



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Orizaba

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“DESARROLLO DE UN MODELO INTELIGENTE
PARA EL DIAGNÓSTICO DE FALLAS EN
TRANSFORMADORES DE POTENCIA”

QUE PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA INDUSTRIAL

PRESENTA

I.Q. Gustavo Ángel Huerta Pérez

DIRECTOR DE TESIS

Dr. Alberto Alfonso Aguilar Lasserre

CODIRECTOR DE TESIS

M.I.I. Marco Julio Del Moral Argumedo

CODIRECTOR EXTERNO DE TESIS

Dr. Gustavo Arroyo Figueroa



ORIZABA, VERACRUZ, MÉXICO

ABRIL 2024

Orizaba, Veracruz, **20/marzo/2024**
Dependencia: **División de Estudios de
Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

C. HUERTA PÉREZ GUSTAVO ÁNGEL
Candidato(a) a Grado de Maestro(a) en:
INGENIERÍA INDUSTRIAL
PRESENTE

De acuerdo con el reglamento de Titulación vigente de los Centros e Institutos Tecnológicos Federales del Tecnológico Nacional de México, de la Secretaría de Educación Pública, y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

“DESARROLLO DE UN MODELO INTELIGENTE PARA EL DIAGNÓSTICO DE FALLAS EN TRANSFORMADORES DE POTENCIA”

Comunico a usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®


OFELIA LANDETA ESCAMILLA
**JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN**

CSR/magh



OG-13-F06



Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.
Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx





Orizaba Veracruz, 22/febrero/2024
Asunto: **Revisión de trabajo escrito**

**C. OFELIA LANDETA ESCAMILLA
ENCARGADA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
P R E S E N T E.-**

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.:

HUERTA PÉREZ GUSTAVO ÁNGEL

La cual lleva el título de:

**DESARROLLO DE UN MODELO INTELIGENTE PARA EL DIAGNÓSTICO DE FALLAS EN
TRANSFORMADORES DE POTENCIA**

y concluyen que se acepta.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

PRESIDENTE: **DR. ALBERTO AGUILAR LASSERRE**

FIRMA

SECRETARIO: **MII JULIO DEL MORAL ARGUMEDO**

FIRMA

VOCAL: **DR. GUSTAVO ARROYO FIGUEROA**

FIRMA

VOCAL SUP.: **MCI MAGNO ANGEL GONZÁLEZ HUERTA**

FIRMA

TA-09-F18



Declaración de originalidad y cesión de derechos

Orizaba, Veracruz, el día 08 del mes de abril del año 2024.

El(la) que suscribe

C. Gustavo Ángel Huerta Pérez

Declaro que esta tesis, que tiene una extensión de 176 cuartillas, ha sido escrita por mí y constituye el registro escrito del trabajo de la tesis titulada

“DESARROLLO DE UN MODELO INTELIGENTE PARA EL DIAGNÓSTICO DE FALLAS EN TRANSFORMADORES DE POTENCIA”

del programa: Maestría en Ingeniería Industrial bajo la asesoría y dirección del (la) Dr. Alberto Alfonso Aguilar Lasserre y codirección M.I.I Marco Julio Del Moral Argumedo y no ha sido sometida en ninguna otra institución previamente.

Todos los datos y las referencias a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal y, en los casos que así lo requieran, cuento con las debidas autorizaciones de quienes poseen los derechos patrimoniales. Por lo tanto, me hago responsable de cualquier litigio o reclamación relacionada con derechos de propiedad intelectual, exonerando de toda responsabilidad al Tecnológico Nacional de México campus Orizaba.

También declaro que, al presentar esta tesis, cedo los derechos del trabajo al Tecnológico Nacional de México campus Orizaba para su difusión, con fines académicos y de investigación, bajo las regulaciones propias de la institución y que si existe algún acuerdo de confidencialidad de la información lo haré saber en forma escrita para que se omitan las secciones correspondientes.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: depi_orizaba@tecnm.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente de este.

