o crear modelos matemáticos, para representar de manera precisa el comportamiento del equipo (Kanawaday & Sane, 2017).

El mantenimiento predictivo se podría considerar como un mantenimiento preventivo forzado por algún evento, por tal razón no se considera el tiempo de uso del equipo para programar el mantenimiento, sus principales ventajas y desventajas se listan en la Tabla 2.3 (Ageeva, 2018).

**Tabla 2.3 Ventajas y desventajas del mantenimiento predictivo**

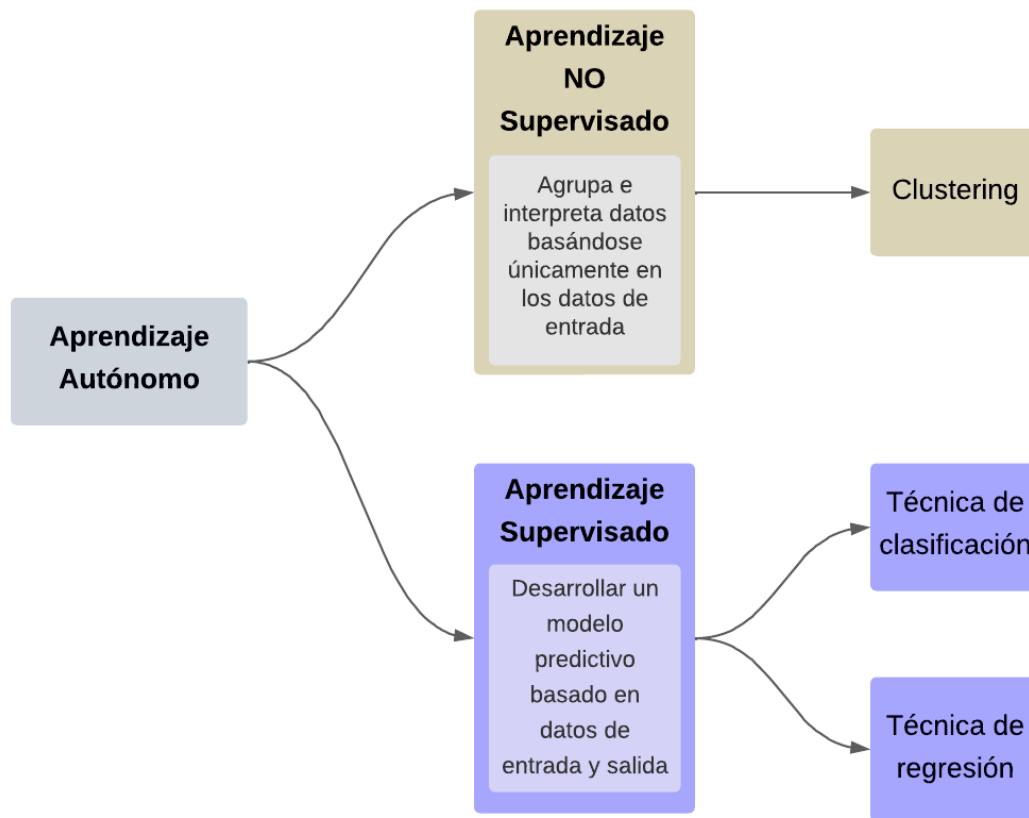
Mantenimiento Predictivo	
Ventajas	Desventajas
Ahorro en costos de repuestos	Personal especializado
Mayor confiabilidad	Requiere inversión inicial alta
Menos costo de mantenimiento	
Prolonga la vida útil de los equipos	
Tiempo de inactividad de equipos mínima	

## 2.2 Técnicas de aprendizaje autónomo

Es una disciplina del campo de la inteligencia artificial que utiliza análisis de datos, para dotar a los ordenadores la capacidad de hacer cosas que, para las personas, resulta natural mediante la utilización de algoritmos. Los algoritmos de aprendizaje autónomo hacen uso de métodos de cálculo para “aprender” directamente de los datos sin utilizar o depender de alguna ecuación predeterminada como modelo, son capaces de analizar gran cantidad de datos para predecir un resultado (Bonaccorso, 2017).

El aprendizaje autónomo se ha desarrollado durante los últimos años hasta convertirse en una técnica clave para solucionar problemas en diversas áreas como finanzas, informática, procesamiento de imágenes, visión artificial, generación de energía eléctrica, telecomunicaciones y mantenimiento predictivo, entre otras. Las características de todas estas aplicaciones es la gran cantidad de datos y variables que se van generando y están disponibles para su análisis. El objetivo del aprendizaje autónomo es establecer un modelo que nos permita dar solución a una tarea determinada (Alpaydin, 2021).

Estas técnicas incorporan una variedad de enfoques que permiten su clasificación como aprendizaje supervisado y no supervisado como se puede observar en la Figura 2.4. Entre las principales diferencias de estas dos técnicas de aprendizaje autónomo es la capacidad de disponer los datos etiquetados o no, para poder predecir una determinada entrada (Bonaccorso, 2017, pp. 6-19).



**Figura 2.4 Técnicas de aprendizaje autónomo**

**Nota.** Tomado de MATLAB (2020) del estudio denominado "Machine Learning: Tres cosas que es necesario saber"

### 2.2.1 Aprendizaje supervisado

Es una técnica que permite crear modelos que ayudan a realizar predicciones utilizando un conjunto de datos etiquetados para entrenar los algoritmos. El proceso de entrenamiento crea predicciones como respuesta a nuevos datos de entrada. Se utiliza en aplicaciones en las cuales se cuenta con un conjunto de datos conocidos para la salida que se intenta predecir (Brownlee, 2016).

Los diferentes algoritmos parten de un conjunto de datos de entrenamiento. Al conocer las respuestas correctas, el algoritmo iterativamente realiza predicciones sobre el conjunto de datos de entrenamiento y finaliza su

proceso de aprendizaje cuando logran alcanzar un determinado nivel de rendimiento (Simeone, 2018).

El aprendizaje supervisado se puede agrupar en técnica de clasificación y regresión para desarrollar modelos predictivos, las mismas se explican a continuación:

### **Técnica de clasificación**

Se utiliza cuando las variables de salida son una respuesta discreta o una categoría. Las técnicas de clasificación organizan los datos de entrada en categorías, entre las principales aplicaciones se encuentran: gestión de recursos en comunicaciones satelitales, despliegue de redes celulares 5G, imágenes médicas, reconocimiento de voz y detección de fallas en centrales IP (Brownlee, 2020, pp. 15-18).

Las técnicas de clasificación se utilizan si los datos se pueden etiquetar, categorizar o dividir en grupos o clases concretos. Existen varios algoritmos que emplean técnicas de clasificación para la detección de fallas con una precisión mayor al 85% (Kumbhar et al., 2018).

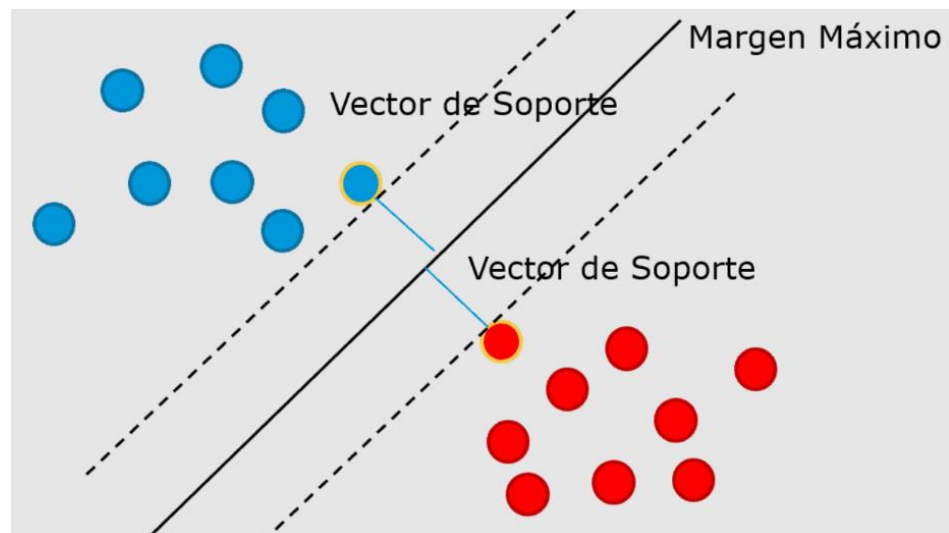
### **Técnica de regresión**

Permite predecir respuestas continuas como variación de temperatura, fluctuación del consumo energético o tiempo que tardará una pieza o equipo en fallar. Se utilizan cuando se trabajan con intervalos de datos o si la naturaleza de la respuesta es un número específico (Brownlee, 2020, pp. 15-20).

Existen varios algoritmos que utilizan en aprendizaje supervisado para la detección de fallas entre los más utilizados y que mejor precisión ofrecen se mencionan a continuación (Camejo Corona et al., 2019):

- **Máquina de vectores de soporte**

Es un algoritmo que puede ser utilizado en problemas de clasificación o regresiones. Es necesario definir un hiperplano que es el límite de decisión, de modo que separe de la mejor forma posible dos clases diferentes. La separación entre dos clases se representa en el hiperplano con signos más y menos cómo se aprecia en la Figura 2.5. El vector de soporte indica el subconjunto de datos de entrenamiento que determinan la ubicación del hiperplano de separación (Cuenca & Leon, 2012).



**Figura 2.5 Estructura del algoritmo Support Vector Machines**

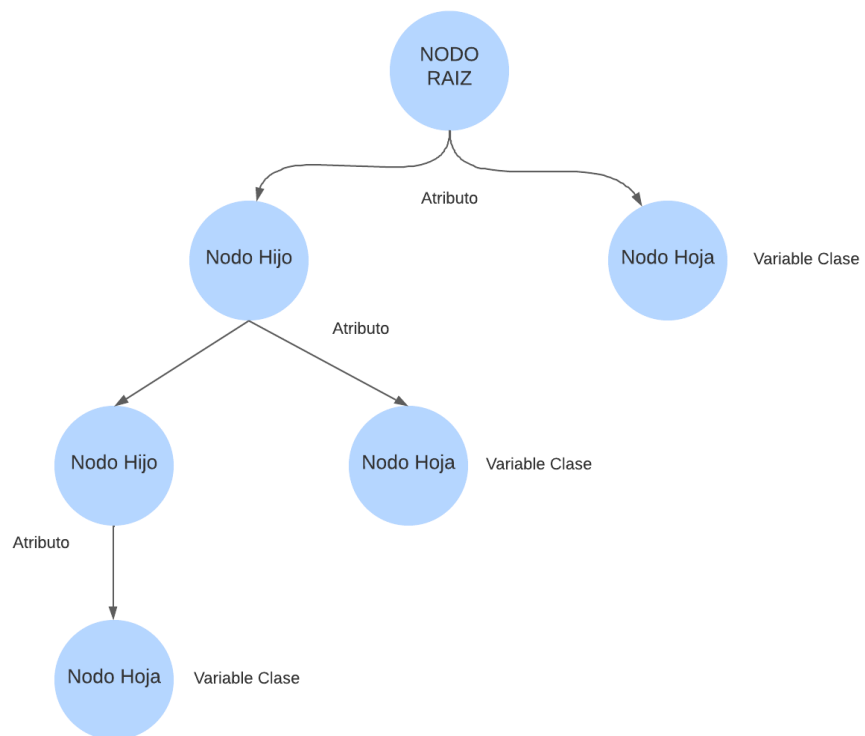
**Nota.** Tomado de Cuenca D. (2012) del estudio denominado "SUPPORT VECTOR MACHINE"

Los algoritmos Máquina de Vectores de Soporte (SVM, por sus siglas en inglés) presentan varias ventajas como facilitar el manejo de datos semiestructurados y estructurados. Es menos probable que se genere un sobre ajuste y permite el manejo de funciones complejas utilizando la función de Kernel.

La desventaja de los algoritmos SVM es la disminución del rendimiento con una gran cantidad de datos debido al incremento del tiempo necesario para el entrenamiento y son sensible a datos con ruido.

- **Árbol de decisiones**

El proceso de aprendizaje de este algoritmo se representa mediante un árbol como se representa en la Figura 2.6. Este árbol está formado por nodos raíz, hojas y ramas. El nodo raíz es el atributo a partir del cual se da inicio el proceso de clasificación o regresión. Las respuestas a cada pregunta se representan mediante un nodo hijo. Las ramas que salen de cada nodo están etiquetadas con cada posible valor del atributo. Los nodos hojas representan la decisión que toma el algoritmo, la misma coincide con la variable clase del problema al cual se está buscando solución (Navada et al., 2011).



**Figura 2.6 Estructura del algoritmo de árbol de decisión**

**Nota.** Tomado de Navada (2011) del estudio denominado " Overview of use of decision tree algorithms in machine learning"

Los algoritmos de árbol de decisión presentan varias ventajas como la facilidad de interpretación, puede completar valores faltantes con el valor más probable, presenta un alto rendimiento, facilita el manejo de

valores cuantitativos y categóricos. La desventaja es que tienden al sobreentrenamiento, cambios pequeños en los datos de entrada puede causar inestabilidad y se necesita conocer a profundidad los conceptos del algoritmo para tener un óptimo funcionamiento.

- **K Vecinos más próximos**

Posiblemente es el algoritmo de aprendizaje autónomo supervisado más simple y eficiente para su implementación. Utiliza un conjunto de datos agrupados en varias clases para el entrenamiento, el algoritmo intenta realizar una predicción para nuevos datos de entrada, simplemente devuelve los puntos de datos más similares o cercanos aprendidos en la etapa de aprendizaje.

La letra K indica el número de vecinos más cercanos que se tomará en cuenta para clasificar un dato. Un valor de K alto causará una predicción sin muchos cambios de un punto a otro lo que podría causar un subentrenamiento, mientras que un valor de K bajo causará que la predicciones se acerque mucho a los datos de entrenamiento (Ray, 2019).

El algoritmo K-Vecinos más próximos se ejecuta en tres etapas. Primero, se calcula la distancia entre los datos a clasificar y los datos utilizados para el entrenamiento del algoritmo. Segundo, se debe seleccionar el valor de K de los vecinos más cercanos. Tercero, se realiza una votación de mayoría entre los K puntos para seleccionar a la clase con la que se clasificara.

- **Naive Bayes**

Está basado en la probabilidad condicional y la aplicación del teorema de Bayes. En este algoritmo existe una tabla de probabilidad que sirve de modelo y se actualiza mediante el entrenamiento. La tabla de probabilidad depende de los valores característicos en los que es



necesario buscar las probabilidades de cada clase para predecir cada nuevo dato (Chen et al., 2020).

Los algoritmos Naive Bayes requiere de menos datos para el entrenamiento, son fácil de implementar, son resistentes al ruido, puede manejar datos continuos y discretos. La desventaja de este algoritmo es el no poder ser utilizado con aplicaciones que tienen una gran cantidad de clases debido a que utilizan una gran cantidad de recursos.

- **Regresiones Lógicas**

Estos algoritmos se utilizan para resolver problemas de clasificación, el cual se basa en métodos probabilísticos. En se calcula la probabilidad de que un dato de entrada pertenezca a una categoría apropiada. La salida en este tipo de algoritmo es discreta (Dreiseitl & Ohno-Machado, 2002).

Las regresiones lógicas presentan las siguientes ventajas: son fáciles de implementar, muy eficientes en el uso de recursos computacionales, no necesitan muchos datos para su entrenamiento y no se ven afectadas por ruido en los datos. Las principales desventajas son: no se puede aplicar en problemas que sus datos no sean lineales y no funcionan correctamente si no se identifican todas las variables independientes.

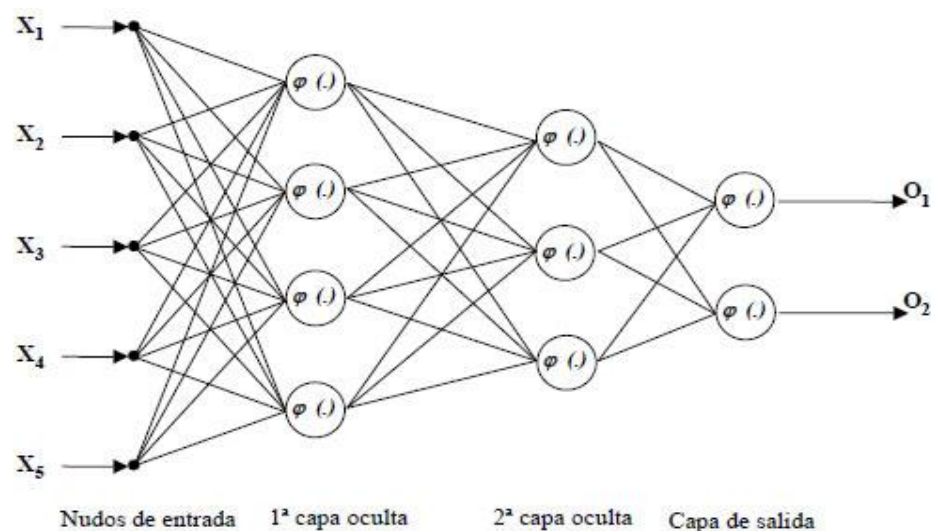
- **Regresiones Lineales**

Está basado en métodos estadísticos y se utiliza para encontrar la relación entre dos variables continuas. En el algoritmo de regresiones lineales tenemos un conjunto de datos de entrada etiquetados y el valor de la variable de salida depende del valor de los datos de entrada. Este algoritmo intenta obtener una línea que mejor se ajuste al conjunto de datos. La línea que más se ajuste es aquella en la que el error de predicción total es el menos posible (Ray, 2019).

Presentan varias ventajas como fáciles de aplicar el modelo, permite aplicar modelos estadísticos y procesamiento de datos con aprendizaje autónomo. Las principales desventajas son: no se recomienda para tratar problemas de la vida real por su simplicidad. De igual manera no se puede manejar patrones complejos y al realizar ajuste a una línea de regresión se pierden datos que causarían errores al momento de realizar predicciones.

- **Redes neuronales**

Estos algoritmos están basados en un mecanismo de aprendizaje que replica en el comportamiento del cerebro humano, donde existen millones de neuronas que se interconectan en forma de red para intercambiar mensajes una con otra. Establece relaciones no lineales entre los datos de entrada y la variable de salida. En la Figura 2.7 se indica la estructura de una red neuronal así como los elementos que la confirman (Sánchez et al., 2016).



**Figura 2.7 Estructura de una red neuronal**

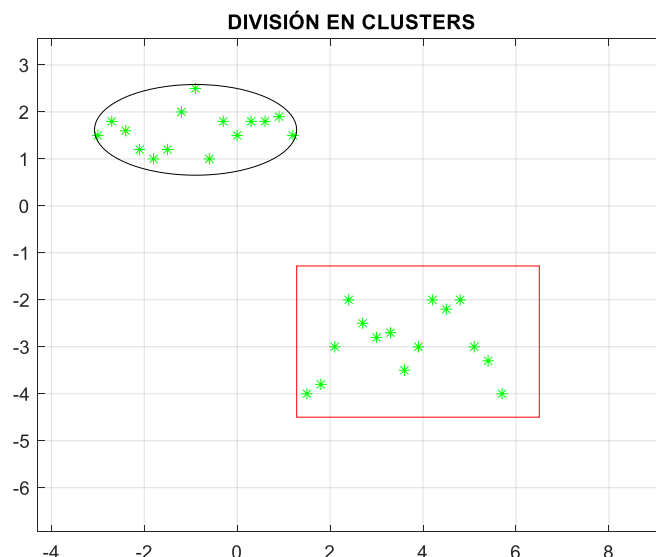
**Nota.** Tomado de Villada (2016) del estudio denominado "Redes Neuronales Artificiales aplicadas a la Predicción del Precio del Oro"

Las variables  $X_i$  son los datos de entrada que no sufren ninguna alteración,  $O_i$  son los datos de salida que se predicen mediante el algoritmo,  $\phi$  es la función de activación y las capas ocultas se encargan de recibir los datos procedentes de las capas anteriores procesar y enviar los datos a la siguiente capa hasta llegar la salida.

La principal ventaja de este tipo de algoritmo es que facilita el análisis y procesamiento de información con patrones complejos. La desventaja es que al estar basado en el cerebro humano son lentos de entrenar y necesitan mucha capacidad de cómputo.

### 2.2.2 Aprendizaje no supervisado

Este tipo de aprendizaje se caracteriza por no disponer de datos etiquetados. Permite encontrar patrones ocultos o estructuras intrínsecas en los datos. Se utiliza para descubrir información a partir de un grupo de datos de los cuales se conoce la entrada, pero no existen datos de salida que correspondan a una determinada entrada (Bonaccorso, 2017).



**Figura 2.8 Agrupación por clusters**

En el aprendizaje no supervisado la principal técnica utilizada es el clustering. Esta se utiliza para analizar datos exploratorios, encontrar similitudes o agrupaciones ocultas en los datos, pero nada garantiza que estas agrupaciones tengan significado o utilidad. El clustering es el proceso de dividir un conjunto de datos en varias subclases llamadas grupos (clusters). Cada agrupación corresponden a datos similares unos con otros en la Figura 2.8 se representa la división por clusters (Brownlee, 2016).

Los algoritmos que utiliza clustering presentan varias ventajas como: son fáciles de escalar y no necesitan gran cantidad de datos. Pueden ser utilizados con diferentes atributos como binarios y numéricos nominales. Permite introducir restricciones y es resistente al ruido en los datos y puede operar eficientemente con alta dimensionalidad.

Las principales aplicaciones de clusters se encuentran la secuenciación genética, análisis de mercado y reconocimiento de objetos. Los algoritmos más utilizados con clustering se indican a continuación (Ray, 2019):

#### ➤ **K-Means**

Es uno de los algoritmos de aprendizaje no supervisado más utilizado por ser fácil de implementar y muy eficiente en problemas de agrupamiento. Este algoritmo opera de forma iterativa, divide de manera óptima el conjunto de datos iniciales en un número de K clúster basando en sus características de similitud. El proceso de ejecución del algoritmo K-Means se define a priori el número de agrupaciones a formar (Cáceres, 2015).

Presenta varias ventajas como: es altamente eficiente en el uso computacional, fácil de interpretar e implementar y facilita el manejo de datos que contenga gran cantidad de variables. Las desventajas del algoritmo son: resulta complicado determinar el valor de K apropiado, son sensible a datos que presentan una desviación grande porque

distorsionan la distribución de datos, su rendimiento se ve afectado si los datos de entrenamiento no tienen el mismo tamaño.

### ➤ **Análisis de Componentes Principales**

El Análisis de Componentes Principales (PCA), es un algoritmo tipo estadístico que trata de establecer una estructura a un conjunto de datos contenidos en una matriz. El proceso consiste en homologar la matriz de datos a un espacio vectorial tratando de encontrar los ejes o dimensiones que serán combinación de las variables introducidas (García-Alvarez & Fuente, 2011).

El PCA es una técnica que facilita la extracción de características mediante la combinación de las entradas de forma que se pueda eliminar alguna de las variables de menos importancia y conservando las variables de mayor importancia. Este algoritmo se ejecuta en los siguientes pasos (García-Alvarez & Fuente, 2011):

1. Estandariza los datos de entrada.
2. Obtiene la matriz de covarianza.
3. Ordena los autovectores y autovalores de mayor a menor y selecciona el número de dimensiones (K).
4. Construye la matriz de proyección con el valor de K.
5. Transforma el dataset original para obtener las nuevas características.

## **2.3 Métricas de evaluación de los algoritmos de clasificación**

Existen varias métricas que se pueden utilizar para la comparación y evaluación del rendimiento de algoritmos de aprendizaje autónomo. Estas técnicas están basadas en etiquetas booleanas que se asignan a uno o varios datos. Al finalizar el entrenamiento de los algoritmos su salida puede ser una respuesta booleana o una salida numérica (Lu & Lysecky, 2017).