Gustavo Ángel Huerta Pérez

Nombre y firma

AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a todas las personas que han contribuido de alguna manera al desarrollo y éxito de este proyecto:

Quiero agradecer a mis asesores, el Dr. Alberto Alfonso Aguilar Lasserre, M.I.I. Marco Julio del Moral Argumedo y al Dr. Gustavo Arroyo Figueroa, por brindarme la oportunidad de realizar este proyecto y por su dedicación, paciencia y orientación invaluable a lo largo de este proceso. Su apoyo y mentoría fueron fundamentales para alcanzar los resultados obtenidos.

Agradezco sinceramente a todos los profesores que me impartieron clases durante mi formación académica. Su conocimiento y enseñanzas han sido la base sobre lo cual he construido este proyecto y han contribuido significativamente a mi desarrollo profesional.

Quiero expresar mi gratitud a mis amigos de toda la carrera, quienes siempre han sido una fuente de motivación y apoyo incondicional. En especial a mi amiga Quetzalli, su compañerismo, ánimo y apoyo fueron fundamental para superar los desafíos académicos y personales.

Por último, pero no menos importante, quiero agradecer a mi familia por ser mi principal fuente de inspiración. A mi hermano Francisco, por su constante apoyo y consejos sabios, y a mi madre, por su amor incondicional y dedicación incansable. Su influencia positiva ha sido fundamental en mi vida y en el logro de mis metas.

A todos ustedes, mi más sincero agradecimiento por formar parte de este viaje y por contribuir al éxito de este proyecto. Sin su apoyo y aliento, este logro no habría sido posible.

DEDICATORIA

Quiero dedicarles este trabajo a todas las personas que han cruzado mi camino, dejando una huella imborrable en mi corazón. Cada encuentro, cada conversación y cada experiencia compartida ha contribuido a mi desarrollo y formación como individuo.

Gracias a todos ustedes por ser parte de mi viaje, por inspirarme, motivarme y ayudarme a alcanzar mis metas. Su presencia en mi vida ha sido un regalo invaluable que atesoraré siempre con gratitud y cariño.

RESUMEN

Los transformadores de potencia son equipos de gran importancia y su disponibilidad es fundamental para la seguridad y continuidad del suministro eléctrico para los usuarios domésticos e industriales. La sobrecarga en la capacidad nominal de los transformadores, la reducción en gastos de mantenimiento y la austeridad en la inversión, son los desafíos que deben enfrentar las compañías eléctricas en la actualidad. Durante su ciclo de vida, el transformador está expuesto a una serie de condiciones ambientales y operacionales que afectan a su comportamiento, especialmente si ellas exceden a las condiciones de diseño.

Los transformadores se ven afectados por el tiempo de operación (horas), por esta razón, a medida que lleva al tiempo de operación para el cual fue diseñado, se incrementa su probabilidad de falla. Existen diferentes métodos de diagnóstico que utilizan técnicas inteligentes para la detección de fallas en los transformadores de potencia. Estos métodos, aunque tienen buenos resultados, son difíciles de ejecutar en la práctica y encuentran restricciones en la detección de los fallos.

En general, para decidir si remover, reparar o reemplazar un transformador en presencia de fallas térmicas, se requiere determinar la severidad de la falla, por lo que se prefiere un diagnóstico de fallas en transformadores (DFT) más fino.

Es por ello que el DFT requiere de métodos más robustos basados en algoritmos de Inteligencia Artificial. Esta propuesta de investigación se presenta el desarrollo de un modelo de DFT basado en algoritmos de Machine Learning (ML) y la Lógica Difusa (LD). Los algoritmos de ML utilizan datos del estado de salud de los transformadores y el modelo de Lógica Difusa utiliza la experiencia y conocimiento de los expertos; con el objetivo de mejorar el rendimiento de DFT y tratar de evitar los inconvenientes mencionados anteriormente. Los métodos de IA brindan una gran flexibilidad: pueden manejar relaciones lineales y no lineales, son resistentes al ruido, y brindan un alto rendimiento en la clasificación de fallas. Los algoritmos de machine Learning utilizados fueron: Decision Tree Classification, Support Vector Machine, XGBoost, K-Nearest Neighbors, con rendimientos no mayores al 70%. En contraste los modelos de LD: binario y multiclase, obtuvieron rendimiento de 0.96 y 0.80 respectivamente.