### 2.3.1 Matriz de confusión

Es la forma más simple que permite evaluar el desempeño de los algoritmos de clasificación donde existen salidas de dos o más variables. Permitiendo indicar la precisión y exactitud del modelo a entrenado.

La Figura 2.9 representa la forma básica de la matriz de confusión, la fila Positivo representa a los elementos que se han clasificado como positivos, la fila Negativo representa a los elementos que se han clasificado como negativos. La columna Positivo indica los elementos que en realidad son positivos, la columna Negativo indica los elementos que en realidad son negativos (*Matriz de confusión | Interactive Chaos, s. f.*).

		DATO REAL	
		Positivo	Negativo
PREDICCIÓN	Positivo	Verdadero Positivo	Falso Positivo
	Negativo	Falso Positivo	Verdadero Negativo

**Figura 2.9 Matriz de confusión**

De la matriz indicada en la Figura 2.9, se deduce cuatro clasificaciones:

1. Verdadero Positivo: La clasificación es correcta, se predice como positivo y es positivo.
2. Falso Positivo: La clasificación es incorrecta, se predice como positivo y es falso.
3. Falso Negativo: La clasificación es incorrecta, se predice como negativo y falso.
4. Verdadero Negativo: La clasificación es correcta, se predice como negativo y es negativo.

Para poder interpretar la matriz de confusión se utiliza varias métricas entre las principales se menciona a continuación (Powers, 2020):

- **Exactitud.** Indica la cercanía entre los datos arrojados por los algoritmos de clasificación y los datos reales. Mientras mayor sea la coincidencia la matriz permite interpretar los datos de forma exacta.
- **Precisión.** Se refiere a lo cercano que se encuentran los valores entregados por los algoritmos de clasificación al valor verdadero. Es el grado de concordancia entre los resultados.
- **Sensibilidad.** Permite determinar la capacidad del algoritmo de clasificación, para detectar casos positivos.
- **Especificidad.** Expresa la tasa de aciertos que los algoritmos tiene con respecto a casos negativos. Es la relación entre la cantidad de casos negativos detectados con la cantidad de casos negativos totales.

### 2.3.2 Curva ROC

La métrica Características de Funcionamiento del Receptor (ROC, por sus siglas en inglés), es una gráfica que permite evaluar la eficiencia de cualquier modelo de clasificación. Nos indica que tan bueno es el algoritmo en distinguir entre dos cosas. Los algoritmos más eficientes pueden distinguir con mayor precisión entre dos variables, mientras que un algoritmo ineficiente tendrá problemas para distinguir entre dos clases de variables (Cerdeira & Cifuentes, 2012).

La curva ROC es una gráfica bidimensional que en su eje Y, se representa la sensibilidad y el eje X se compone de 1-Especificidad. La medida de comparación en esta gráfica es el área bajo la curva y sus valores se encuentran entre 0 y 1 (Narkhede, 2021):

- Valores cercanos a 0 el modelo se encuentra reciprocando las clases, tiene la peor medida de separabilidad. Esto quiere decir que está prediciendo 0 como 1 y 1 como 0.
- Valores cercanos a 0,5 indica que el modelo no cuenta con la capacidad de discriminación suficiente para poder distinguir entre clase positiva y clase negativa siendo la peor situación.
- Valores cercanos a 1 el modelo es muy bueno, tiene una buena medida de separabilidad. En este caso el modelo puede diferenciar entre diferentes clases.

## **2.4 Sistemas de adquisición de datos**

Es un conjunto de herramientas y procesos que permiten realizar mediciones de fenómenos físicos y registrarlos para su posterior análisis. Entre los principales componentes que forman un sistema de adquisición de datos se indican a continuación (Timmermann, 2013):

- Sensor: permite convertir los parámetros físicos en señales eléctricas.
- Circuito de adiciónamiento de señal: facilita convertir las señales provenientes del sensor en valores digitales.
- Conversor analógico-digital: encargado de convertir las señales de los sensores acondicionados en valores digitales.

### **2.4.1 Sistemas de monitoreo de red**

El continuo avance tecnológico facilita la implementación de grandes sistemas de monitoreo dentro de una red de telecomunicaciones, proporcionando información del estado de la red. Esta constituidos por un conjunto de herramientas de software y hardware que permiten dar seguimiento a los diversos parámetros de la red y su funcionamiento. El término monitoreo de red indica la utilización de uno o varios sistemas, que de forma continua vigilan una red de telecomunicaciones permitiendo identificar componentes defectuosos o equipos que están operando fuera



de parámetros. Entre sus principales ventajas se mencionan a continuación (Delgado et al., 2016):

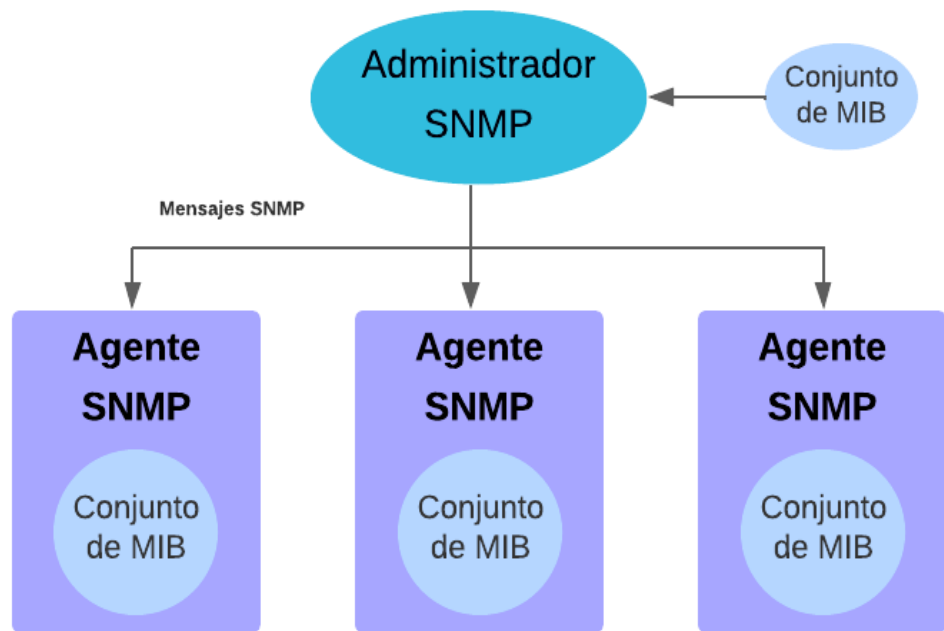
- Reducción del tiempo empleado para solucionar problemas.
- Optimiza el uso de recursos de empresas.
- Utilización al máximo de los dispositivos dentro de una red.
- Permite tener un control global de la red.
- Mantener un histórico de datos de funcionamiento de los dispositivos dentro de la red.

#### 2.4.2 Protocolo Simple de Administración de Red

Es un protocolo de capa de aplicación definido en el estándar RFC-1157 que permite realizar el monitoreo de dispositivos y servicios de red. El Protocolo Simple de Administración de Red (SNMP, por sus siglas en inglés), hace uso de un sistema de llamada y respuesta para comprobar el estado de los dispositivos. No es un protocolo propietario, por tal razón puede ser utilizado para monitorear dispositivos de diferentes fabricantes.

Se basa en una estructura cliente-servidor que se utiliza en redes TCP/IP. La estructura del protocolo SNMP se representa en la Figura 2.10 en la misma se definen los siguientes componentes (Fedor et al., 1990):

- **Agente SNMP.** Es un software de administración que se encuentra dentro de un dispositivo de red que facilita el acceso a la información monitoreada de cada dispositivo, dicha información está disponible para que el administrador SNMP pueda accederla.
- **Administrador SNMP.** Es el equipo central que administra o gestiona los procesos de monitoreo de toda red, a través del software se encarga de recibir y enviar los mensajes a los agentes SNMP. Entre sus funciones están, realizar consultas, reconocer eventos o alarmas, recibir respuestas y ejecutar configuraciones en los agentes.



**Figura 2.10 Arquitectura de SNMP**

**Nota.** Tomado de Fedor (1990) del estudio denominado "Simple Network Management Protocol (SNMP)"

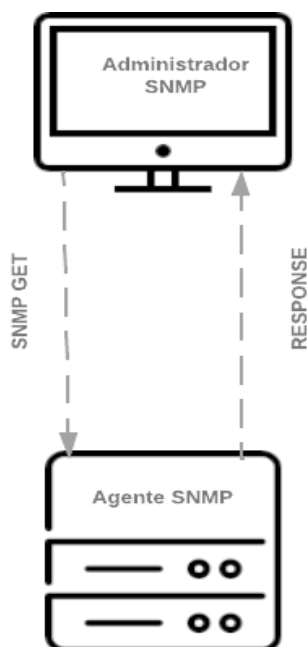
- **Dispositivos administrados.** Son todos los dispositivos dentro de una red de telecomunicaciones que requieren ser monitoreados o administrados por el administrador SNMP, generalmente estos dispositivos pueden ser enrutadores, servidores o dispositivos inalámbricos.
- **Base de Información de Gestión (MIB, por sus siglas en inglés).** Es un conjunto de datos que contiene la información necesaria para acceder a los parámetros a monitorear de cada dispositivo gestionado. Los MIB son generados por cada fabricante para facilitar el acceso a información específica de los agentes SNMP (ORACLE, 2015).

En la actualidad existen tres versiones de SNMP lo que distingue a cada una de ellas es la forma en que se implementa las medidas de seguridad como se lista en la Tabla 2.4.

**Tabla 2.4 Versiones de SNMP**

Nombre	Cambios
SNMPv1	La seguridad está basada en comunidades que utilizan contraseñas sin encriptación en texto plano, se puede acceder al dispositivo por fuerza bruta.
SNMPv2	Disminuye el tráfico para la monitorización de los agentes y permite solucionar los problemas de monitorización distribuida en las redes de telecomunicaciones. SNMPv2 puede acceder a datos de dispositivos con SNMPv1.
SNMPv3	Mejora la seguridad en las transmisión e intercambio de información con cifrado y autenticación. Proporciona compatibilidad completos con SNMPv1 y v2 mediante la adición de una cabecera adicional.

Dentro de una red se encuentra un administrador SNMP y varios dispositivos, incluidos clientes que no son agentes SNMP. La comunicación entre el administrador y los agentes se realiza mediante el envío de mensajes utilizando un esquema cliente-servidor como se indica en la Figura 2.11 esto permite la inserción o extracción de información.



**Figura 2.11 Comunicación SNMP**

**Nota.** Tomado de Fedor (1990) del estudio denominado "Simple Network Management Protocol"

Existen varios tipos de mensajes SNMP que se utilizan para el intercambio de información entre el administrador y los agentes SNMP, los mismos se indican a continuación (Briceño, 2004):

**GET:** Es un mensaje de solicitud enviada por el administrador al agente SNMP. Se utiliza para recuperar uno o más valores de los agentes.

**GET NEXT:** Es similar al GET, la diferencia es que recupera el valor del siguiente OID en el árbol MIB. Se inicia en OID 0, el administrador puede enviar una solicitud hasta recuperar todos los datos OID.

**GET BULK:** Es un mensaje desarrollado en SNMPv2, es una versión optimizada de GET NEXT. Es utilizado para recuperar una gran cantidad de datos de tabla MIB grande con la utilización de un solo mensaje.

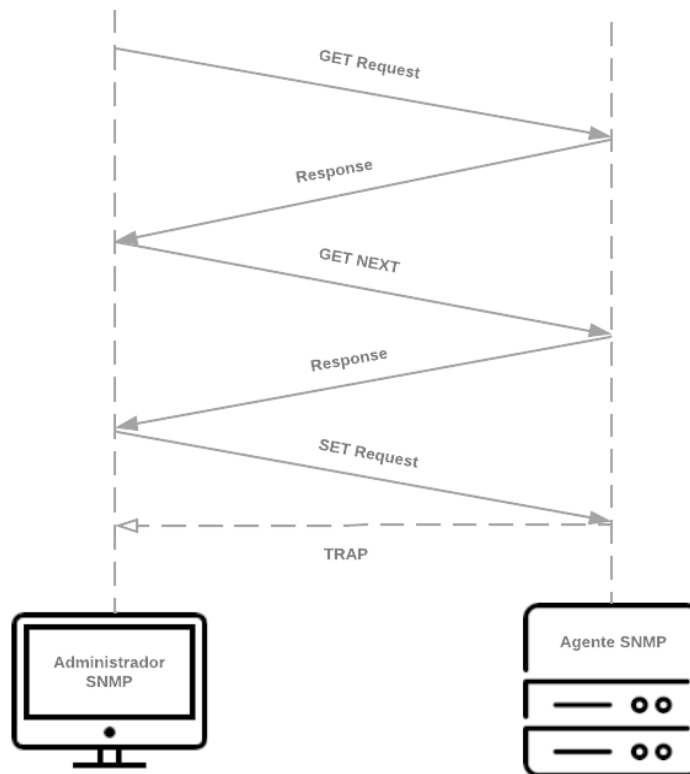
**SET:** Es un mensaje enviado por el administrador SNMP, este mensaje es utilizado para modificar o asignar uno o varios valores en los agentes SNMP.

**TRAPS:** Son mensajes enviados por los agentes SNMP sin que el administrador SNMP lo solicite. Es una petición al administrador SNMP por parte del agente cuando ocurre un evento determinado, como un error o sobrepasar un umbral.

**INFORM:** Fue incorporado en SNMPv2, es iniciado por el agente SNMP permite al administrador SNMP confirmar que se ha recibido un mensaje.

**RESPONSE:** Es el mensaje utilizado para transportar los valores solicitados por el administrador SNMP.

En la Figura 2.12 se muestra el proceso de intercambio de mensajes entre un administrador y el agente SNMP, para establecer la comunicación utiliza el puerto UDP 161 para el intercambio de mensajes y el puerto UDP 162 para el envío de TRAPS.



**Figura 2.12 Intercambio de mensajes SNMP**

**Nota.** Adaptado de Briceño (2004) del estudio denominado " PROTOCOLO SNMP (PROTOCOLO SENCILLO DE ADMINISTRACIÓN DE REDES)

### 2.4.3 Herramientas para monitoreo de redes de telecomunicaciones

El avance de la tecnología ha facilitado el desarrollo de diversas aplicaciones para el monitoreo de redes de Telecomunicaciones basadas en software libre y versiones pagadas. En el mercado se puede encontrar diversas herramientas para el monitoreo de redes como se mencionan a continuación:

#### **Zabbix**

Es un software de código abierto para monitorización en tiempo real de servidores, dispositivos de telecomunicaciones y aplicaciones web. Además, ofrece características avanzadas de monitoreo, alertas, almacenamiento y visualización en un solo software (Zabbix, 2022).

Zabbix facilita el monitoreo de diversos agentes o dispositivos dentro de una red. Utiliza el protocolo SNMP en sus tres versiones para comunicarse con los dispositivos de diferentes fabricantes. Cuenta con una interfaz web que facilita la visualización de los datos monitoreados. A continuación, se indican las principales ventajas de Zabbix:

- Reportes detallados en tiempo real.
- Fácil de configurar.
- Estadísticas en tiempo real de los diferentes agentes SNMP.
- Configuración de notificaciones.
- Perfiles de usuarios.
- Generación de gráficas.
- Zabbix permite almacenar los datos de monitoreo en bases de datos tanto relacional como no relacional.

## **Nagios**

Es un sistema de monitoreo basada en software libre, es utilizado por grandes empresas como Disney, Universal y Cisco. Cuenta con una interfaz web muy potente facilitando la configuración de notificaciones y alertas. Entre sus principales ventajas (Barth, 2008, pp. 38-45):

- Facilita la creación de plugins para cada aplicación.
- Facilita la configuración manual del sistema de monitorización.
- Permite la comunicación mediante protocolo SNMP.

## **SolarWinds**

Es un software para la administración y monitorización de redes de telecomunicaciones que permite desarrollar aplicaciones para detectar, diagnosticar, resolver rápidamente cortes y problemas de rendimiento de los equipos dentro de la red (SolarwWinds, 2022).

Permite la administración de archivos de configuración dentro de redes cableadas e inalámbricas mediante una interfaz web, facilitando la configuración de agentes SNMP y el envío de alertas inteligentes y personalizables sobre cualquier anomalía. Los cambios de configuración de los agentes SNMP se los realiza en tiempo real mejorando la seguridad en la red. Una de las principales desventajas es que no se puede integrar con aplicaciones en la nube.

### **Cacti**

Es un sistema de gestión de fallas y monitoreo de redes. Incluye un marco para recopilar datos de forma distribuida y altamente tolerable a fallas. Permite desarrollar funciones avanzadas de automatización basadas en pantallas y gráficos para cada dispositivo. Facilita la configuración de alarmas mediante umbrales. Para la comunicación con los dispositivos monitoreados puede utilizar diversos protocolos de comunicación incluido SNMP (Kharb, 2019).

### **OpenNMS**

Es un sistema de monitoreo de redes de código abierto basados en Java, siendo la primera aplicación de monitoreo de redes a nivel empresarial. Facilita la creación de soluciones interoperables, normaliza el intercambio de mensajes de los dispositivos en un solo protocolo. Se puede acceder a los datos de los diferentes agentes SNMP a través de API REST además se puede utilizar scripts en diferentes lenguajes de programación.

#### **2.4.4 Base de datos relacional**

Las bases de datos relacional permiten almacenar y acceder a los datos que están relacionados entre sí. Basan su funcionamiento en modelos relacionales e intuitivos que permiten representar los datos en tablas. Cada fila de la tabla representa un registro identificado de forma única. Cada

columna indica el atributo de los datos almacenados permitiendo la creación de relación (Cabello, 2010).

Las principales características de una base de datos relacional son:

- Evita la duplicidad de registros.
- Garantiza la integridad referencial de los datos, es decir si eliminamos un registro el resto de los registros mantienen sus valores.
- Cada tabla es considerada como un conjunto de registros relacionados (filas y columnas).
- Facilita el manejo de gran cantidad de datos y su gestión gracias a la simplicidad del modelo relacional.

Este tipo de base de datos son deficientes para manejar datos provenientes de gráficos, multimedia y sistemas de referenciación geográficos. No permite el desarrollo de tablas organizadas de forma jerárquicas.

Las bases de datos relacionales se dividen en dos partes:

1. Datos que pueden ser numéricos, caracteres, texto, fecha, moneda entre otro.
2. Esquema, es el parámetro que establece la estructura de la base de datos con el nombre de la tabla, el tipo de dato y el registro (Cabello, 2010).



# CAPÍTULO 3

## 3. IMPLEMENTACIÓN DEL ALGORITMO PARA LA DETECCIÓN DE FALLAS

En este capítulo se describen las consideraciones para el diseño, funcionamiento y características de la solución al problema descrito. Se detalla el análisis para desarrollar el algoritmo para la detección temprana de fallas en equipos inalámbricos en una empresa proveedora de Internet.

### 3.1 Adquisición de datos

#### 3.1.1 Identificación del escenario de estudio

Para la implementación del algoritmo que permite la detección temprana de fallas en equipos inalámbricos es necesario contar con un conjunto de datos que indican los principales parámetros de funcionamiento de los equipos. Estos datos fueron adquiridos de los diferentes CPE para su posterior análisis.

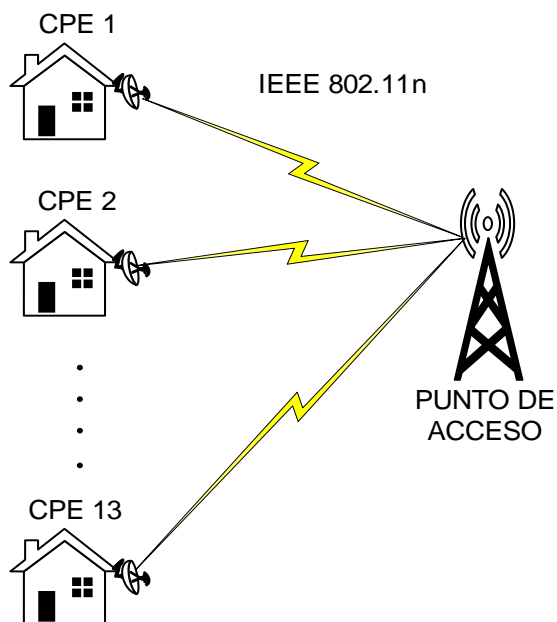


Figura 3.1 Escenario de estudio

El escenario de estudio que se utilizó para la adquisición de datos se puede observar en la Figura 3.1. El mismo consta de una red Punto a Multipunto, formada por 13 equipos CPE de marca Mikrotik los cuales se conectan a un Punto de Acceso de forma inalámbrica utilizando el estándar IEEE 802.11n. Las principales características de los equipos utilizados en el escenario de estudio se listan en la Tabla 3.1.

**Tabla 3.1 Características de los CPE y Punto de Acceso**

	CPE	Punto de Acceso
Modelo	RBLHG-5nD	RBSXTG-5HPacD
Marca	Mikrotik	
CPU	AR9344	QCA9557
Números CPU	1	1
Frecuencia del CPU	600 MHz	720 MHz
Licencia RouterOS	3	4
RAM	64 MB	128 MB
Temperatura de operación	-40°C - 70°C	-40°C - 70°C
Data Rate	300 Mbit/s	867 Mbit/s
Número de canales	2	2
Estándar	802.11a/n	802.11a/n/ac
Ganancia de la antena (dBi)	24.5	16
Potencia de transmisión (dBm)	25	31
Sensibilidad de recepción (dBm)	-96	-96

Todos los equipos CPE del escenario de estudio fueron configurados con los siguientes parámetros:

- Operación en modo puente.
- Una dirección IP privada.
- Nombre de identificación.
- Se trabaja con un Esquema de Codificación y Modulación (MCS, por sus siglas en inglés) máxima de 11 con doble polarización.

**Tabla 3.2 Ubicación de los CPE y el Punto de Acceso**

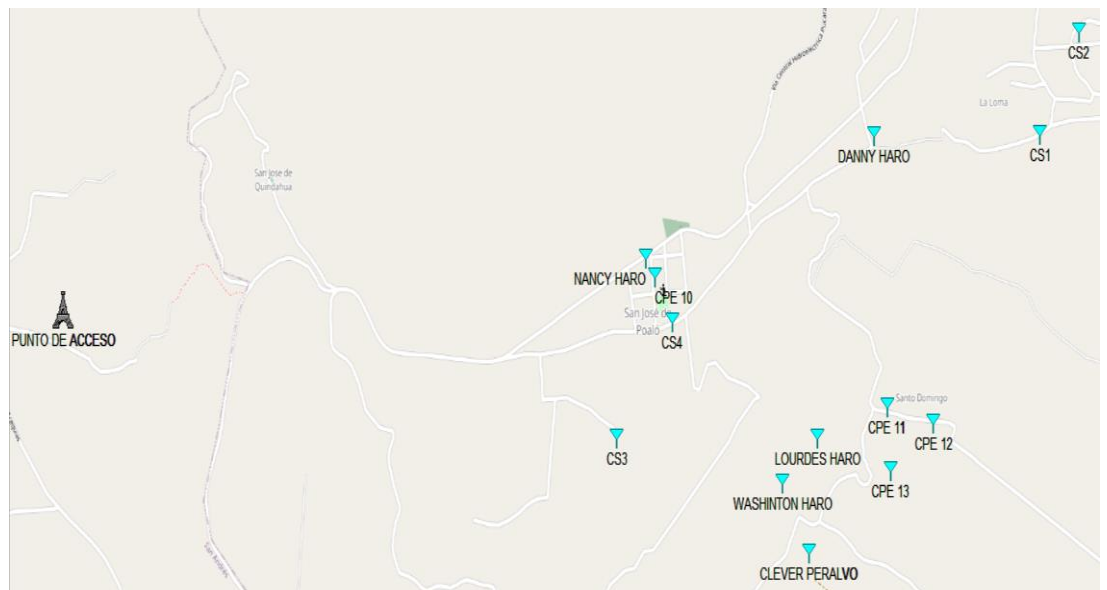
CPE	NOMBRE	IP	Coordenada	Distancia AP-CPE (Km)	Altura Antena (m)
1	CLEVER PERALVO	192.172.100.230	-1.10798, -78.47394	3,08	5,5
2	WASHINTON HARO	192.172.100.231	-1.106298, -78.4749	2,94	6,0
3	DANNY HARO	192.172.100.232	-1.097854, -78.47157	3,31	2,5
4	NANCY HARO	192.172.100.233	-1.100843, -78.47986	2,36	6,5
5	LOURDES HARO	192.172.100.234	-1.105199, -78.47364	3,07	3,0
6	CS1	192.172.100.235	-1.097821, -78.46553	3,98	4,0
7	CS2	192.172.100.236	-1.095355, -78.4641	4,17	7,0
8	CS3	192.172.100.237	-1.105194, -78.48093	2,26	4,5
9	CS4	192.172.100.238	-1.102388, -78.4789	2,46	7,5
10	CPE 10	192.172.100.239	-1.10127, -78.47955	2,39	8,0
11	CPE 11	192.172.100.241	-1.104437, -78.47107	3,34	3,0
12	CPE 12	192.172.100.242	-1.104827, -78.46941	3,53	3,0
13	CPE 13	192.172.100.243	-1.105998, -78.47095	3,37	3,0
	AP ADQUISICIÓN	192.172.100.100	-1.10206, -78.50107		12

El AP de la red punto multipunto indicado en la Figura 3.1, se encarga de brindar el acceso de forma inalámbrica a todos los equipos CPE, para el escenario de estudio se contó con un AP configurado con los siguientes parámetros:

- Operación en modo AP BRIGE.
- Una dirección IP privada en el mismo rango de los CPE.
- Nombre de identificación.
- Modulación máxima de los CPE de MCS:11, con doble polarización.
- Utiliza protocolo de comunicación propietario de Mikrotik.
- Ancho de canal de 20 MHz.
- Frecuencia de operación 5.8 GHz.

Los datos para el análisis e implementación del algoritmo de detección de fallas se obtuvieron de los equipos CPE. Cada uno de los CPE fueron ubicados en diferentes sitios, distancias y alturas. Esto permitió dar diversidad en el proceso de adquisición de datos. En la Tabla 3.2 se lista las coordenadas donde se ubicaron cada uno de los CPE además se observa la distancia AP-CPE.

En la Figura 3.2 se puede observar cómo fueron distribuidos geográficamente los equipos CPE acorde con las coordenadas listadas en la Tabla 3.2. Para el escenario de estudio, los CPE se conectaron hacia el AP para formar la red punto multipunto.

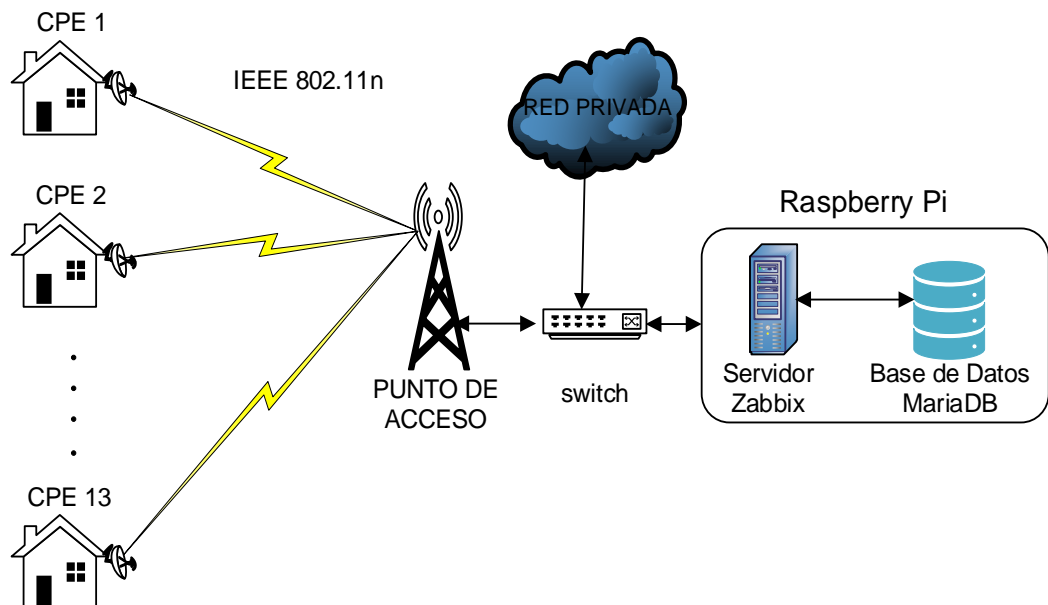


**Figura 3.2 Distribución de CPE**

### **3.1.2 Implementación de sistema de adquisición de datos**

Una vez instalados los equipos en campo, se continuó con la implementación del sistema para la adquisición de datos históricos de los equipos inalámbricos del ISP. Se hace uso de la herramienta Zabbix para adquirir los datos de los CPE.

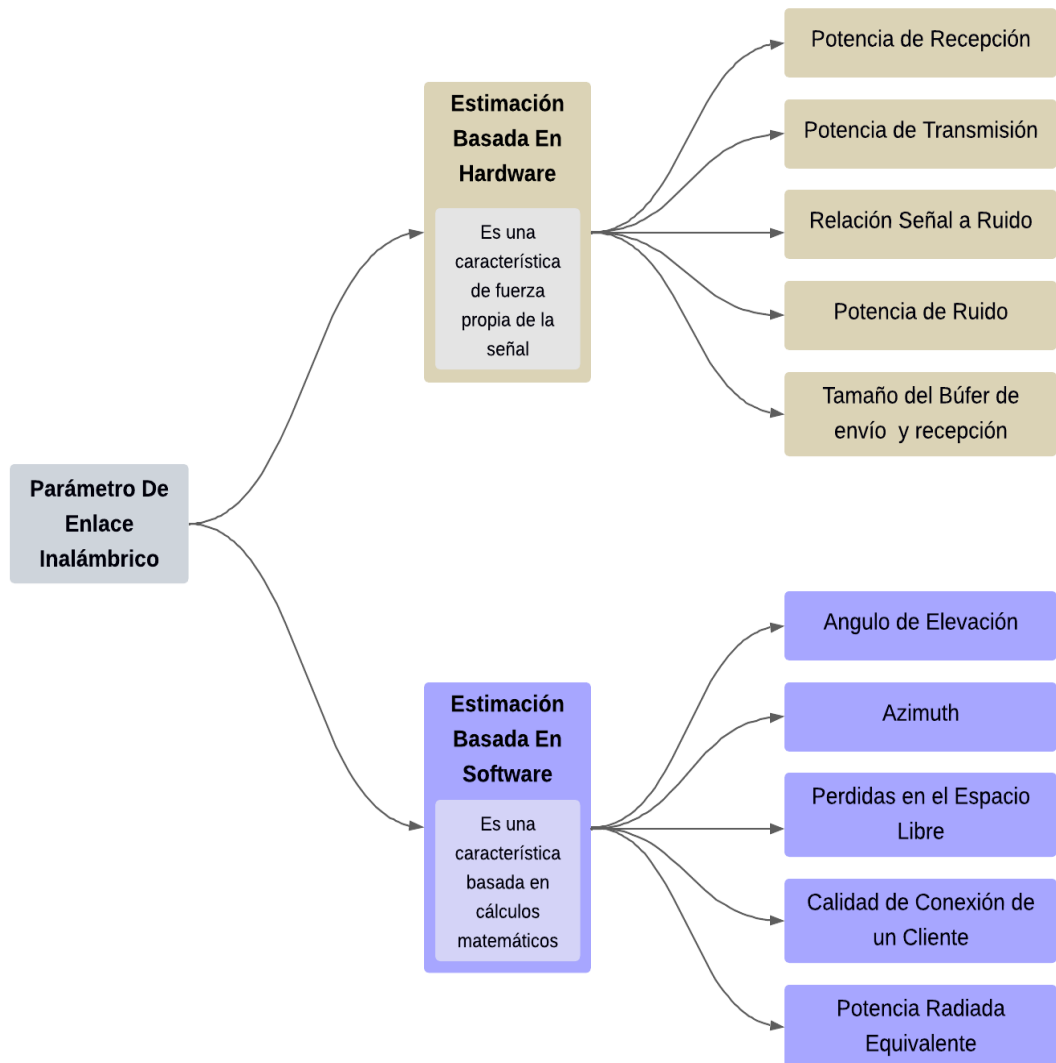
El servidor Zabbix está instalado en un Raspberry Pi que es un ordenador de bajo costo y compacto, basado en hardware libre y utiliza un sistema operativo GNU/Linux. Para este microordenador se desarrolló un sistema operativo conocido como Raspbian basado en Debian. La instalación de Zabbix en Raspberry Pi, permite monitorear cientos de dispositivos recopilando métricas en tiempos que van desde 1 segundo en adelante. En la Figura 3.3 se observa los principales componentes del sistema de adquisición de datos de los equipos CPE.



**Figura 3.3 Sistema de adquisición de datos**

La instalación del servidor Zabbix, la base de datos MariaDB y los parámetros de configuración del protocolo SNMP se describen en el manual de implantación (Anexo 1). El sistema de monitoreo facilita el acceso a la información de varios parámetros de los equipos mediante la utilización de MIB proporcionado por el fabricante y dicha información es almacenada en la base datos para su posterior análisis y tratamiento.

La Figura 3.4 indica los principales parámetros que permiten estimar la calidad de un enlace inalámbrico se dividen en estimaciones basadas en hardware y software. (Wang et al., 2007)



**Figura 3.4 Parámetro de enlace inalámbrico**

Los sistemas de monitoreo facilitan el acceso a la información de parámetros de enlaces inalámbricos con estimación basada en hardware, por tal razón dentro del presente trabajo se monitoreará las variables listadas en la Tabla 3.3. Cada una de estas variables se obtienen de los CPE que forman el escenario de estudio.

**Tabla 3.3 Variable monitoreadas**

Variable	Descripción
PRxStrength	Potencia de recepción promedio
PRxStrengthCh0	Potencia de recepción en el canal horizontal
PRxStrengthCh1	Potencia de recepción en el canal vertical
PTxStrength	Potencia de transmisión promedio
PTxStrengthCh0	Potencia de transmisión en el canal horizontal
PTxStrengthCh1	Potencia de transmisión en el canal vertical
SNR	Relación Señal a Ruido
NoiseFloor	Potencia de Ruido

La información de las variables fue accedida por el servidor Zabbix utilizando el protocolo de comunicación SNMP. Toda la información de las variables se registra en la base datos cada minuto aproximadamente.

### 3.1.3 Recreación de los distintos escenarios de funcionamiento del sistema

Dentro de una red inalámbrica punto multipunto se puede tener varios escenarios de funcionamiento que indican distintas características del enlace inalámbrico entre los CPE y el AP. Para el presente trabajo se consideran tres escenarios de funcionamiento:

- 1) **Correcto.** Los equipos CPE y el AP estarán operando en condiciones normales, brindando un servicio de calidad a los clientes.



**Figura 3.5 Alineación correcta**

En la Figura 3.5 los equipos CPE y el Punto de Acceso se encuentran alineados de forma óptima, de tal manera que sus lóbulos principales están uno frente al otro.

- 2) **Advertencia.** Los equipos están funcionando de forma correcta, pero presentan degradación de los parámetros del enlace inalámbrico, lo que indica que se debe tomar medidas para evitar que los equipos CPE entren en falla.
- 3) **Fallo.** Los CPE están operando bajo condiciones que afecta la calidad del servicio brindado a los clientes, en este estado se tiene cortes del servicio e intermitencia esto por la degradación de los parámetros del enlace inalámbrico entre los equipos CPE y el AP.

Para el presente trabajo y la recreación de los escenarios de funcionamiento Fallo y Advertencia, se toma como referencia los errores de alineación entre los equipos CPE y el AP. A continuación, se describen los errores de alineación:

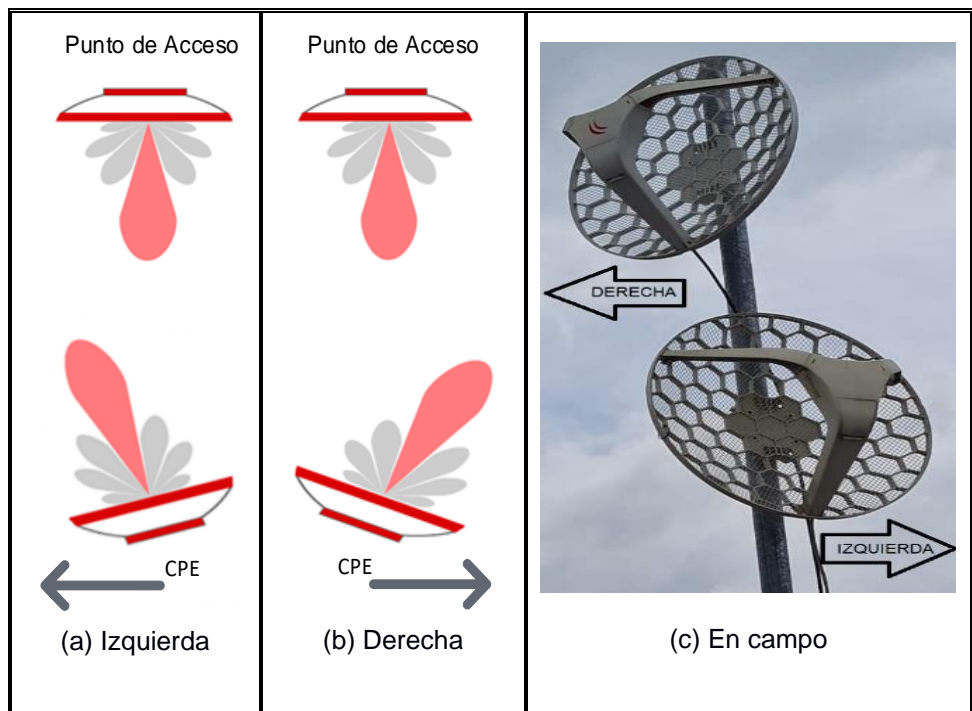
➤ **Error de alineación horizontal.**

Este tipo de error se produce cuando el lóbulo principal del equipo CPE no está correctamente alineado con el lóbulo principal del punto de acceso. Este error puede producirse por movimiento mecánicos o falla al momento de la instalación del equipo CPE.

En la Figura 3.6 el lóbulo principal del equipo CPE no está alineado correctamente con el lóbulo principal del AP. Esto causa un error de alineación en sentido horizontal hacia la Derecha (Figura 3.6-b) o un error de alineación en sentido horizontal hacia la Izquierda (Figura 3.6-a).



Los equipos CPE instalados en campo campos (Figura 3.6-c) tienen una desalineación hacia la derecha e izquierda para la recreación de los escenarios de estudio Fallo y Advertencia.

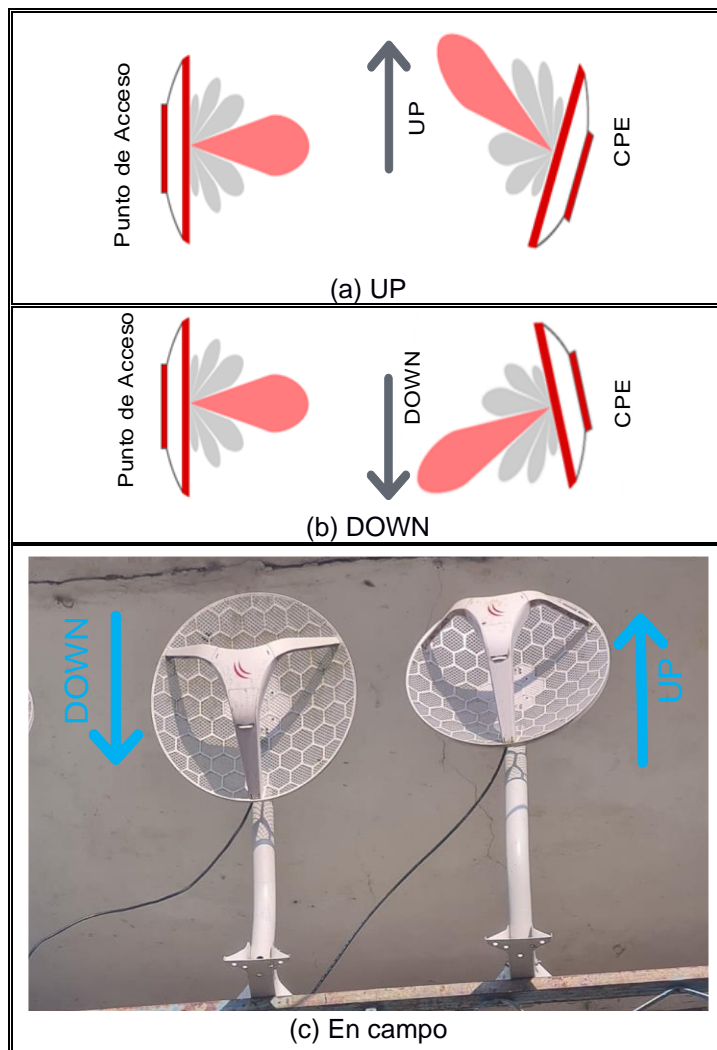


**Figura 3.6 Error de alineación horizontal**

➤ **Error de alineación vertical.**

Este error de alineación se provoca cuando el ángulo de elevación del equipo CPE es mayor o menor al ángulo de elevación calculado para tener una conexión óptima con el AP.

Cuando el ángulo de elevación es mayor al calculado se lo llama error de UP (Figura 3.7-a) y cuando es menor se lo llama error de DOWN (Figura 3.7-b). En la Figura 3.7-c se puede observar cómo fueron instalados los equipos CPE en campo para recrear los escenarios de UP y DOW.



**Figura 3.7 Error de alineación vertical**

Para el escenario de estudio se considera que el ISP ofrece en la actualidad los planes cuya velocidad y nivel de compartición se listan en la Tabla 3.4. Estas velocidades están limitadas en el AP y serán las máximas alcanzadas por los CPE.

**Tabla 3.4 Planes de Internet**

Nombre del Plan	Velocidad (Mbps)	Compartición
Básico	15	2:1
Mejorado	20	2:1
Premium	25	2:1

Para brindar un servicio de calidad y cumplir con las velocidades ofrecidas a los clientes se debe garantizar unos niveles mínimos de potencia en los equipos CPE y lograr un Esquema de Modulación y Codificación (MCS, por su sigla en inglés) adecuado (Garcia, 2015).

En la Tabla 3.5 se indican los niveles de potencia mínimos para alcanzar cierto MCS en los equipos CPE, tipo de modulación y las velocidades máximas que se pueden alcanzar utilizando un ancho de canal de 20MHz.

**Tabla 3.5 Esquema de Codificación y Modulación**

Índice MCS	# Canales	Modulación	IEEE 802.11		RSSI (dBm)
			20MHz		
			0.8µs GI	0.4µs GI	
0	1	BPSK	6.5	7.2	-88
1	1	QPSK	13	14.4	-82
2	1	QPSK	19.5	21.7	-79
3	1	16-QAM	26	28.9	-76
4	1	16-QAM	39	43.3	-73
5	1	64-QAM	52	57.8	-68
6	1	64-QAM	58.5	65	-65
7	1	64-QAM	65	72.2	-63
8	2	BPSK	13	14.4	-85
9	2	QPSK	26	28.9	-79
10	2	QPSK	39	43.3	-76
11	2	16-QAM	52	57.8	-73
12	2	16-QAM	78	86.7	-70
13	2	64-QAM	104	115.6	-65
14	2	64-QAM	117	130	-62
15	2	64-QAM	130	144.4	-60

**Nota.** Valores tomados de la presentación de Mario Clep denominado "Redes Neuronales Artificiales aplicadas a la Predicción del Precio del Oro" (Clep, 2015, p. 14)

De acuerdo con los planes ofrecidos como se listan en la Tabla 3.4 y tomando en cuenta que la máxima velocidad es de 25 Mbps, es necesario contar con un MCS:10 y una potencia mínima de -76 dBm para garantizar dichas velocidades. Considerando que mientras más alta sea la potencia

recibida en el CPE mayor será el nivel del MCS alcanzado. En la Tabla 3.6 se indican los rangos de potencia recibida por los CPE que son considerados en cada uno de los escenarios de funcionamiento.

**Tabla 3.6 Rangos de potencia de escenarios de funcionamiento**

	Potencia de Recepción (dBm)	
	min	max
Correcto	-65	-40
Advertencia	-66	-76
Fallo	-77	-88

Los tres escenarios de funcionamiento se los recreó tomando como base los errores de alineación Horizontal (Derecha e Izquierda) y Vertical (UP y DOWN), el periodo de muestreo fue de 10 días por cada escenario. Se consideró que todos los CPE iniciaron operando en el escenario Correcto y luego de su periodo de muestreo pasa al escenario de Advertencia y finalmente a Fallo.

**Tabla 3.7 Escenarios de funcionamiento**

# CPE	Escenarios de funcionamiento de los CPE	
	Abreviatura	Descripción
1 al 13	Correcto	Correcto
1-5-11-13	AHD	Advertencia por desalineación hacia la derecha
	FHD	Fallo por desalineación hacia la derecha
2-9-12	AHI	Advertencia por desalineación hacia la izquierda
	FHI	Fallo por desalineación hacia la izquierda
3-6-7	AUP	Advertencia por desalineación hacia arriba
	FUP	Fallo por desalineación hacia arriba
4-8-10	ADW	Advertencia por desalineación hacia abajo
	FDW	Fallo por desalineación hacia abajo

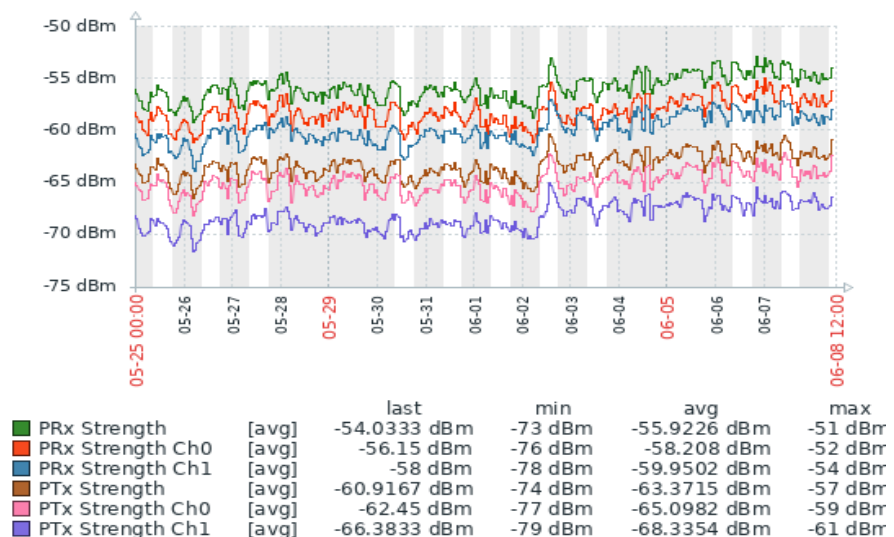
Para cada escenario se degradó la alineación de los CPE con respecto al AP mediante manipulación manual y así conseguir que los niveles de señal recibidos por el CPE desde el AP disminuyan.

Como se indicó el escenario de estudio está formado por 13 CPE monitoreados continuamente por el servidor Zabbix mediante el protocolo SNMP. Los datos obtenidos de los CPE fueron almacenados en la base MariaDB, con una frecuencia de toma de datos de un minuto. Los diferentes escenarios de funcionamientos de los CPE se listan en la Tabla 3.7.