CONTENIDO

AGRADECIMIENTOS	i
DEDICATORIA	ii
RESUMEN	iii
CONTENIDO	iv
ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE TABLAS	xi
CAPÍTULO I. GENERALIDADES	1
1.1 Introducción	2
1.2 Descripción de la empresa (colaboración)	4
1.3 Planteamiento del problema.....	6
1.4 Hipótesis	7
1.5 Objetivo general.....	7
1.6 Objetivos específicos	8
1.7 Justificación	8
CAPÍTULO II. MARCO TEÓRICO	10
2.1 Transformadores de potencia	11
2.2 Índice de salud	13
2.2.1 Descargas parciales acústicas (DP acústicas)	14
2.2.2 Gases disueltos en el aceite (DGA).....	15
2.2.2.1 Dornenburg	15
2.2.2.2 Rogers	16
2.2.2.3 Duval.....	17
2.2.2.4 Técnica de Gases Claves	19
2.2.3 Rigidez Dieléctrica.....	19
2.2.4 Contenido de Humedad.....	19
2.2.5 Análisis de Contenido de Furanos.....	20
2.2.6 Diagnóstico de devanados.....	21
2.3 Monitoreo para la detección y prevención de fallas de transformadores de potencia.....	22

2.4	Inteligencia Artificial	24
2.4.1	Lógica Difusa.....	25
2.5	Aprendizaje automático	28
2.6	Aprendizaje automático supervisado	29
2.6.1	Aprendizaje Automático Supervisado de Clasificación.....	30
2.6.2	Aprendizaje Automático Supervisado de Regresión (SMLR)	31
2.7	Aprendizaje no Supervisado	32
2.8	Algoritmos de aprendizaje supervisado	34
2.8.1	Máquinas de soporte vectorial (Support Vector Machines o SVM)	34
2.8.2	Decision Trees.....	35
2.8.3	Random Forest	37
2.8.4	Gradient Boosting	39
2.8.5	XGboost	40
2.8.6	Naive Bayes	40
2.8.7	K Nearest Neighbor	41
2.8.8	Linear Discriminant Analysis	42
2.8.9	Quadratic Discriminant Analysis	42
2.9	Hiperparámetros.....	43
2.10	Métricas de desempeño	44
2.11	Conclusión	47
CAPÍTULO III. ESTADO DEL ARTE.....		48
3.1	Introducción	49
3.2	Lógica difusa para la detección de fallas	49
3.3	Machine Learning para el diagnóstico de transformadores	50
3.4	Redes neuronales artificiales en la detección de fallas	51
3.5	Algoritmos genéticos en la estimación de parámetros de transformadores	51
3.6	Trabajos de inteligencia artificial relacionados con el diagnóstico de fallas en transformadores	52
3.7	Conclusión	55
CAPÍTULO IV. METODOLOGÍA		56
4.1	Metodología	57
4.2.	Análisis General.....	61
4.2.1.	Definición del problema bajo su dominio	61