### Recreación del escenario de funcionamiento Correcto

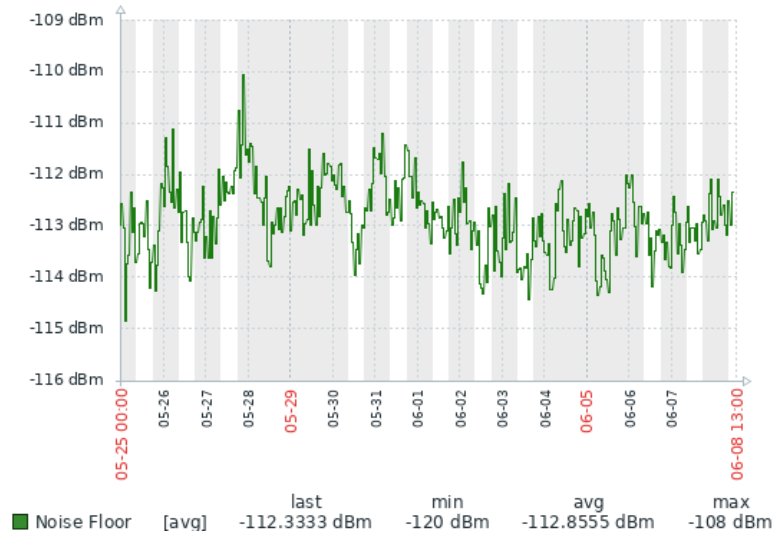
Para la recreación del escenario de funcionamiento Correcto se tomó como base los rangos potencias de recepción listados en la Tabla 3.6, estos valores indican que los equipos CPE se encuentran instalados de forma correcta y no existe ningún error de alineación.

Los niveles de potencia recibidos desde los CPE y almacenados en la base datos se encuentran en el rango -40 a -65 dBm, como se puede observar en la parte inferior de la Figura 3.8. También se indica las potencias de recepción y transmisión de los canales horizontal y vertical de los diferentes CPE con sus máximos y mínimos registrados.

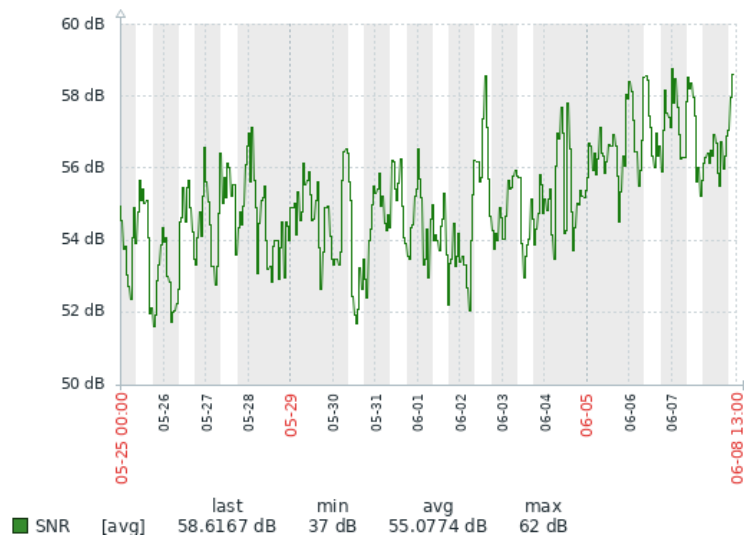


**Figura 3.8 Niveles de Potencia en funcionamiento Correcto**

Otras variables que fueron monitoreadas son la Potencia de Ruido (Figura 3.9) y la Relación Señal a Ruido (Figura 3.10) al estar en funcionamiento Correcto estas variables tiene un comportamiento relativamente estable.



**Figura 3.9 Potencia de Ruido en funcionamiento Correcto**

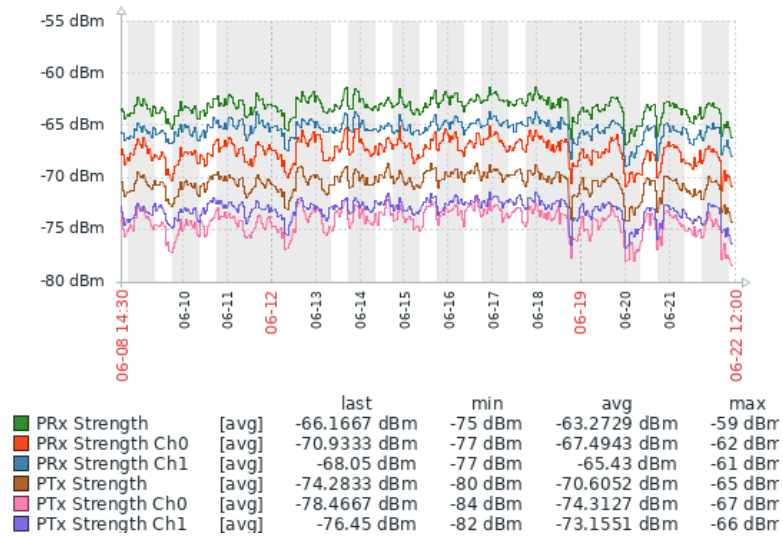


**Figura 3.10 SNR en funcionamiento Correcto**

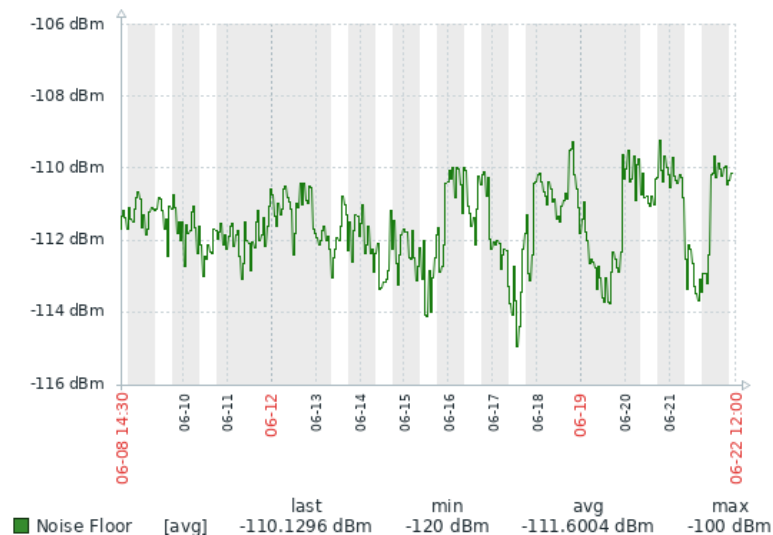
### Recreación del escenario de funcionamiento Advertencia

En este escenario de funcionamiento se realizó manipulación manual para causar un error de alineación horizontal (Derecha e Izquierda) y vertical (UP y DOWN) a los diferentes CPE según la distribución listada en la Tabla 3.7.

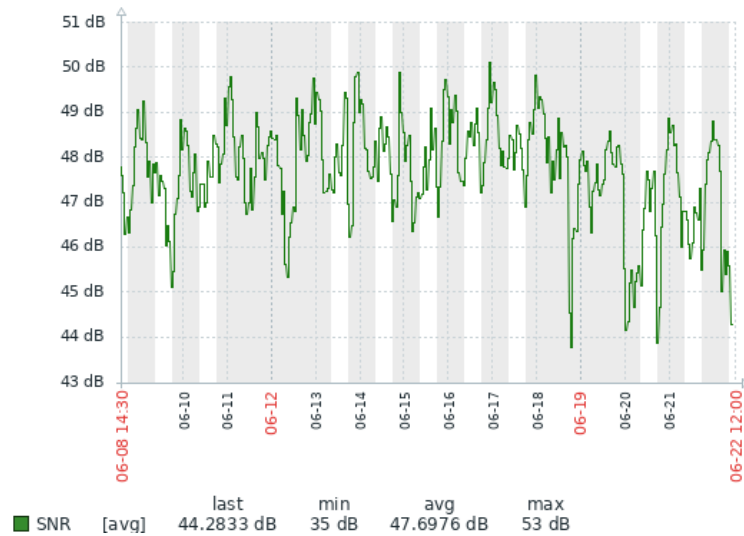
Al generar estos errores en la alineación de los CPE respecto al AP se consiguió que los niveles de la potencia de recepción disminuyan, hasta el rango de -66 a -76 dBm (Figura 3.11) en todos los CPE. La variable Potencia de Ruido (Figura 3.12) tiene una variación entre -109 a -115 dBm, la Relación Señal a Ruido (Figura 3.13) se observa que tienen una mayor variación entre 43 a 51 dB.



**Figura 3.11 Niveles de Potencia en funcionamiento Advertencia**



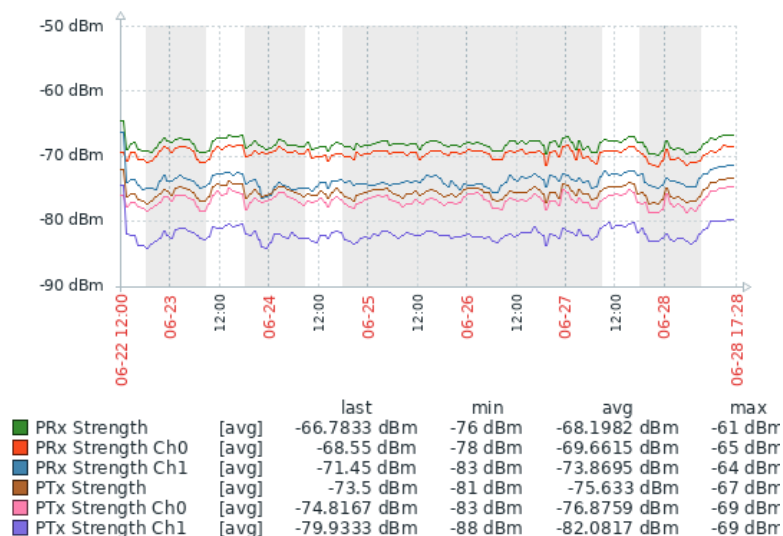
**Figura 3.12 Potencia de Ruido en funcionamiento Correcto**



**Figura 3.13 SNR en funcionamiento Advertencia**

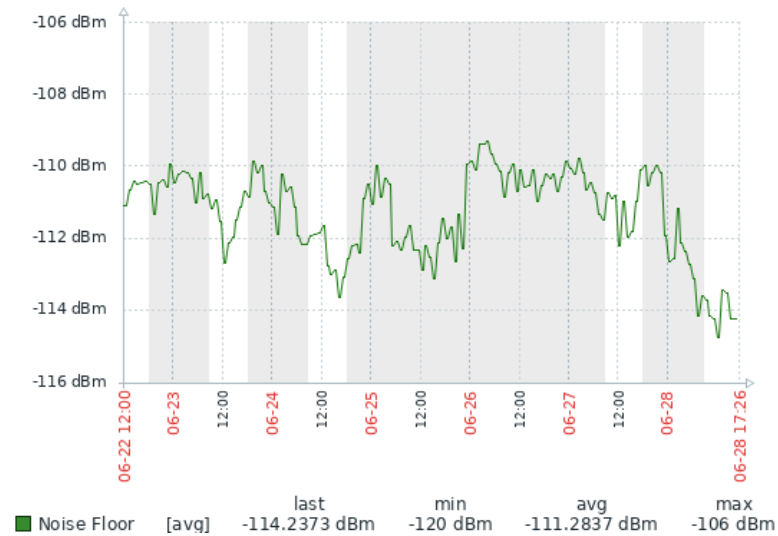
### Recreación del escenario de funcionamiento Fallo

El ultimo escenario recreado fue el de Fallo, de igual manera se realizó manipulación manual de los CPE para que los mismos operen en este escenario. Sus niveles de potencia recibidos disminuyen hasta el rango de -77 a -88 dBm (Figura 3.14), la variable de Potencia de Ruido (Figura 3.15) existe una variación en el rango de -109 a -115 dBm, la Relación Señal a Ruido (Figura 3.16) se observa una menor variación entre 46 a 41 dB.

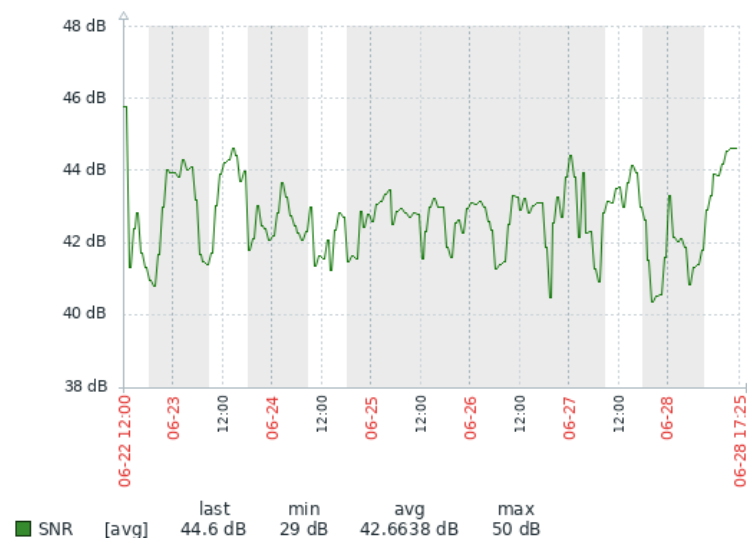


**Figura 3.14 Niveles de Potencia en funcionamiento Fallo**





**Figura 3.15 Potencia de Ruido en funcionamiento Fallo**



**Figura 3.16 SNR en funcionamiento Fallo**

### 3.2 Procesamiento de datos

Para continuar con el desarrollo del trabajo, fue necesario comprender el dataset que se ha utilizado. La base de datos se la obtuvo mediante la recreación de los diferentes escenarios de funcionamiento de los equipos CPE.

El tiempo que los equipos CPE operaron en cada uno de los escenarios (Correcto, Advertencia y Fallo), fue de 10 días por cada uno, teniendo un total de 30 días de

adquisición, estos datos se obtienen de cada uno de los CPE cada minuto y almacenados en la base datos.

### 3.2.1 Exploración de datos

El primer es importar la base de datos y revisar las variables que contiene de cada uno de los CPE que se están monitoreando.

**Tabla 3.8 Descripción del dataset**

DATASET	
Número de CPE	13
Número de columnas por CPE	12
Número de variables por CPE	8
Número de muestras por variable	42770
Condiciones de operación	9
Modo de fallo	4

Como se indica en la Tabla 3.8 el conjunto de datos está formado por 13 equipos CPE del mismo modelo. El número de muestras por CPE es de 42770 datos recolectados durante los 30 días de recreación de los diferentes escenarios de funcionamiento.

**Tabla 3.9 Dataset de cada CPE**

CPE	Muestra	clock	Noise Floor	PRx Strength	PRx Strength Ch0	PRx Strength Ch1	PTx Strength	PTx Strength Ch0	PTx Strength Ch1	SNR	Estado
1	1	1658257214	-112	-54	-56	-58	-63	-64	-67	58	Correcto
2	2	1658257274	-112	-56	-58	-59	-62	-64	-68	56	Correcto
4	3	1658257334	-112	-54	-57	-58	-63	-64	-68	58	Correcto
7	4	1658257394	-112	-54	-56	-60	-62	-64	-68	58	Correcto
11	5	1658257454	-112	-55	-58	-60	-63	-64	-69	57	Correcto
13	6	1658257514	-112	-55	-57	-59	-63	-64	-69	57	Correcto

Para tener diversidad en la toma de datos cada uno de los CPE fueron ubicados en diferentes puntos geográficas, altura y distancias como se detalla en la Tabla 3.2. De las 12 columnas de cada fichero, la columna 1 indica el número de CPE, la columna 2 el número de muestra al que corresponde los datos que se observan en cada fila, la columna 3 la hora en la que se tomó la muestra, de la columna 4 a la 11 los datos de las variables monitoreadas y la columna 12 el estado o etiqueta de las condiciones de operación. En la Tabla 3.9 se presenta cómo está formado el Dataset de cada uno de los CPE.

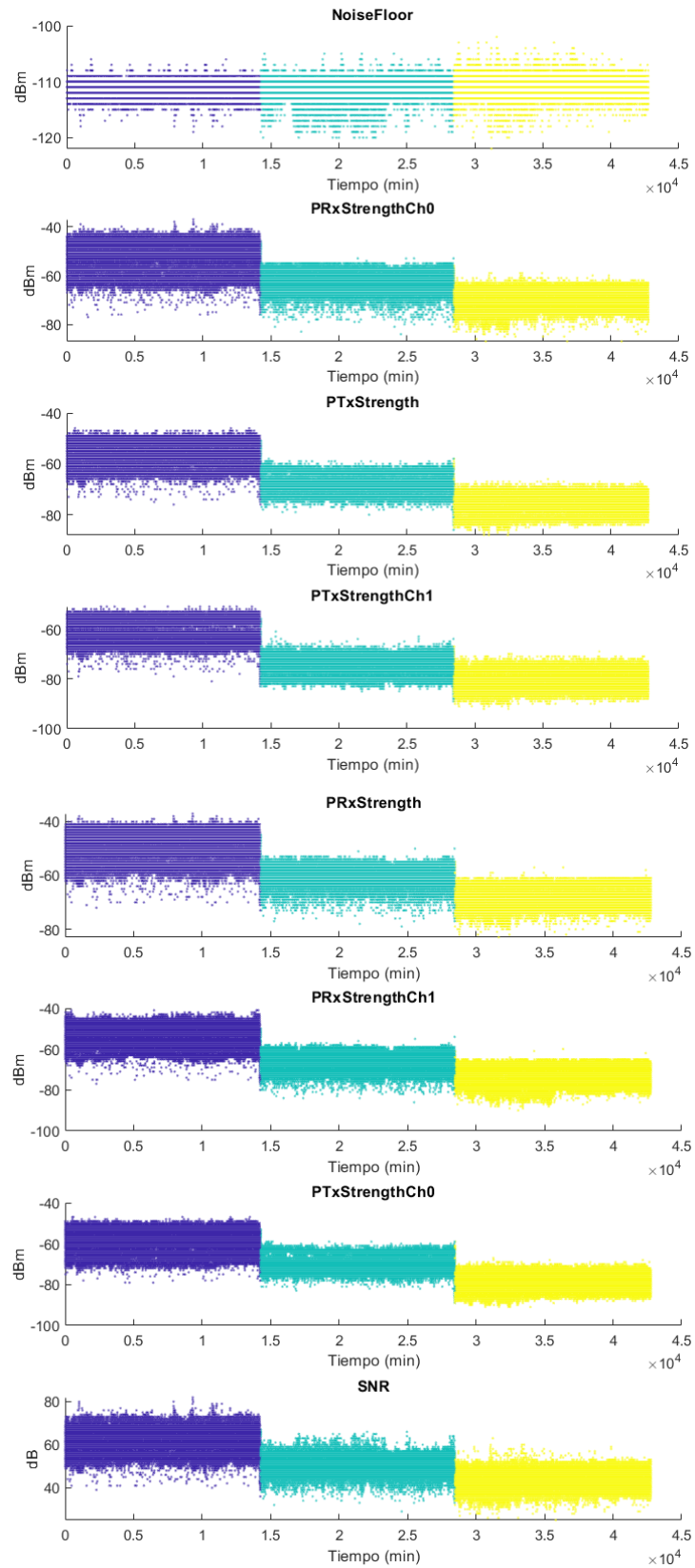
La distribución de frecuencias de todo el dataset de acuerdo cada uno de los escenarios de funcionamiento se indica en la Tabla 3.10, observando que se encuentran equilibrados.

**Tabla 3.10 Distribución de frecuencias de todo el dataset**

Estado	Muestras	Porcentaje
Correcto	185315	33.33%
Advertencia	184600	33.20%
Fallo	186082	33.47%

### 3.2.2 Preprocesar datos

Para iniciar se procedió a cargar la base de datos disponible en archivos “.csv” que contienen toda la información de las variables monitoreadas, para esto se desarrolló una función en MATLAB. Se realizó un análisis visual de todo el dataset para verificar si existe valores atípicos o valores faltantes en las diferentes muestras. Se verifica que la data de cada CPE contenga el mismo número de muestras, en este caso de estudio todos los CPE tienen un total de 42770 muestras por variable.

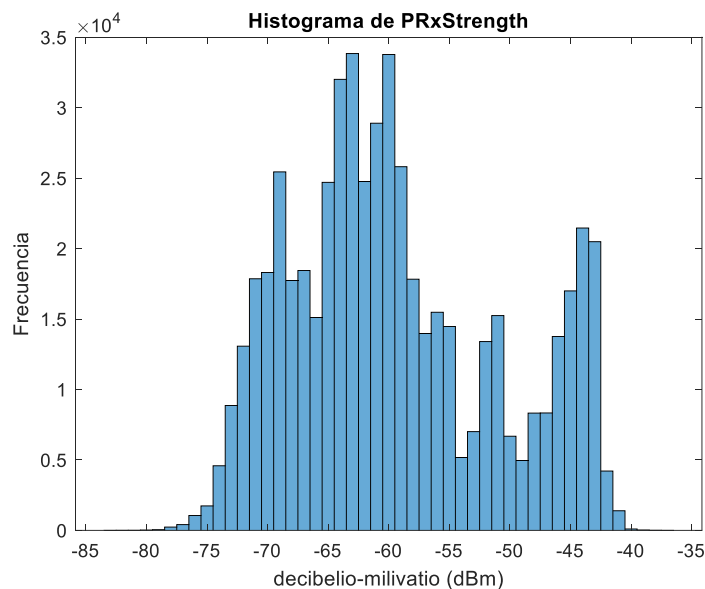


**Figura 3.17 Comportamiento de las variables**

### 3.2.3 Identificación de indicadores

Para continuar con el procesamiento de la información es necesario comprender cómo las variables se comportan a lo largo del tiempo de toma de datos. En la Figura 3.17 se indican todas las variables monitoreadas y su comportamiento a lo largo del tiempo. Los datos de las diferentes variables están agrupados; de color azul los datos correspondientes a las etiquetas de operación Correcto; de color celeste los datos correspondientes a las etiquetas de operación Advertencia; y de color amarillo los datos correspondientes a las etiquetas de operación Fallo.

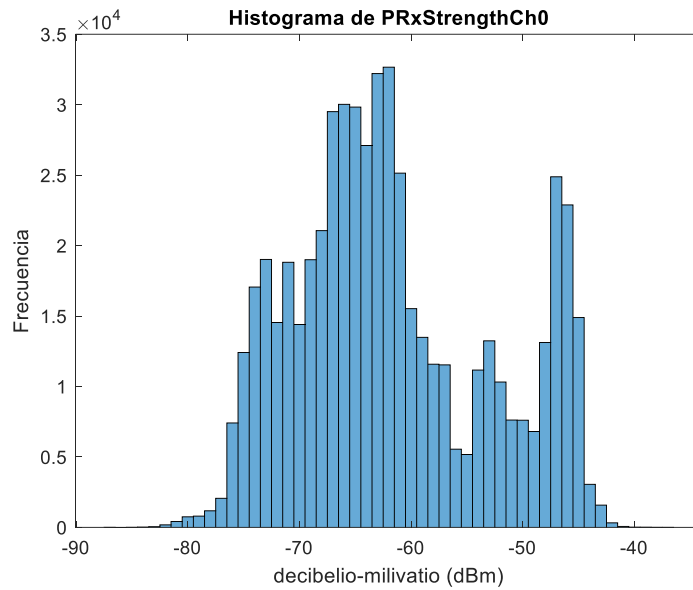
Una vez realizado el preprocesado de datos, se realizó un análisis de las principales características que fueron útiles para predecir las fallas en equipos inalámbricos. Para ver el comportamiento de las variables a lo largo del dataset se utiliza el histograma.



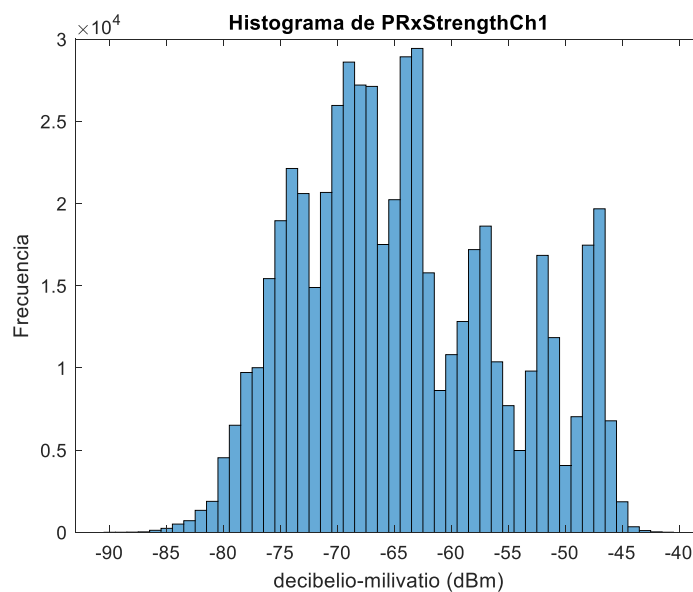
**Figura 3.18 Histograma de la variable PRxStrength**

El histograma de la Figura 3.18 muestra la distribución de datos observando que existe una concentración en los rangos de -75 a 70 dBm y de -65 a -60 dBm. Esto indica que hay una mayor frecuencia de eventos o mediciones

en esos intervalos específicos. Por otro lado, vemos una disminución en la frecuencia de datos en el rango de -60 a -55 dBm, seguida de un aumento notable en el rango de -45 a -40 dBm. Esto sugiere una tendencia o patrón particular en la distribución de los datos.



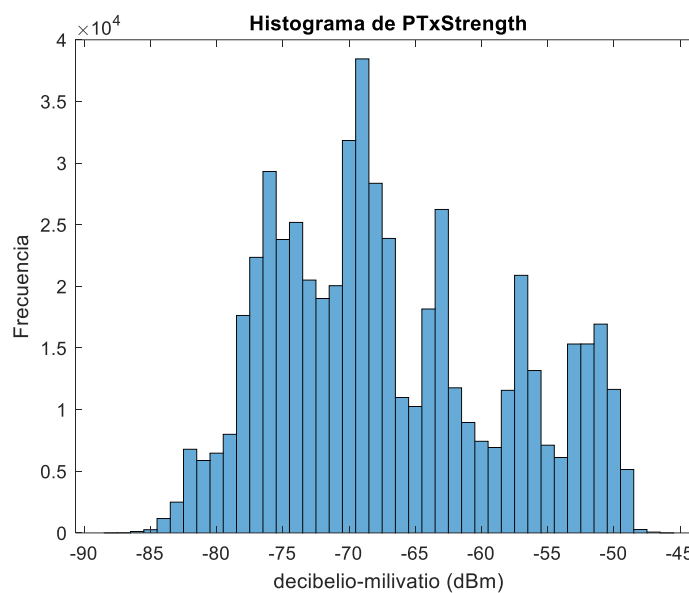
**Figura 3.19** Histograma de la variable PRxStrengthCh0



**Figura 3.20** Histograma de la variable PRxStrengthCh1

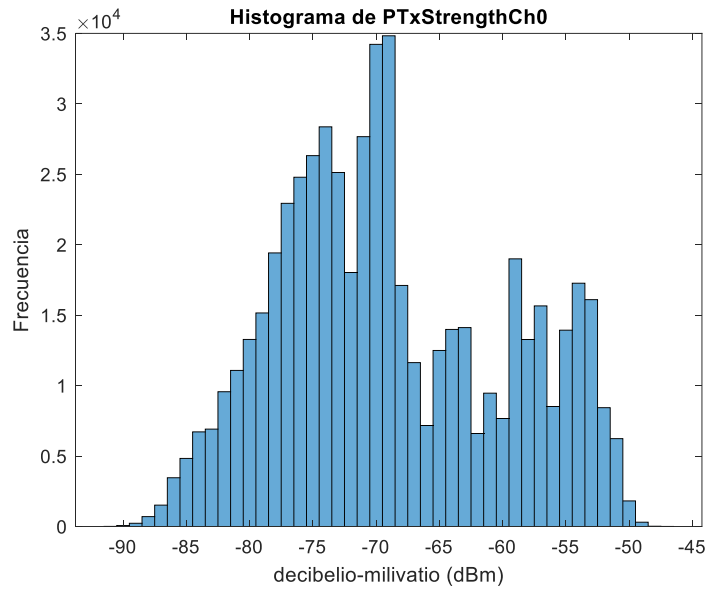
Del Histograma de la Figura 3.19 se observa un patrón en el rango de -80 a -70 dBm, donde existe un aumento progresivo en la frecuencia de los eventos medidos. Los datos que se encuentran entre -60 a -50 dBm indican una disminución gradual seguida de un nuevo incremento en el rango de -50 a -40 dBm.

Se evidencia un crecimiento constante en la frecuencia en el rango -85 a -75 dBm como se observa en el histograma de la Figura 3.20. A partir de ahí, se observa fluctuaciones en los rangos -75 a -70 y -70 a -65 dBm, pero en general se mantiene una tendencia descendente seguido de fluctuaciones en frecuencia lo cual podría ser indicativos de la existencia de diferentes grupos o categorías en los datos.

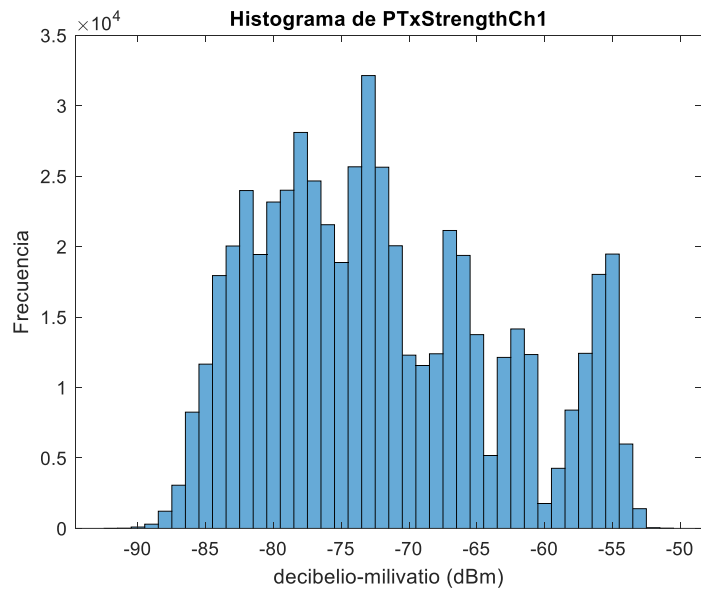


**Figura 3.21 Histograma de la variable PTxStrength**

Se observa un patrón caracterizado por un aumento gradual en su frecuencia a lo largo del rango entre -85 a -75 dBm como se indica en el histograma de la Figura 3.21 a partir de -75 dBm se evidencia un decrecimiento de frecuencia.



**Figura 3.22 Histograma de la variable PTxStrengthCh0**

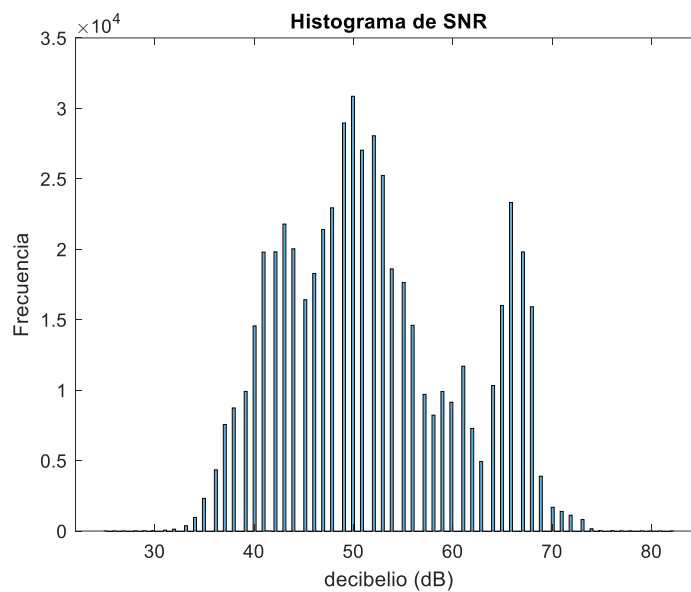


**Figura 3.23 Histograma de la variable PTxStrengthCh1**

Al analizar el histograma de la Figura 3.22 se observa un crecimiento constante en el rango de -90 a -75 dBm. A partir de ahí, se identifica un decrecimiento de -75 a -70 dBm, seguido de un aumento significativo. En el rango de -70 a -50 se observa fluctuaciones en sus frecuencias.



Se observa patrones interesantes en la distribución de datos Enel histograma de la Figura 3.23. en los rangos iniciales, evidencia un crecimiento contante de la frecuencia, seguido de un decrecimiento y fluctuaciones en los valores registrados. Sin embargo, destaca el rango de -75 a 70 dBm, donde se alcanza el valor máximo de frecuencia, este resultado siguiere una mayor concentración de datos en esa región específica.



**Figura 3.24 Histograma de la variable SNR**

Se evidencia en el histograma de la Figura 3.24 que, en el rango de 30 a 40 dB, un crecimiento constante de valores y a medida que se avanza en el rango de 40 a 50 dB, produce una ligera variación con incrementos y decrementos. En general e histograma muestra un patrón variable de frecuencia.

Los resultados del análisis de medias y desviaciones realizado al dataset se indican en la Tabla 3.11. La mayoría de las variables tienen una desviación estándar similar. Los máximos y mínimos en las variables de potencia están de acorde a los niveles de potencia utilizados para la recreación de cada escenario de estudio. Estos valores similares se deben

a que todos los CPE operaron los escenarios Correcto, Advertencia y Fallo. La variable NoiseFloor tiene una desviación pequeña en comparación con el resto de las variables esto indica que el sitio donde están ubicados los CPE no se tiene señales que introduzcan ruido en el enlace.

**Tabla 3.11 Descripción de las variables**

Variable	Muestras	Mínimo	Máximo	Media	Moda	Desviación Estándar
PRxStrength	42770	-83	-37	-59,38	-63	8,73
PRxStrengthCh0		-87	-37	-61,62	-62	8,91
PRxStrengthCh1		-90	-41	-64,17	-63	9,08
PTxStrength		-88	-46	-66,88	-69	8,71
PTxStrengthCh0		-91	-51	-68,84	-69	8,98
PTxStrengthCh1		-92	-51	-72,09	-73	8,82
SNR		25	82	51,96	50	8,95
NoiseFloor		-122	-102	-111,33	-111	1,11

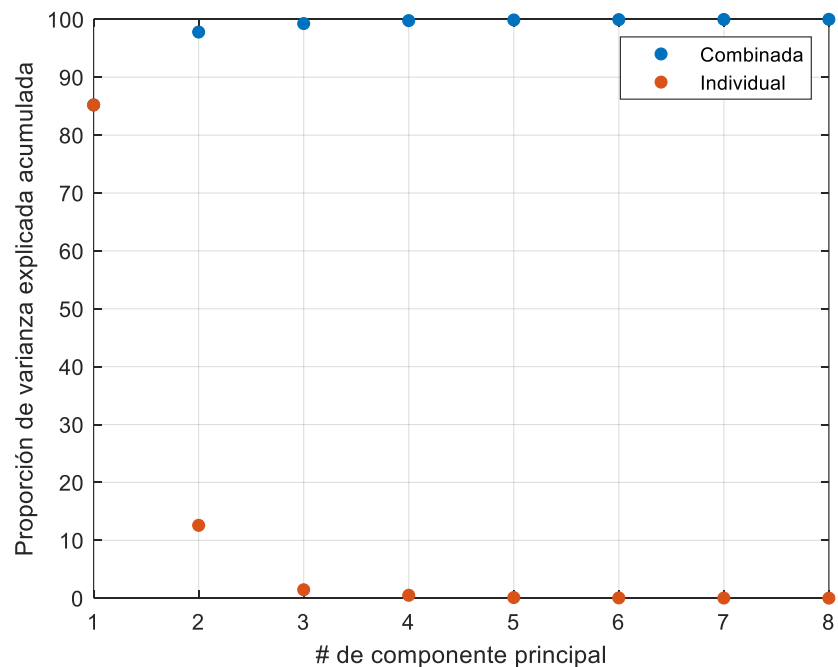
### 3.2.4 Análisis de componentes principales

El análisis de componentes principales (PCA, por sus siglas en inglés), es un método estadístico utilizado para el análisis de datos que contengan múltiples variables cuantitativas. Su utilidad principal es reducir las dimensiones de la base de datos, esta técnica permite elegir el menor número de predictores para pronosticar una variable. (Bedoya-Marrugo et al., 2017, p. 2)

Una vez asegurado que el dataset estudiado en la Figura 3.17, y la revisión de los principales indicadores estadísticos, se procede con el PCA para lo cual se realiza el siguiente procesamiento de datos.

1. Estandarización o normalizar los datos, se realiza debido a que cada una de las variables tiene diferentes unidades y rangos de variación. Consiste en tomar toda la información de las variables y escalarlas para que tengan una media de cero y una desviación típica de uno.

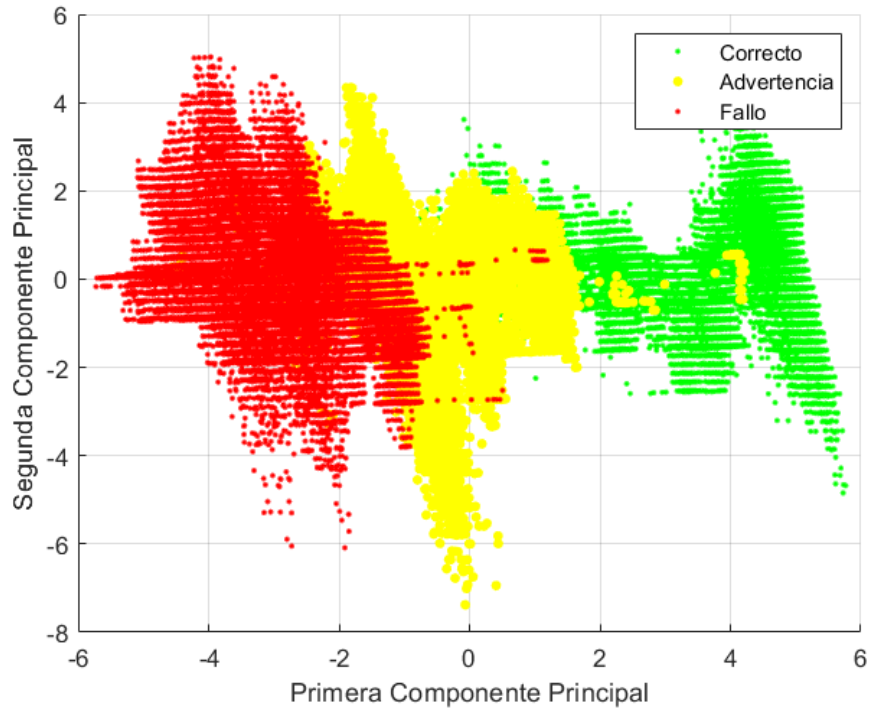
2. Calcular las componentes principales del dataset con las variables normalizadas para esto se, desarrolló una función en MATLAB, en la parte superior de la Figura 3.25, se observar la variabilidad explicada con la combinación de cada componente principal y en la parte inferior la variabilidad explicada con cada componente principal de forma individual. Con la primera componente principal se explica el 85% de la variabilidad y con la segunda componente principal puede explicar el 12.5% de la variabilidad.



**Figura 3.25 Varianza individual y acumulada explicada por PCA**

3. El comportamiento de las dos primeras componentes principales, resultado del PCA se indica en la Figura 3.26, los tres escenarios de funcionamiento de los CPE; de color verde los datos en operación Correcto; de amarillo los datos de operación Advertencia; rojo los datos de operación Fallo.  
Se encuentra tres patrones claros correspondiente a los escenarios de funcionamiento (Correcto, Advertencia, Fallo). La mayoría de los

datos atípicos se genera en el escenario de funcionamiento Fallo, los mismo tienden a sobreponerse con el escenario de Advertencia.



**Figura 3.26 Datos de PCA en escenario Correcto, Advertencia y Fallo**

4. La matriz de correlación permite determinar el coeficiente de correlación lineal entre dos variables que pueden ajustarse muy bien a una recta o cualquier función matemática. Mientras el valor sea más cercano a uno indica que las dos variables crecen o decrecen de forma simultánea presentando una alta correlación. Si el coeficiente de correlación es cero o se aproxima a cero esto indica que no existe ninguna relación entre las variables de estudio. Si el coeficiente se acerca a menos uno indica que las variables tienen una correlación inversa (IBM, 2022).

En la Tabla 3.12 muestra la correlación entre las variables del dataset observado que entre la mayoría de las variables se tiene una correlación positiva alta con excepción de la variable NoiseFloor la

cual no tiene correlación con el resto de las variables. La correlación entre las variables de potencia PRxStrengthCh0/1 y PTxStrengthCh0/1 indica que cualquier efecto o variación en una de ellas puede llegar afectar al resto de variables esto se debe que una vez que se realiza la manipulación para que los CPE operen en los estados de funcionamiento Correcto, Advertencia y Fallo los niveles de señal se ven afectados en los dos canales.

**Tabla 3.12 Matriz de correlación**

MATRIZ DE CORRELACIÓN								
	NoiseFloor	PRxStrength	PRxStrengthCh0	PRxStrengthCh1	PTxStrength	PTxStrengthCh0	PTxStrengthCh1	SNR
NoiseFloor	1,000	-0,117	-0,123	-0,108	-0,053	-0,060	-0,057	-0,213
PRxStrength	-0,117	1,000	0,976	0,969	0,975	0,963	0,961	0,980
PRxStrengthCh0	-0,123	0,976	1,000	0,924	0,964	0,971	0,922	0,965
PRxStrengthCh1	-0,108	0,969	0,924	1,000	0,948	0,915	0,971	0,958
PTxStrength	-0,053	0,975	0,964	0,948	1,000	0,984	0,970	0,958
PTxStrengthCh0	-0,060	0,963	0,971	0,915	0,984	1,000	0,934	0,947
PTxStrengthCh1	-0,057	0,961	0,922	0,971	0,970	0,934	1,000	0,944
SNR	-0,213	0,980	0,965	0,958	0,958	0,947	0,944	1,000

# CAPÍTULO 4

## 4. ANÁLISIS DE RESULTADOS

Se presentan los resultados que se obtuvieron luego de haber realizado el entrenamiento de los diferentes algoritmos de aprendizaje autónomo en base a los datos obtenidos a través del sistema de monitoreo.

### 4.1 Evaluación y entrenamiento de algoritmos de aprendizaje autónomo

Una vez identificadas las características del dataset para el estudio, se procedió a entrenar los diferentes algoritmos de aprendizaje autónomo. Tanto para el entrenamiento como para la evaluación se utilizó todas las variables indicadas en la Tabla 3.11.

El dataset de entrenamiento fue formado por los datos que aportaron los 13 equipos CPE en cada uno de sus escenarios de funcionamiento. El dataset de evaluación de los algoritmos se formó por el 25% del total del dataset estos datos no se utilizaron en el entrenamiento. Al contar con el conjunto de datos etiquetados de acuerdo con el escenario de funcionamiento, se utilizó los modelos de aprendizaje supervisado cuyo valor a predecir se encuentra en la columna de nombre Estado, el resto de las variables del dataset son predictoras.

Para el entrenamiento de los algoritmos se utilizó la herramienta Classification Learner de MATLAB. Se realizó utilizando los principales algoritmos de aprendizaje supervisado para detección de fallas como se describe a continuación:

#### 4.1.1 Entrenamiento con los algoritmos de árbol de decisión

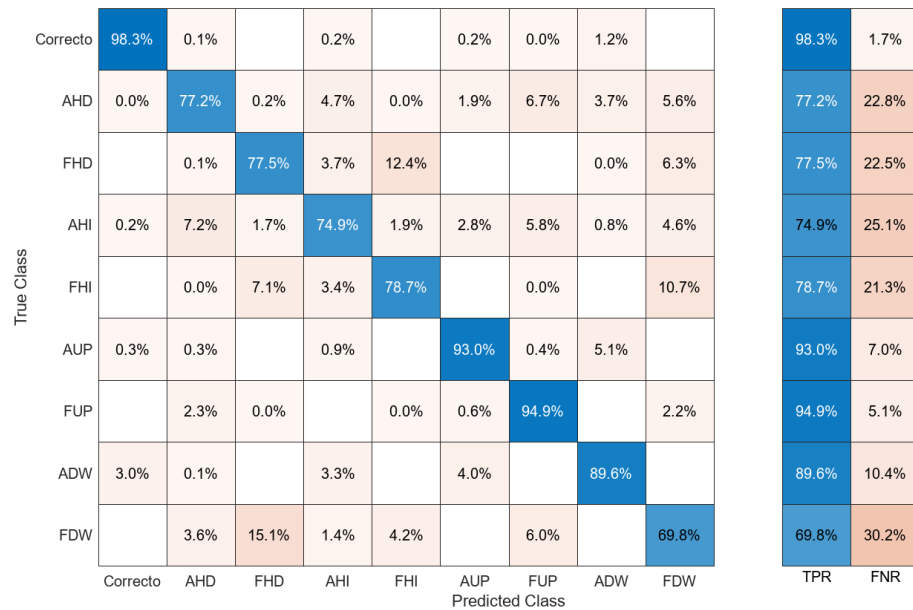
Para el entrenamiento, se tomó como referencia los tres modelos indicados en la Tabla 4.1, el parámetro que se varia en estos algoritmos es el número de ramificaciones dentro de las cuales Breiman (2017) manifiesta que se puede considera; Coarse Tree cuando su número de ramificaciones es

menor a 10; Medium Tree con número de ramificaciones se encuentran entre 11 a 25; Fine Tree un número de ramificaciones entre 26 y 100.

**Tabla 4.1 Precisión de los algoritmos de árbol de decisión**

Algoritmo	# Ramificaciones	Precisión (%)
Fine Tree	100	87,15
Medium Tree	20	78,67
Coarse Tree	4	57,98

Las matrices de confusión de las Figuras 4.1 a 4.3, obtenidas al finalizar el entrenamiento, se obtuvo una precisión global de 87,15% en el modelo Fine Tree como se aprecia en la Tabla 4.1. Mientras mayor sea el número de ramificaciones el algoritmo logra predecir con mayor precisión cada una de las categorías.



**Figura 4.1 Matriz de confusión Fine Tree**

True Class	Correcto	96.2%	0.2%		0.4%		0.1%		3.1%		96.2%	3.8%
	AHD	0.0%	77.2%	0.0%	9.9%		3.7%	4.7%	3.2%	1.3%	77.2%	22.8%
	FHD		3.7%	87.5%	0.1%	5.3%	0.0%	1.8%	0.0%	1.5%	87.5%	12.5%
	AHI	0.1%	40.0%	3.5%	51.0%		3.1%	1.3%	0.1%	0.9%	51.0%	49.0%
	FHI		4.8%	29.5%		53.7%	0.0%	0.1%		11.8%	53.7%	46.3%
	AUP	0.1%	12.3%		0.7%		81.9%	0.1%	4.8%	0.0%	81.9%	18.1%
	FUP		10.4%	0.2%	0.0%			89.3%		0.1%	89.3%	10.7%
	ADW	1.2%	1.9%		15.5%		2.3%	0.0%	79.1%	0.0%	79.1%	20.9%
	FDW		22.8%	31.0%	0.0%	7.2%		7.6%		31.4%	31.4%	68.6%
			Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR
		Predicted Class										

**Figura 4.2 Matriz de confusión Medium Tree**

True Class	Correcto	97.1%	0.5%				2.4%				97.1%	2.9%
	AHD	0.0%	90.5%	1.3%			8.1%				90.5%	9.5%
	FHD		5.7%	94.3%			0.0%				94.3%	5.7%
	AHI	0.1%	83.1%	4.3%			12.5%					100.0%
	FHI		4.9%	95.1%			0.0%					100.0%
	AUP	0.1%	13.0%	0.0%			86.9%				86.9%	13.1%
	FUP		99.7%	0.3%								100.0%
	ADW	5.3%	9.4%	0.0%			85.3%					100.0%
	FDW		30.4%	69.6%								100.0%
			Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR
		Predicted Class										

**Figura 4.3 Matriz de confusión Coarse Tree**

#### 4.1.2 Entrenamiento con los algoritmos de análisis discriminante

Para el entrenamiento de este modelo se tomó como referencia los modelos lineales y cuadráticos. Una vez finalizado el entrenamiento los resultados se observan en la matriz de confusión de la Figura 4.4 a-b. Se



obtuvo una precisión global de 75,96% para el modelo lineal y 89,50% para el modelo cuadrático.

True Class	Correcto	92.1%	1.7%		0.2%		1.6%	0.0%	4.3%		92.1%	7.9%	
	AHD		64.5%	1.2%	16.4%	0.1%	7.6%	4.5%	1.2%	4.6%	64.5%	35.5%	
	FHD		1.6%	83.0%	0.5%	6.2%		3.8%		4.8%	83.0%	17.0%	
	AHI		10.0%	23.9%	44.7%	0.4%	18.1%	2.5%	0.1%	0.4%	44.7%	55.3%	
	FHI		0.1%	19.3%	0.8%	64.5%		0.8%		14.5%	64.5%	35.5%	
	AUP		0.0%	5.1%		0.3%	0.0%	91.5%	1.2%	1.9%	91.5%	8.5%	
	FUP			6.3%		2.3%	0.2%	9.5%	81.1%	0.0%	0.7%	81.1%	18.9%
	ADW		5.3%	20.6%		4.2%		6.2%	0.0%	63.7%	63.7%	36.3%	
	FDW			6.7%	14.9%	10.5%	11.6%	0.0%	10.3%	0.0%	46.0%	46.0%	54.0%
			Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR
		Predicted Class											

(a) Discriminante Lineal

True Class	Correcto	96.8%	0.2%	0.0%	0.5%	0.0%	0.2%		2.4%	0.0%	96.8%	3.2%		
	AHD		0.6%	78.4%	0.2%	15.3%	0.2%	0.1%	1.2%	2.6%	1.3%	78.4%	21.6%	
	FHD			0.0%	88.7%	0.4%	4.7%				6.3%	88.7%	11.3%	
	AHI		0.1%	4.1%	5.9%	86.0%	0.2%	0.3%	1.8%	1.0%	0.6%	86.0%	14.0%	
	FHI				7.6%	1.1%	78.6%		0.0%		12.6%	78.6%	21.4%	
	AUP		0.9%	1.0%	0.0%	0.5%	0.0%	96.5%	0.2%	0.9%		96.5%	3.5%	
	FUP			0.0%	2.8%	0.1%	0.5%	0.3%		95.2%		1.2%	95.2%	4.8%
	ADW		5.5%	1.2%	0.0%	1.5%	0.0%	3.0%			88.7%	88.7%	11.3%	
	FDW			0.1%	1.8%	4.9%	8.0%	4.3%		4.7%		76.2%	76.2%	23.8%
			Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR	
		Predicted Class												

(b) Discriminante Cuadrático

Figura 4.4 Matriz de confusión de los algoritmos de análisis discriminante

### 4.1.3 Entrenamiento con los algoritmos Naive Bayes

Para el entrenamiento con los algoritmos tipo Bayes se utilizó una distribución Gaussiana y de Núcleo. Luego del entrenamiento con los modelos se obtuvo una precisión de 74,4% para el modelo con distribución

Gaussiana y 79,1% para el modelo con distribución de Núcleo. Los resultados se pueden apreciar en la matriz de confusión de la Figura 4.5.



Figura 4.5 Matriz de confusión de los algoritmos Naive Bayes

#### 4.1.4 Entrenamiento con los algoritmos SVM

Los algoritmos SVM fueron entrenados utilizando las funciones lineales, cuadrática y cúbicas. Una vez concluido el entrenamiento se obtuvo una precisión global de 87,0% para la función lineal, 94,4% de precisión para la

función cuadrática y 94,5% de precisión para una función cúbica. En la Figura 4.6-4.7 se puede observar la matriz de confusión de los resultados del entrenamiento.

True Class	Correcto	98.4%	0.2%		0.0%		0.1%		1.3%		98.4%	1.6%
	AHD	0.0%	72.0%	0.0%	16.8%	0.0%	0.6%	7.0%	1.7%	1.7%	72.0%	28.0%
	FHD			84.8%	1.1%	6.6%		0.7%		6.9%	84.8%	15.2%
	AHI	0.3%	6.9%	3.4%	84.1%	0.3%	2.3%	0.6%	0.8%	1.2%	84.1%	15.9%
	FHI			13.6%	0.6%	72.7%		0.0%		13.0%	72.7%	27.3%
	AUP	0.3%	1.6%		1.0%		95.2%	0.0%	2.0%		95.2%	4.8%
	FUP		9.1%		1.1%	0.2%		88.6%		1.0%	88.6%	11.4%
	ADW	7.2%	2.2%		0.6%		2.0%		88.0%		88.0%	12.0%
	FDW		2.4%	9.4%	6.9%	11.0%		2.7%		67.6%	67.6%	32.4%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW		
											TPR	FNR

(a) Lineal

True Class	Correcto	98.9%	0.0%	0.0%		0.1%		1.0%		98.9%	1.1%	
	AHD	0.1%	92.9%	0.0%	4.4%		0.1%	0.8%	0.8%	0.9%	92.9%	7.1%
	FHD		0.0%	90.7%	0.8%	5.7%		0.0%		2.8%	90.7%	9.3%
	AHI	0.0%	4.1%	1.2%	92.1%	0.6%	0.3%	0.6%	0.4%	0.7%	92.1%	7.9%
	FHI		0.0%	6.2%	0.7%	86.3%				6.8%	86.3%	13.7%
	AUP	0.4%	0.2%		0.3%		97.9%		1.1%		97.9%	2.1%
	FUP		0.7%	0.0%	0.3%	0.0%		97.2%		1.7%	97.2%	2.8%
	ADW	2.7%	0.4%		0.5%		2.0%		94.4%		94.4%	5.6%
	FDW		0.6%	5.0%	1.1%	4.2%		2.5%		86.6%	86.6%	13.4%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW		
											TPR	FNR

(b) Cuadrático

Figura 4.6 Matriz de confusión SVM Lineal y Cuadrático

True Class	Correcto	99.1%	0.0%		0.0%		0.1%		0.8%		99.1%	0.9%
	AHD	0.1%	93.7%	0.0%	4.3%		0.2%	0.9%	0.3%	0.5%	93.7%	6.3%
	FHD		0.0%	83.2%	0.6%	10.6%		0.0%		5.5%	83.2%	16.8%
	AHI	0.1%	2.2%	0.8%	94.8%	0.4%	0.3%	0.4%	0.3%	0.6%	94.8%	5.2%
	FHI		0.0%	6.0%	0.8%	86.6%		0.0%		6.5%	86.6%	13.4%
	AUP	0.4%	0.1%		0.2%		98.0%	0.0%	1.2%		98.0%	2.0%
	FUP		0.6%	0.0%	0.3%	0.1%		97.4%		1.6%	97.4%	2.6%
	ADW	2.3%	0.3%		0.5%		1.7%		95.3%		95.3%	4.7%
	FDW		0.4%	2.0%	0.7%	3.0%		2.2%	0.0%	91.7%	91.7%	8.3%
			Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR
		Predicted Class										

**Figura 4.7 Matriz de confusión SVM-Cubico**

#### 4.1.5 Entrenamiento con los algoritmos KNN

Para el entrenamiento de los modelos K vecinos se hizo uso de las variantes del algoritmo listadas en la Tabla 4.2, en los cuales el parámetro que se varia es el número de vecinos más cercano y la métrica para medir la distancia entre vecinos.

**Tabla 4.2 Parámetros de entrenamiento de KNN**

Algoritmo	Número de vecinos	Métrica de Distancia	Precisión (%)
Fine KNN	5	Euclidean	95,44
Medium KNN	15		95,69
Coarse KNN	50		95,53
Cubic KNN	15	Minkowski	95,67
Weighted KNN	15	Euclidean	95,48

Al finalizar el entrenamiento los resultados obtenidos se indican en las matrices de confusión de la Figura 4.8-4.9. Se obtiene una precisión global de 95,7% para el modelo Medium y Cubic.

True Class	Correcto	99.4%	0.0%		0.0%		0.1%		0.4%		99.4%	0.6%
	AHD	0.1%	95.0%		2.9%		0.1%	1.0%	0.4%	0.5%	95.0%	5.0%
	FHD			90.4%	0.6%	5.7%		0.0%		3.2%	90.4%	9.6%
	AHI	0.0%	2.8%	1.0%	93.9%	0.5%	0.2%	0.4%	0.4%	0.8%	93.9%	6.1%
	FHI		0.0%	6.1%	0.7%	88.9%		0.0%		4.2%	88.9%	11.1%
	AUP	0.5%	0.2%		0.2%		97.6%	0.0%	1.5%		97.6%	2.4%
	FUP	0.0%	0.7%	0.0%	0.3%	0.0%		97.2%		1.8%	97.2%	2.8%
	ADW	3.0%	0.3%		0.5%		1.8%		94.3%		94.3%	5.7%
	FDW		0.5%	3.0%	0.4%	3.2%		2.2%		90.7%	90.7%	9.3%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR

(a) Fine KNN

True Class	Correcto	99.4%	0.0%		0.0%		0.1%		0.5%		99.4%	0.6%
	AHD	0.0%	94.8%		2.9%		0.1%	1.2%	0.4%	0.6%	94.8%	5.2%
	FHD			91.0%	0.6%	5.3%		0.0%		3.0%	91.0%	9.0%
	AHI	0.0%	2.3%	1.0%	94.4%	0.5%	0.2%	0.4%	0.3%	0.7%	94.4%	5.6%
	FHI			5.4%	0.7%	89.9%		0.0%		4.0%	89.9%	10.1%
	AUP	0.3%	0.2%		0.1%		98.0%	0.1%	1.3%		98.0%	2.0%
	FUP		0.5%	0.0%	0.3%	0.0%		97.6%		1.5%	97.6%	2.4%
	ADW	2.5%	0.3%		0.5%		2.0%		94.8%		94.8%	5.2%
	FDW		0.5%	3.1%	0.5%	2.6%		2.3%		91.0%	91.0%	9.0%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR

(b) Medium KNN

True Class	Correcto	99.2%	0.0%		0.0%		0.1%		0.6%		99.2%	0.8%
	AHD	0.0%	94.2%	0.0%	3.1%		0.2%	1.4%	0.4%	0.6%	94.2%	5.8%
	FHD			90.2%	0.7%	5.6%		0.1%		3.4%	90.2%	9.8%
	AHI	0.0%	2.2%	1.1%	94.2%	0.5%	0.3%	0.5%	0.5%	0.8%	94.2%	5.8%
	FHI			4.8%	0.7%	90.2%		0.0%		4.2%	90.2%	9.8%
	AUP	0.2%	0.3%		0.2%		98.1%	0.1%	1.1%		98.1%	1.9%
	FUP		0.4%	0.0%	0.3%	0.0%		97.8%		1.4%	97.8%	2.2%
	ADW	2.4%	0.3%		0.5%		2.1%	0.0%	94.7%		94.7%	5.3%
	FDW		0.4%	3.0%	0.5%	2.4%		2.6%		91.0%	91.0%	9.0%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR

(c) Coarse KNN

Figura 4.8 Matriz de confusión Fine-Medium-Coarse KNN

True Class	Correcto	99.4%	0.0%	0.0%	0.0%	0.1%	0.5%	0.0%	0.0%	99.4%	0.6%	
	AHD	0.0%	94.8%	0.0%	2.9%	0.1%	1.2%	0.4%	0.6%	94.8%	5.2%	
	FHD	0.0%	0.0%	91.1%	0.6%	5.2%	0.0%	0.0%	3.0%	91.1%	8.9%	
	AHI	0.0%	2.3%	1.0%	94.4%	0.5%	0.3%	0.4%	0.7%	94.4%	5.6%	
	FHI	0.0%	0.0%	5.4%	0.7%	89.9%	0.0%	0.0%	4.0%	89.9%	10.1%	
	AUP	0.3%	0.3%	0.0%	0.2%	0.0%	97.9%	0.1%	1.3%	97.9%	2.1%	
	FUP	0.0%	0.6%	0.0%	0.3%	0.0%	0.0%	97.6%	1.5%	97.6%	2.4%	
	ADW	2.5%	0.3%	0.0%	0.5%	0.0%	0.0%	0.0%	94.7%	94.7%	5.3%	
	FDW	0.0%	0.5%	3.2%	0.5%	2.7%	0.0%	2.3%	0.0%	90.9%	90.9%	9.1%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR

(a) Cubic KNN

True Class	Correcto	99.4%	0.0%	0.0%	0.0%	0.1%	0.5%	0.0%	0.0%	99.4%	0.6%	
	AHD	0.0%	94.9%	0.0%	2.8%	0.1%	1.2%	0.4%	0.6%	94.9%	5.1%	
	FHD	0.0%	0.0%	90.4%	0.6%	5.8%	0.0%	0.0%	3.2%	90.4%	9.6%	
	AHI	0.0%	2.7%	0.9%	94.0%	0.5%	0.3%	0.4%	0.8%	94.0%	6.0%	
	FHI	0.0%	0.0%	5.5%	0.7%	89.7%	0.0%	0.0%	4.0%	89.7%	10.3%	
	AUP	0.4%	0.2%	0.0%	0.2%	0.0%	97.9%	0.0%	1.3%	97.9%	2.1%	
	FUP	0.0%	0.6%	0.0%	0.3%	0.0%	0.0%	96.9%	2.2%	96.9%	3.1%	
	ADW	2.7%	0.3%	0.0%	0.6%	0.0%	0.0%	0.0%	94.7%	94.7%	5.3%	
	FDW	0.0%	0.4%	2.9%	0.5%	3.4%	0.0%	2.2%	0.0%	90.6%	90.6%	9.4%
		Correcto	AHD	FHD	AHI	FHI	AUP	FUP	ADW	FDW	TPR	FNR

(b) Weighted KNN

Figura 4.9 Matriz de confusión Cubic-Weighted KNN

## 4.2 Comparativo de eficiencia por algoritmo de predicción

En esta sección se realiza el análisis y comparación de las métricas obtenidas luego de haber culminado el entrenamiento de los diferentes algoritmos. Se tomará como principal parámetro analizar la eficiencia presentada en todas las matrices de confusión indicadas en la sección 4.1. Esto con el fin de determinar cuál de los diferentes algoritmos entrenados es el más eficiente para el caso de estudio.

**Tabla 4.3 Métricas de los algoritmos entrenados**

#	Tipo de Modelo	Estado	Precisión (%)	Costo de Validación	Tiempo de Entrenamiento (seg)
1	Fine Tree	Trained	87,15	18002	33
2	Medium Tree	Trained	78,67	29886	56
3	Coarse Tree	Trained	57,98	58868	42
4	Linear Discriminant	Trained	75,96	33688	15
5	Quadratic Discriminant	Trained	89,50	14717	16
6	Gaussian Naive Bayes	Trained	74,44	35819	12
7	Kernel Naive Bayes	Trained	79,13	29237	12769
8	Linear SVM	Trained	87,05	18145	5701
9	Quadratic SVM	Trained	94,43	7800	9239
10	Cubic SVM	Trained	94,53	7661	19286
11	Fine KNN	Trained	95,44	6396	56
12	Medium KNN	Trained	95,69	6038	62
13	Coarse KNN	Trained	95,53	6257	82
14	Cubic KNN	Trained	95,67	6069	510
15	Weighted KNN	Trained	95,48	6328	61

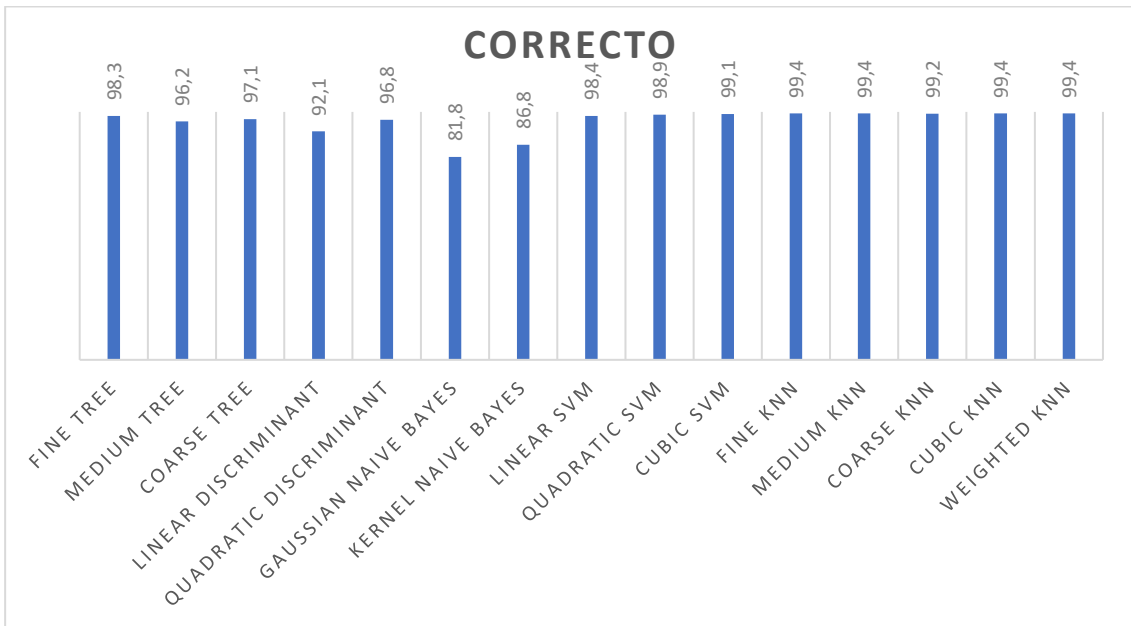
En la Tabla 4.3 se puede observar las diferentes métricas obtenidas luego de realizar el entrenamiento de los algoritmos. También se puede apreciar que los algoritmos KNN tienen la mejor precisión y un menor costo de validación para el caso de estudio.

Otro parámetro importante indicado en la Tabla 4.3 es el tiempo de entrenamiento de cada algoritmo, del cual se observa que los algoritmos SVM necesitan un mayor tiempo para su entrenamiento lo que a su vez exige mayor capacidad de cómputo.

#### **4.3 Comparativo de eficiencia por estado de funcionamiento**

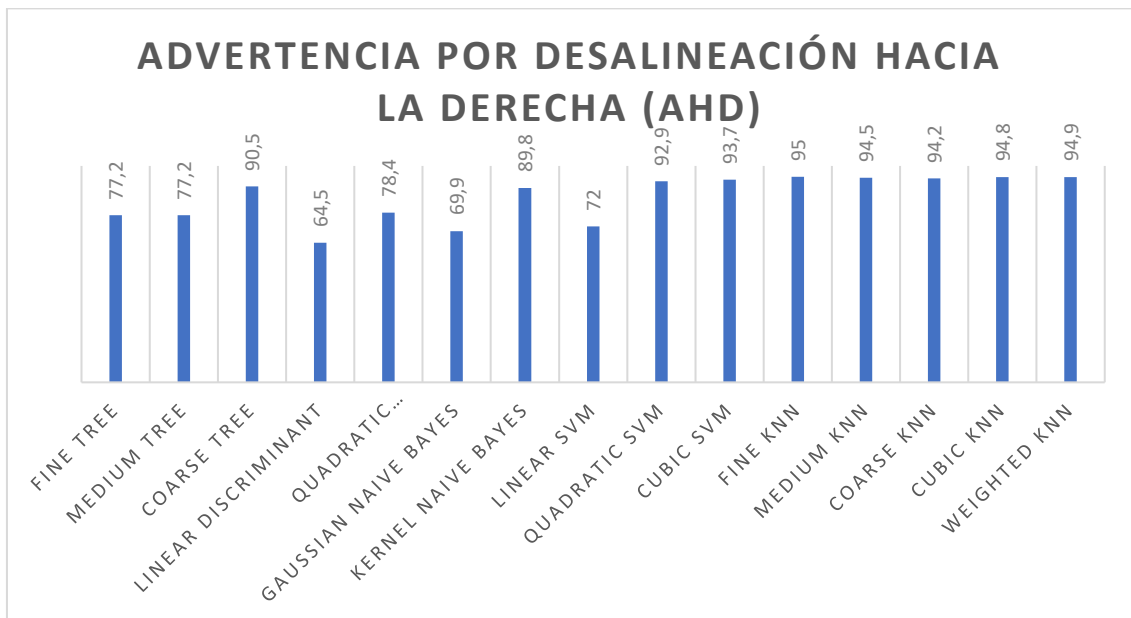
Para la selección del mejor algoritmo se analizó el comportamiento de acuerdo cada estado de funcionamiento. La precisión obtenida con cada uno de los algoritmos de acuerdo con cada estado se indica a continuación:

En el estado de funcionamiento Correcto, se obtuvo las presiones indicadas en la Figura 4.10 en las cuales se observa que todos los algoritmos tienen buenos resultados al predecir este estado.



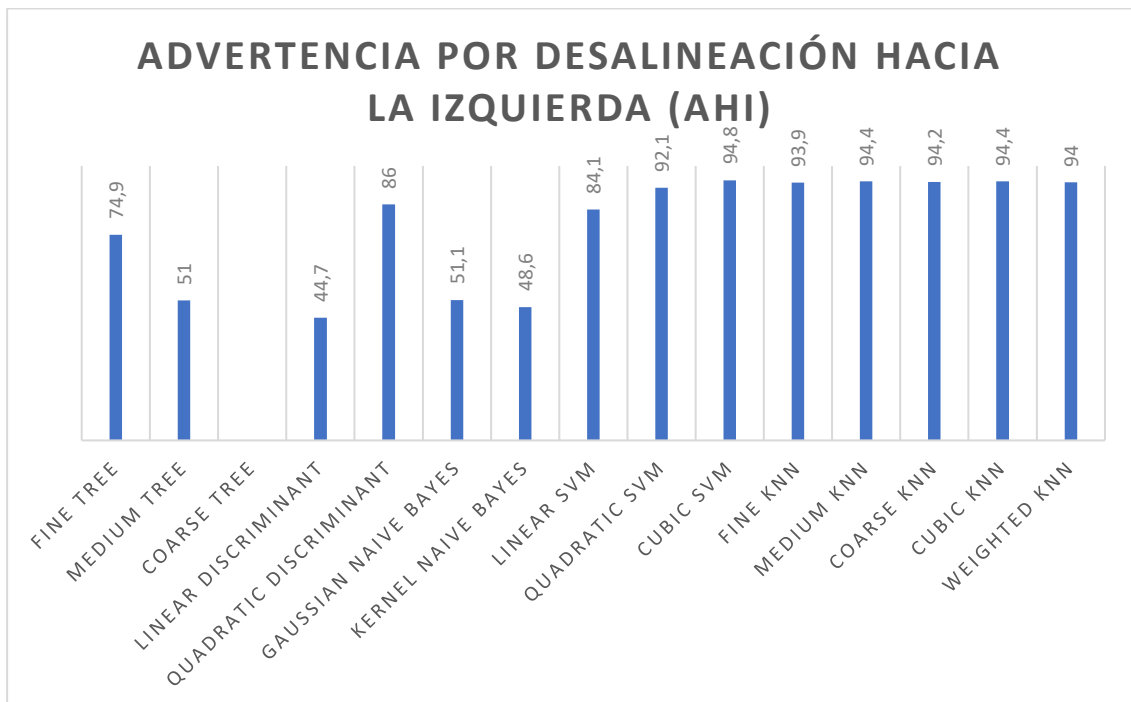
**Figura 4.10 Eficiencia de algoritmos por estado Correcto**

Para el estado de funcionamiento Advertencia, se obtuvieron las presiones indicadas de la Figura 4.11 a la Figura 4.14, se observa que el algoritmo Árbol de Decisión no pueden predecir con exactitud este estado por desalineación hacia la izquierda.

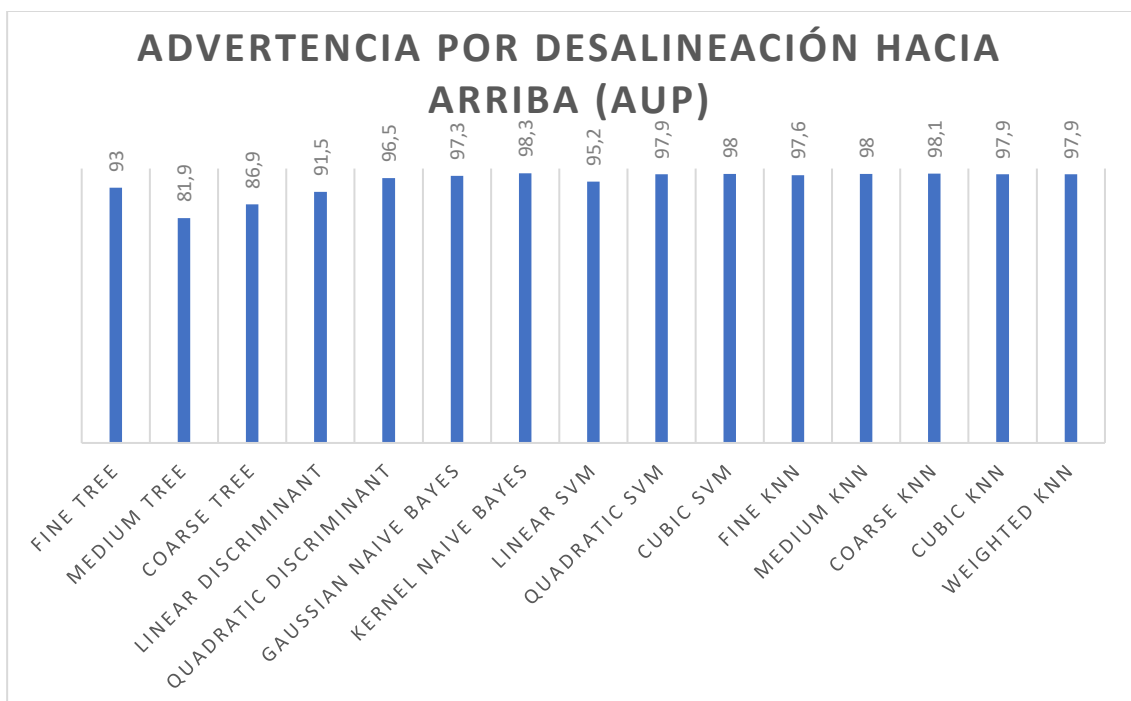


**Figura 4.11 Eficiencia de algoritmos por estado AHD**

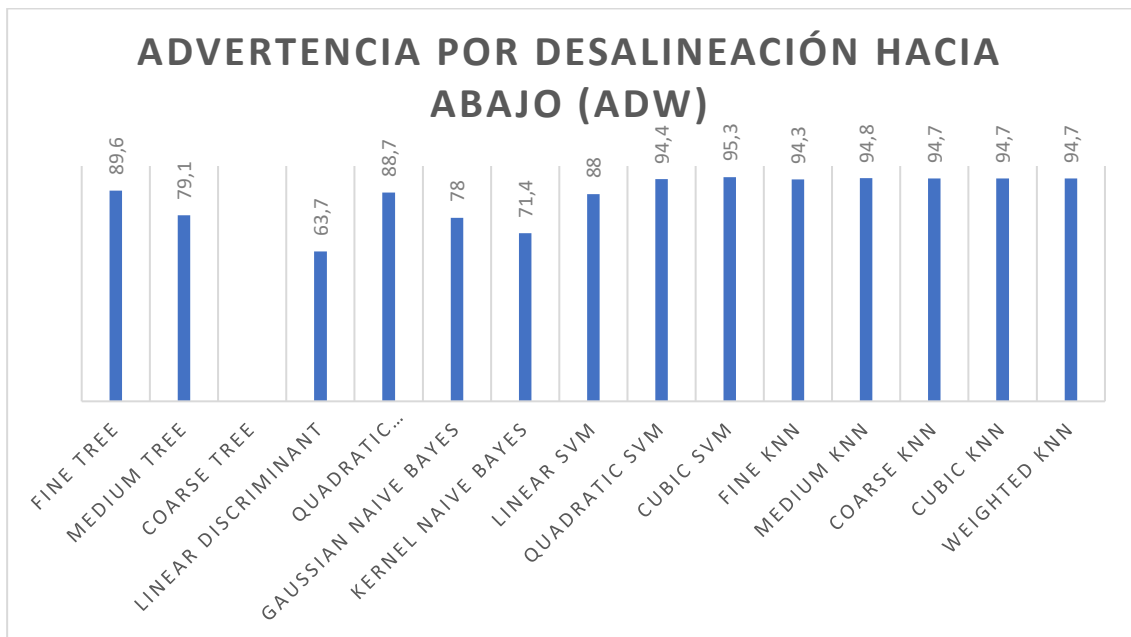




**Figura 4.12** Eficiencia de algoritmos por estado AHI

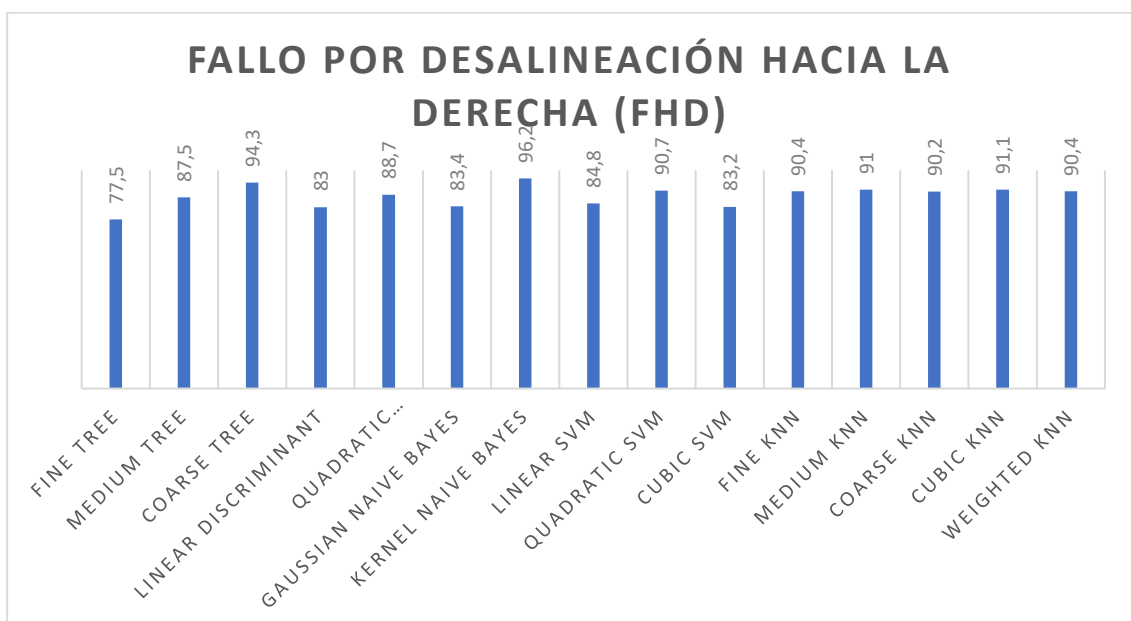


**Figura 4.13** Eficiencia de algoritmos por estado AUP

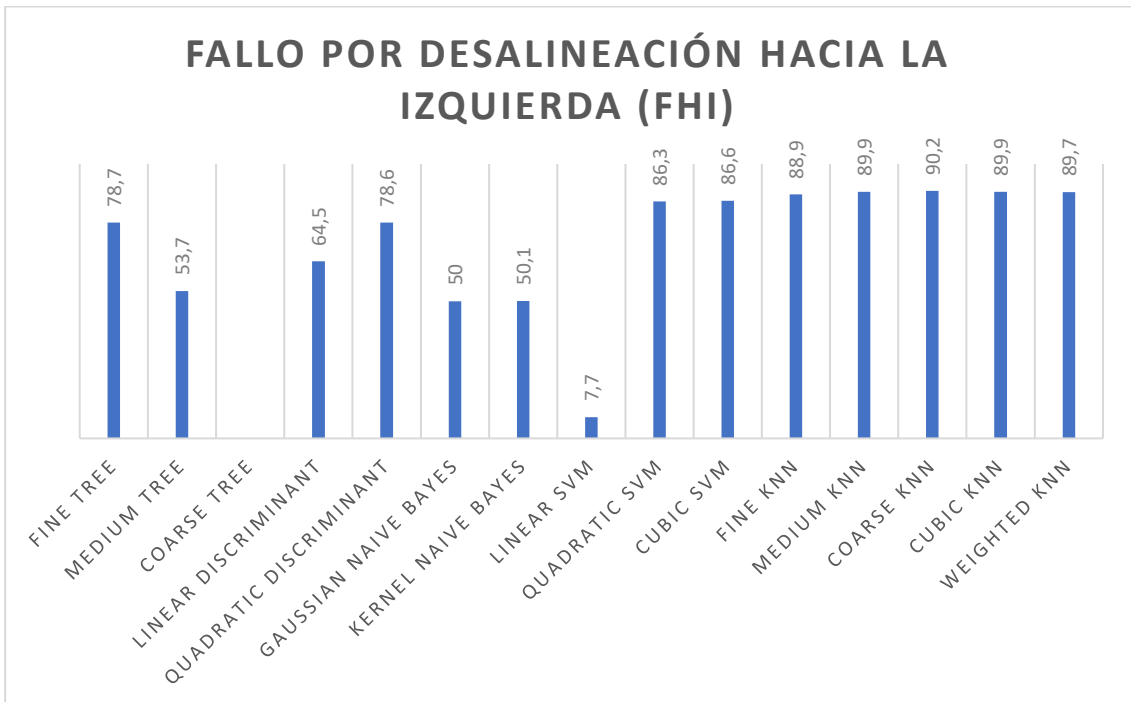


**Figura 4.14 Eficiencia de algoritmos por estado ADW**

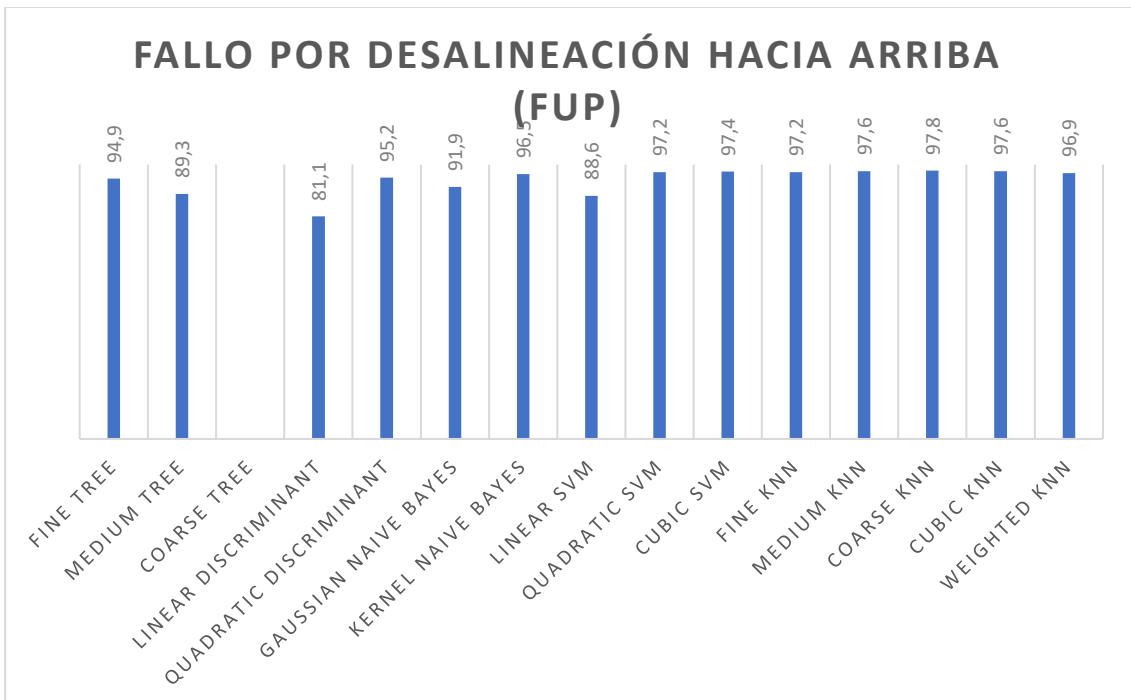
El estado de funcionamiento Fallo, se obtuvo los resultados de precisión como se muestran de la Figura 4.15 a la Figura 4.18. observando que algoritmos como Árbol de Decisión y Bayes tiene una baja precisión al momento de intentar predecir.



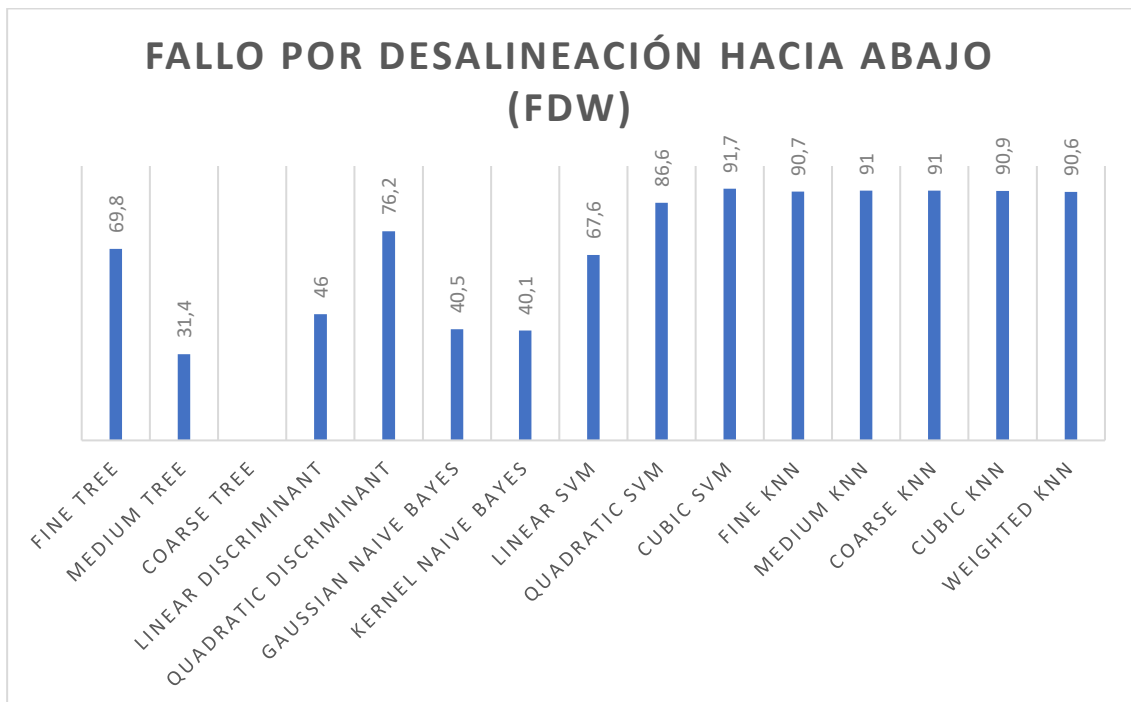
**Figura 4.15 Eficiencia de algoritmos por estado FHD**



**Figura 4.16** Eficiencia de algoritmos por estado FHI



**Figura 4.17** Eficiencia de algoritmos por estado FUP



**Figura 4.18 Eficiencia de algoritmos por estado FDW**

De la Figura 4.10 a la Figura 4.18 se pareció que los algoritmos KNN y los SVM son los algoritmos que mantienen una eficiencia alta de predicción con todos los estados.

# CAPITULO 5

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

- ✓ Se implementó el sistema de adquisición de datos, mediante el uso de equipos inalámbricos de una empresa proveedora de Internet, el aplicativo Zabbix y el protocolo de comunicación SNMP, con la finalidad de monitorizar, recolectar y almacenar los datos provenientes de los equipos inalámbricos de una empresa proveedor de Internet.
- ✓ El procesamiento de los datos históricos mediante la utilización de la matriz de correlación permitió identificar que siete variables están correlacionadas positivamente. El análisis de componentes principales permitió identificar que con las dos primeras componentes principales se puede explicar el 97.5% de la variabilidad.
- ✓ Mediante el entrenamiento de los diferentes algoritmos de aprendizaje autónomo supervisado se pudo determinar que los algoritmos K vecinos más cercanos ofrecen una mejor precisión y un menor costo de validación, por lo cual es la mejor opción para la implementación del algoritmo para la detección temprana de fallas en equipos inalámbricos de una empresa proveedora de servicios de Internet.
- ✓ Para la recreación de los escenarios de funcionamiento (Correcto – Advertencia - Fallo) no se han realizado modificaciones sustanciales a nivel de software, las alteraciones fueron realizadas a nivel de hardware mediante manipulación mecánica de cada uno de los equipos CPE para degradar los niveles de señal que recibe y así lograr recrear cada uno de estos escenarios.

### 5.2 Recomendaciones

- ✓ Realizar el control de datos históricos de los equipos inalámbricos en el servidor Zabbix antes de almacenar en la base de datos.

- ✓ Realizar pruebas de laboratorio para verificar las comunicaciones entre los equipos CPE, el punto de acceso y Zabbix previos a la instalación de los equipos en campo.
- ✓ Utilizar los OID de los fabricantes de los equipos CPE para facilitar el acceso a los datos de monitoreo de cada CPE de forma rápida.
- ✓ Se recomienda ubicar los diferentes equipos CPE en diferentes sitios para dar mayor diversidad al escenario de estudio.
- ✓ Se recomienda realizar futuros estudios generando fallas de software u obstrucciones parciales o totales de la línea de vista entre el equipo CPE y el punto de acceso para la detección de fallas.

# BIBLIOGRAFÍA

Ageeva, Y. (2018, julio 17). *Predictive Maintenance Scheduling with AI and Decision Optimization*. Medium. <https://towardsdatascience.com/predictive-maintenance-scheduling-with-ibm-data-science-experience-and-decision-optimization-25bc5f1b1b99>

Alpaydin, E. (2021). *Machine Learning, revised and updated edition*. MIT Press.

Baghaei, N., & Hunt, R. (2004). Review of quality of service performance in wireless LANs and 3G multimedia application services. *Computer Communications*, 27(17), 1684-1692. <https://doi.org/10.1016/j.comcom.2004.05.019>

Barth, W. (2008). *Nagios, 2nd Edition: System and Network Monitoring*. No Starch Press.

Bedoya-Marrugo, E. A., Vargas-Ortiz, L. E., González-Urango, H. K., & Severiche-Sierra, C. A. (2017). Análisis de componentes principales para caracterización estructural de pymes manufactureras de Cartagena de Indias, Colombia. *Lámpsakos*, 1(17), 52. <https://doi.org/10.21501/21454086.2361>

Bonaccorso, G. (2017). *Machine Learning Algorithms*. Packt Publishing Ltd.

Breiman, L. (2017). *Classification and Regression Trees*. <https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman>

Briceño, C. R. (2004). PROTOCOLO SNMP (PROTOCOLO SENCILLO DE ADMINISTRACIÓN DE REDES). *TELEMATIQUE*, 3(1), Article 1.

- Brownlee, J. (2020). *Master Machine Learning Algorithms* (v1.14).  
<https://machinelearningmastery.com/master-machine-learning-algorithms/>
- Brownlee, J. (2016, marzo 15). Supervised and Unsupervised Machine Learning Algorithms. *Machine Learning Mastery*.  
<https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>
- Cabello, M. V. N. (2010). *Introducción a las Bases de Datos relacionales*. Vision Libros.
- Cáceres, J. H. (2015). *Clustering basado en el algoritmo K-means para la identificación de grupos de pacientes quirúrgicos*. 8.
- Camejo Corona, J., Gonzalez, H., Morell, C., Camejo Corona, J., Gonzalez, H., & Morell, C. (2019). Los principales algoritmos para regresión con salidas múltiples. Una revisión para Big Data. *Revista Cubana de Ciencias Informáticas*, 13(4), 118-150.
- Canizo, M., Onieva, E., Conde, A., Charramendieta, S., & Trujillo, S. (2017a). Real-time predictive maintenance for wind turbines using Big Data frameworks. *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 70-77. <https://doi.org/10.1109/ICPHM.2017.7998308>
- Canizo, M., Onieva, E., Conde, A., Charramendieta, S., & Trujillo, S. (2017b). Real-time predictive maintenance for wind turbines using Big Data frameworks. *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 70-77. <https://doi.org/10.1109/ICPHM.2017.7998308>



- Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137, 106024. <https://doi.org/10.1016/j.cie.2019.106024>
- Cerda, J., & Cifuentes, L. (2012). Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos. *Revista chilena de infectología*, 29(2), 138-141. <https://doi.org/10.4067/S0716-10182012000200003>
- Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, 192, 105361. <https://doi.org/10.1016/j.knosys.2019.105361>
- Clep, M. (2015). *Niveles mínimos de señal en los clientes inalámbricos*. 23.
- Cuenca, D., & Leon, D. (2012). *SUPPORT VECTOR MACHINE*. [https://rstudio-pubs-static.s3.amazonaws.com/570352\\_e34015b16f1a47e883e04c6195d4711f.html](https://rstudio-pubs-static.s3.amazonaws.com/570352_e34015b16f1a47e883e04c6195d4711f.html)
- Delgado, J. K. C., Dulce, E. R., & Toledo, R. A. J. (2016). La importancia del monitoreo en redes de datos. *Boletín Informativo CEI*, 3(2), Article 2. <http://editorial.umariana.edu.co/revistas/index.php/BoletinInformativoCEI/article/view/1086>
- Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, 35(5), 352-359. [https://doi.org/10.1016/S1532-0464\(03\)00034-0](https://doi.org/10.1016/S1532-0464(03)00034-0)

Fedor, M., Schoffstall, M. L., Davin, J. R., & Case, J. D. (1990). *Simple Network Management Protocol (SNMP)* (Request for Comments RFC 1157). Internet Engineering Task Force. <https://doi.org/10.17487/RFC1157>

Garcia, E. O. (2015). *MIKROTIK USER MEETING 2015*. 30.

Garcia-Alvarez, D., & Fuente, M. J. (2011). Estudio comparativo de técnicas de detección de fallos basadas en el Análisis de Componentes Principales (PCA). *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(3), 182-195. <https://doi.org/10.1016/j.riai.2011.06.006>

González, V. L. (2018). *Diagnosis de averías y mantenimiento correctivo de sistemas de automatización industrial*. ELEM0311. IC Editorial.

Holek, R. S. (2002). Tendencias del Mantenimiento Predictivo. *Tecnología Avanzada para Mantenimiento S.A. de C.V*, 7.

IBM. (2022, septiembre 13). *IBM Documentation*. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/spss-statistics/29.0.0?topic=efficiency-correlation-matrix>

IBM Services. (2019, enero 21). *Predictive maintenance | IBM*. IBM Services & Consulting. [https://www.ibm.com/services/technology-support/multivendor-it/predictive-maintenance?lnk=STW\\_US\\_STESCH&lnk2=learn\\_Services&pexp=DEF&psrc=NONE&mhsrc=ibmsearch\\_a&mhq=Predictive%20Maintenance](https://www.ibm.com/services/technology-support/multivendor-it/predictive-maintenance?lnk=STW_US_STESCH&lnk2=learn_Services&pexp=DEF&psrc=NONE&mhsrc=ibmsearch_a&mhq=Predictive%20Maintenance)

Kanawaday, A., & Sane, A. (2017). Machine learning for predictive maintenance of industrial machines using IoT sensor data. *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 87-90. <https://doi.org/10.1109/ICSESS.2017.8342870>

Kharb, L. (2019). *Performance Analytics of Network Monitoring Tools*.

Kumbhar, M. S., Paranjape, N., Bhave, R., & Lahoti, A. (2018). *Comparative study of machine learning algorithms for anomaly detection in Cloud infrastructure*. 4(4), 3.

Lu, S., & Lysecky, R. (2017). Time and Sequence Integrated Runtime Anomaly Detection for Embedded Systems. *ACM Transactions on Embedded Computing Systems*, 17(2), 38:1-38:27. <https://doi.org/10.1145/3122785>

*Matriz de confusión | Interactive Chaos*. (s. f.). Recuperado 31 de marzo de 2022, de <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/matriz-de-confusion>

Narkhede, S. (2021, junio 15). *Understanding AUC - ROC Curve*. Medium. <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

Navada, A., Ansari, A. N., Patil, S., & Sonkamble, B. A. (2011). Overview of use of decision tree algorithms in machine learning. *2011 IEEE Control and System Graduate Research Colloquium*, 37-42. <https://doi.org/10.1109/ICSGRC.2011.5991826>

Olups, R. (2016). *Zabbix Network Monitoring*. Packt Publishing Ltd.

- ORACLE. (2015, junio). *MIB de SNMP de Oracle ILOM - Referencia de gestión de protocolos para SNMP e IPMI de Oracle® ILOM, versión de firmware 3.2.x.*  
[https://docs.oracle.com/cd/E40701\\_01/html/E40347/z4002eb91392027.html#scrolltoc](https://docs.oracle.com/cd/E40701_01/html/E40347/z4002eb91392027.html#scrolltoc)
- Powers, D. M. W. (2020). *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation.*  
<https://doi.org/10.48550/arXiv.2010.16061>
- Primero, D. F., Diaz, J. C., García, L. F., & González-Vargas, A. (2015). MANUAL PARA LA GESTIÓN DEL MANTENIMIENTO CORRECTIVO DE EQUIPOS BIOMÉDICOS EN LA FUNDACIÓN VALLE DEL LILI. *Revista Ingeniería Biomédica*, 9(18), 81-87.
- Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 35-39. <https://doi.org/10.1109/COMITCon.2019.8862451>
- Sánchez, S. E. T., Rodríguez, M. O., Jiménez, A. E., & Soberanes, H. J. P. (2016). Implementación de Algoritmos de Inteligencia Artificial para el Entrenamiento de Redes Neuronales de Segunda Generación. *JÓVENES EN LA CIENCIA*, 2, 6-10.
- Simeone, O. (2018). A Very Brief Introduction to Machine Learning With Applications to Communication Systems. *ArXiv:1808.02342 [Cs, Math]*.  
<http://arxiv.org/abs/1808.02342>

- SolarwWinds. (2022). *Software de administración de TI y plataforma Observability | SolarWinds*. <https://www.solarwinds.com/es/>
- Timmermann, T. (2013). *Criterios para la selección adecuada de una solución de monitoreo de red*. 10.
- Vázquez, T. S. (2014). *Lo secreto del mantenimiento industrial*. Palibrio.
- Vidal, F. (2021, mayo 18). *Mantenimiento Preventivo: Qué es, tipos y cómo hacerlo eficazmente*. <https://www.stelorder.com/blog/mantenimiento-preventivo/>
- Wang, Y., Martonosi, M., & Peh, L.-S. (2007). Predicting link quality using supervised learning in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3), 71-83. <https://doi.org/10.1145/1317425.1317434>
- Zabbix. (2022). *Zabbix Manual*. <https://www.zabbix.com/documentation/5.0/en/manual>

# Anexo 1

## MANUAL DE IMPLANTACIÓN DE ZABBIX

**Paso 1.** Instalar el sistema operativo Raspbian en la placa Raspberry Pi, se puede descargar la imagen del sistema operativo del sitio oficial y escribirlo en una tarjeta SD. En la Figura 1 se puede observar la versión del Raspbian que fue instalado, así como la tarjeta Raspberry Pi física.



**Figura 1 Versión del Raspbian Instalada**

**Paso 2.** Descargar e instalar el servidor Zabbix, interfaz gráfica y el agente Zabbix para esto se debe ingresar a la línea de comandos del servidor Raspbian y descargar Zabbix desde el sitio oficial.

**Paso 3.** Para almacenar los datos provenientes de los equipos CPE monitoreados, se utilizará una base de datos MariaDB. En el proceso de instalación se creó una base de datos con el nombre Zabbix, se configura una contraseña adminDBpassw como contraseña de la base de datos raíz y zabbixDBpassw como contraseña de la base de datos Zabbix.

**Paso 4.** Prueba del servidor Zabbix, una vez realizado todos los pasos anteriores se debe ingresar al servidor Zabbix para lo cual se debe ingresar a "http://IP-SERVIDOR/zabbix", una vez cargado la página de inicio nos mostrara un resumen de todos los

prerrequisitos necesarios para el funcionamiento del servidor Zabbix. Como se puede observar en la Figura 2 todos los prerrequisitos deben de estar en OK.

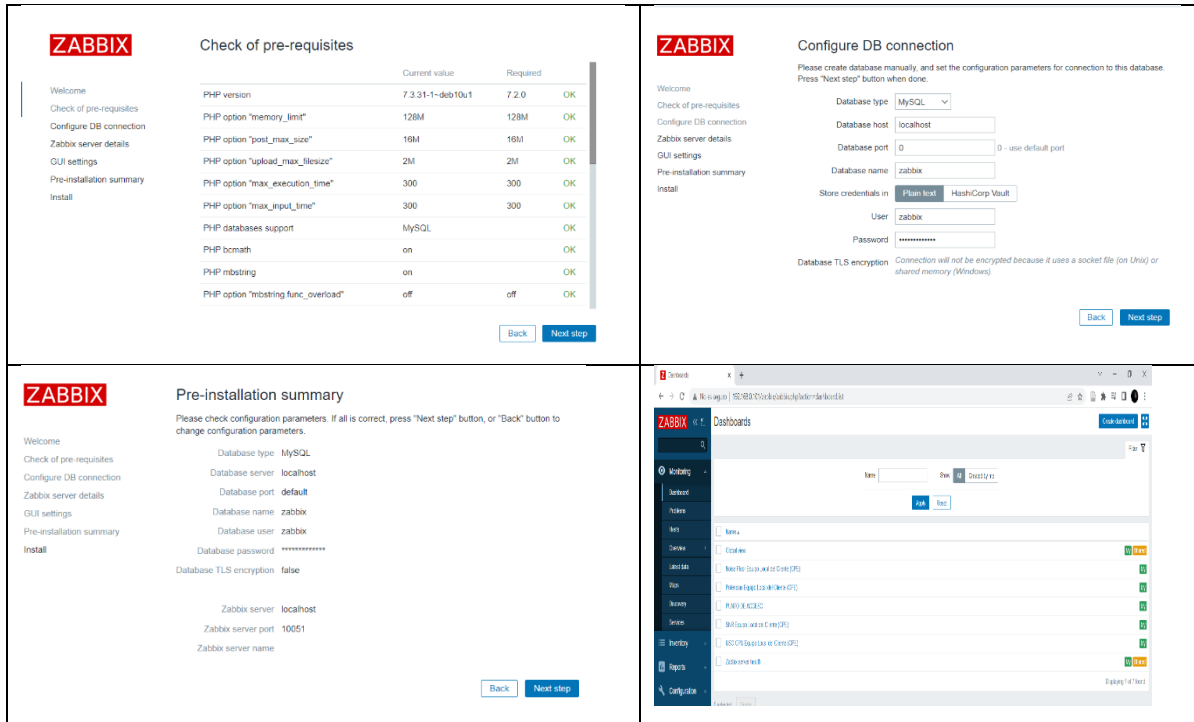


Figura 2 Prueba del servidor Zabbix

**Paso 7.** Para la administración de la base de dato se utiliza phpMyAdmin que es un software de código abierto que facilita la administración de la base de datos MariaDB. Este software permite la importar, exportar, optimizar y eliminar tablas de forma rápida.

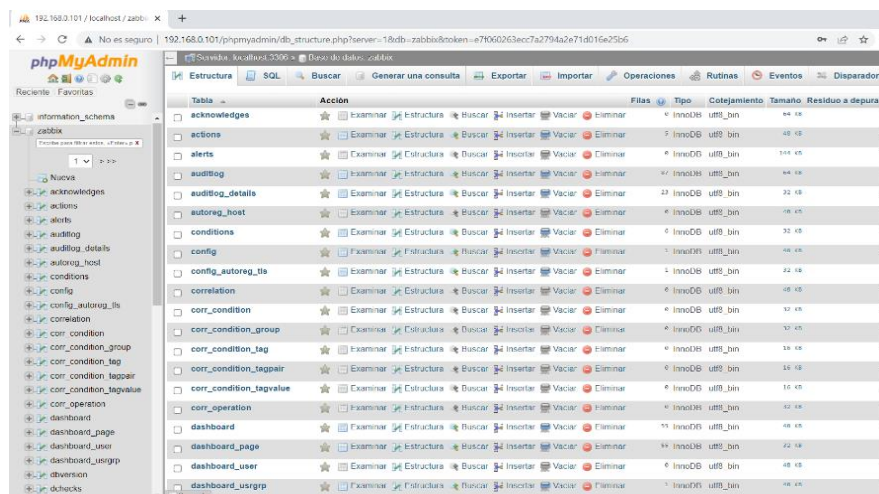
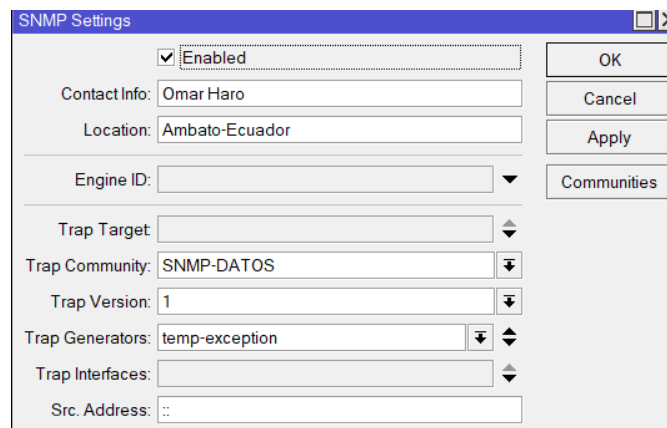


Figura 3 Prueba del servidor Zabbix

Este administrador se utiliza para exportar los datos de los diferentes CPE monitoreados para su posterior análisis y tratamiento. En la Figura 3 se puede observar parte de la estructura de la base datos.

**Paso 8.** Una vez instalado el servidor Zabbix y la base de datos se procedió a realizar la configuración del protocolo SNMP tanto en el punto de acceso y en los equipos CPE para que el servidor Zabbix pueda comunicarse con los equipos y su vez obtener los diferentes datos de dichos equipos. Para esto se configuro los parámetros como se puede observar en la Figura 4.



The image shows a screenshot of a 'SNMP Settings' dialog box. The 'Enabled' checkbox is checked. The 'Contact Info' field contains 'Omar Haro' and the 'Location' field contains 'Ambato-Ecuador'. The 'Engine ID' is a dropdown menu. The 'Trap Target' is a dropdown menu. The 'Trap Community' is 'SNMP-DATOS', 'Trap Version' is '1', and 'Trap Generators' is 'temp-exception'. The 'Trap Interfaces' and 'Src. Address' fields are empty. On the right side, there are buttons for 'OK', 'Cancel', 'Apply', and 'Communities'.

**Figura 4 Configuración SNMP**



## ANEXO 2

### MANUAL DE LA HERRAMIENTA CLASSIFICATION LEARNER

Para el entrenamiento de los diferentes modelos de aprendizaje autónomo se utilizó la herramienta Classification Learner de MATLAB para esto se siguen los siguientes pasos.

**Paso 1.** Se debe cargar el Dataset completo en un solo archivo para este trabajo se desarrolló una función en MATLAB la cual permite cargar en una sola variable todos los archivos “.csv” y guárdalo en una tabla en MATLAB. Esta función se indica a continuación.

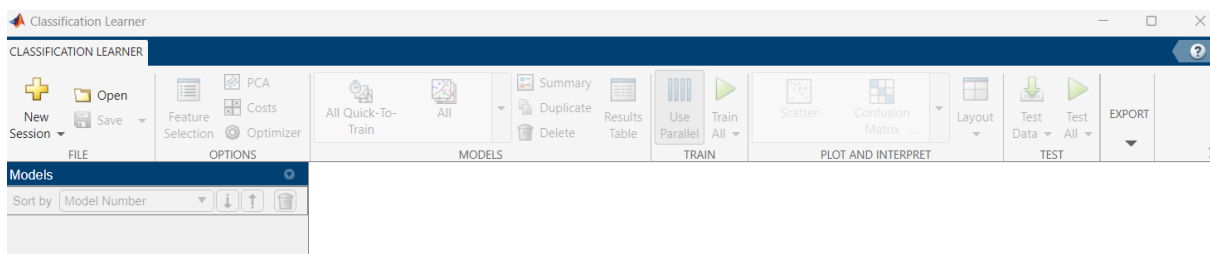
```
% Cargar los datos
filenames = 'CPE*';
files = ['C:\Users\USUARIO\Desktop\TESIS MAESTRIA\Capitulo 3\4. CLASSI MATLAB\Data\'];
filenames '.csv'];
ds = datastore(files,'TreatAsMissing','NA', 'Delimiter',';'. ...
    , 'ReadSize','file');

dataAll = readall(ds);

%Extraer solo datos de los CPE (que no sean datos de #CPE/MUESTRA)

Data = dataAll(:,4:end);
```

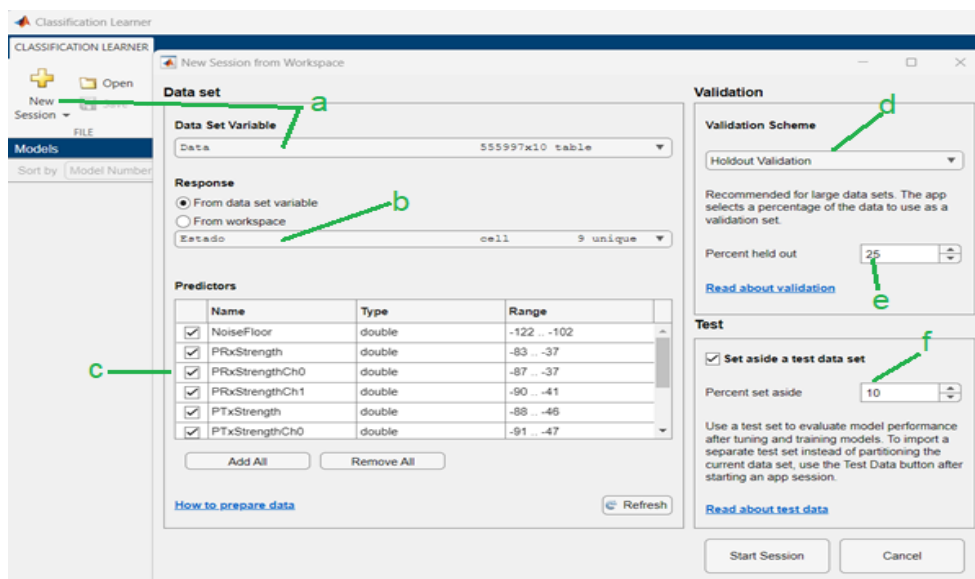
**Paso 2.** Una vez que se cuenta con los datos en una sola variable se ejecuta la herramienta Classification Learner para la carga de datos previo al inicio de entrenamiento. En la Figura 1 se puede observar la pantalla de inicio de la Classification Learner.



**Figura 1** Pantalla de inicio de Classification Learner

**Paso 3.** Para cargar los datos la herramienta Classification Learner permite que los datos sean cargado directamente de un archivo o el uso de una tabla que se encuentre en la memoria de MATLAB como es el caso del presente trabajo. A continuación, se describe los pasos necesarios para la carga del Dataset.

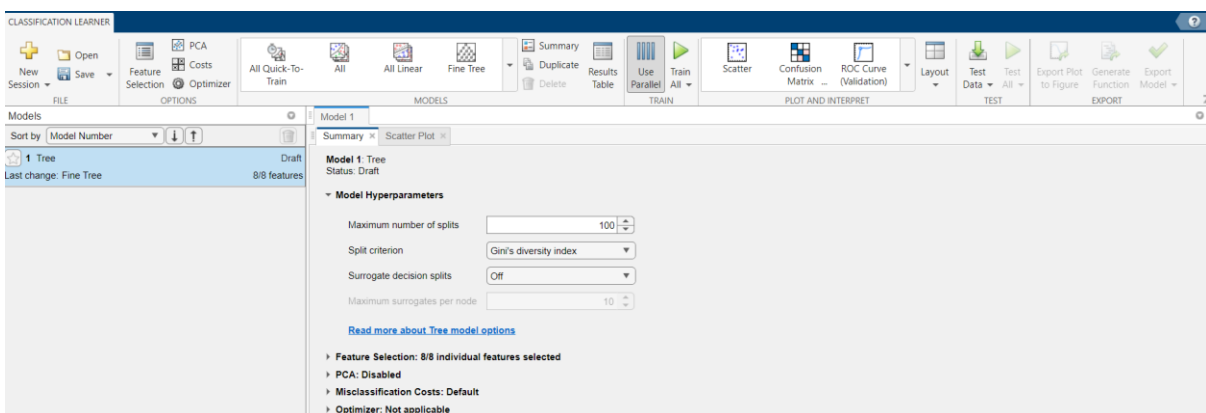
- Iniciar una nueva sección y elegir variable en donde se encuentra almacenada el Dataset en el caso de estudio la variable donde se encuentra almacenado todo el Dataset tiene el nombre de Data.
- Seleccionar la columna de respuesta o etiqueta en el caso de estudio esta columna tiene el nombre de Estado.
- Indicar los predictores o las variables que serán utilizadas para el entrenamiento. En el caso de estudio se utilizarán ocho variables.
- Seleccionar el método de validación al contar con un Dataset grande se utilizará la validación Holdout y así evitar sobre entrenamiento del modelo.
- Indicar el porcentaje del Dataset que será utilizado para la validación en el caso de estudio se utiliza el 25% de todo el Dataset para la prueba.
- Indicar el porcentaje del Dataset utilizado para el la prueba de algoritmo. En el caso de estudio se utiliza un 10% este porcentaje de datos no se utilizarán en el proceso de entrenamiento.



**Figura 2 Carga del Dataset**

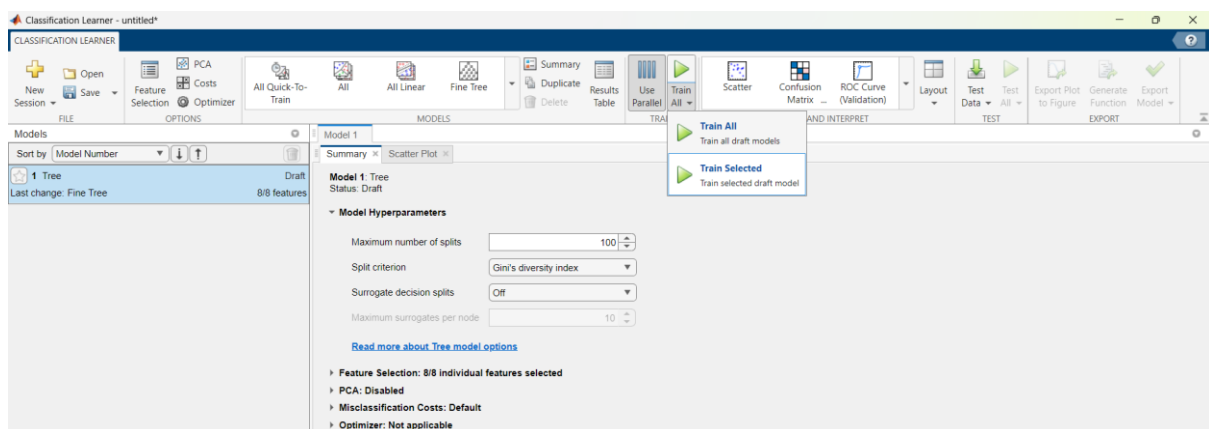
**Paso 4.** Una vez finalizada la carga de datos se debe seleccionar el modelo que se va a entrenar. Todos los modelos de aprendizaje autónomo soportados por MATLAB se encuentran en el apartado “MODEL”.

**Paso 5.** Dependiendo del algoritmo seleccionado se debe indicar ciertos parámetros para el entrenamiento dichos parámetros se pueden ingresar en la pestaña “Summary”. En la Figura 3 se puede observar que el algoritmo seleccionado para el entrenamiento es Fine Tree el cual fue configurado con un máximo de ramificaciones de 100.



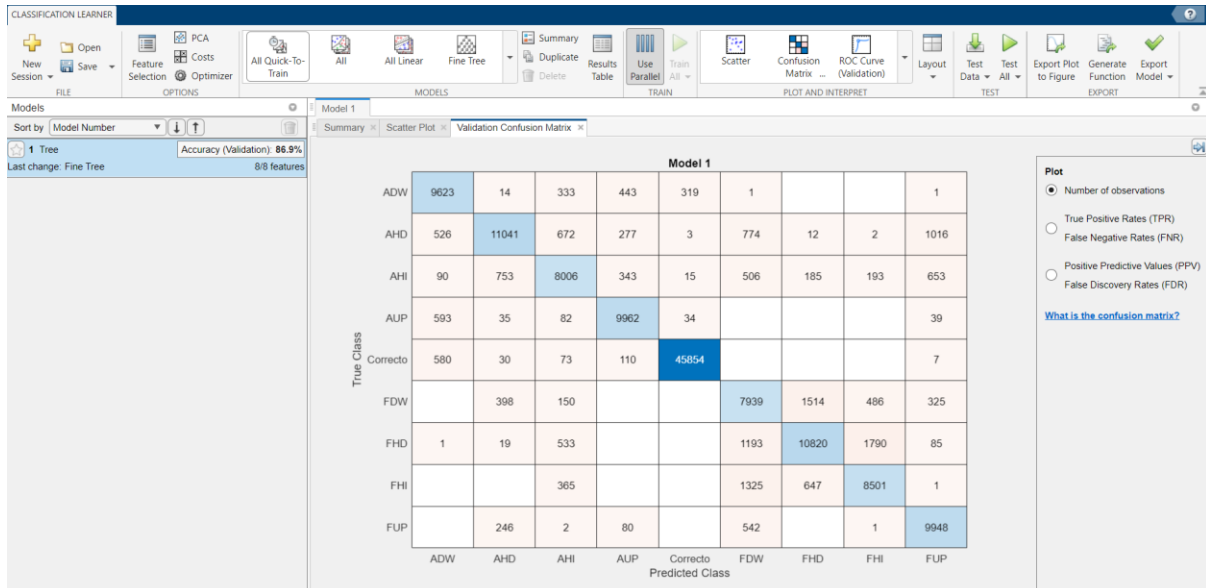
**Figura 3 Configuración de parámetros de algoritmo**

**Paso 6.** Una vez ingresado los parámetros del algoritmo se debe entrenar el modelo para lo cual se debe seleccionar en la pestaña “Train All” o “Train Selected”. Como se indica en la Figura 4.



**Figura 4 Inicio del entrenamiento**

**Paso 7.** Al finalizar el entrenamiento la herramienta Classification Learner nos mostrara el porcentaje de precisión del algoritmo entrenado como se indica en la Figura 5. También se puede observar la matriz de confusión como resultado del entrenamiento.



**Figura 5 Resultado del entrenamiento**