4.2.2.	Determinación del tipo de problema	62
4.2.3.	Generación de supuestos	62
4.2.4.	Propuesta.....	63
4.3.	Obtención de datos.....	63
4.3.1.	Determinación de datos necesarios	63
4.3.2.	Obtención de la base de datos	64
4.3.3.	Entorno de trabajo.....	66
4.3.3.1.	Jupyter Notebooks.....	67
4.3.3.2.	Configuración.....	68
4.3.4.	Datos	69
4.4.	Exploración	69
4.4.1.	Análisis de atributos.....	69
4.4.1.1.	Atributos para la clasificación.....	69
4.4.1.2.	Conteo de clases y categorías.....	71
4.4.2.	Aplicación de Estadísticas	71
4.4.2.1.	Estadística descriptiva a la BD.....	71
4.4.3.	Información.....	73
4.5.	Procesamiento	73
4.6.	Elección algorítmica	74
4.6.1	Prueba de algoritmos.....	74
4.6.2	Métricas de desempeño.....	75
4.6.3	Evaluación del algoritmo Quadratic Discriminant Analysis.....	75
4.6.4	Evaluación del algoritmo Gradient Boosting.....	79
4.6.5	Evaluación del algoritmo K-Nearest Neighbors	82
4.6.6	Evaluación del algoritmo XGBoost	85
4.6.7	Elección para la optimización	88
4.6.8	Optimización del mejor modelo.....	89
4.6.8.1	Optimización de K-Nearest Neighbors	89
4.6.8.2	Optimización de XGBoost	93
4.7.	Modelación Lógica Difusa.....	96
4.7.1.	Modelo de lógica difusa de clasificación binaria.....	98
4.7.1.1	Establecimiento de los parámetros de las variables de entrada.....	99
4.7.1.2	Establecimiento de los parámetros de la variable de salida	101

4.7.1.3	Reglas de inferencia	101
4.7.1.4	Modelación en Matlab	103
4.7.2.	Modelo de lógica difusa de clasificación multiclase.....	108
4.7.2.1	Establecimiento de los parámetros de las variables de entrada.....	109
4.7.2.2	Establecimiento de los parámetros de la variable de salida	110
4.7.2.3	Reglas de inferencia.....	110
4.7.2.4	Modelación en Matlab	112
4.7.3.	Modelo de lógica difusa de clasificación multiclase (Versión 2)	116
4.7.3.1	Establecimiento de los parámetros de las variables de entrada.....	116
4.7.3.2	Establecimiento de los parámetros de la variable de salida	117
4.7.3.3	Reglas de inferencia.....	118
4.7.3.4	Modelación en Matlab	120
4.8.	Conclusión	124
CAPÍTULO V. RESULTADOS		125
5.1.	Resultados de los modelos de algoritmos de ML.....	126
5.1.1.	Evaluación del modelo del algoritmo optimizado de K-Nearest Neighbor	134
5.1.2.	Matriz de confusión	135
5.1.3.	Curva ROC.....	136
5.1.4.	Curva PR.....	137
5.1.5.	Ganancia acumulativa	138
5.2.	Resultados de los modelos de lógica difusa.....	139
5.2.1.	Evaluación del modelo de lógica difusa con salida binaria	140
5.2.1.1	Matriz de confusión	141
5.2.2.	Evaluación del modelo de lógica difusa con salida multiclase	142
5.2.2.1	Matriz de confusión	142
5.2.3.	Evaluación del modelo de lógica difusa con salida multiclase (Versión 2).....	143
5.2.3.1	Matriz de confusión	143
5.2.	Conclusión	144
Conclusiones.....		148
Recomendaciones		149
Referencias bibliográficas.....		151
Anexo 1		155

Anexo 2.....	162
Anexo 3.....	163

ÍNDICE DE FIGURAS

Figura 2.1 Transformador de potencia.....	11
Figura 2.2. Detección de DP acústica en un Transformador.....	14
Figura 2.3. Ubicación de avería en el triángulo de Duval.....	18
Figura 2.4. Circuito equivalente en alta frecuencia de un transformador	22
Figura 2.5. Esquematización de la Lógica Difusa.....	27
Figura 2.6. Elementos neuronales	¡Error! Marcador no definido.
Figura 2.7. Red Neuronal Artificial de propagación hacia adelante ...	¡Error! Marcador no definido.
Figura 2.8. Clasificación de Machine Learning	29
Figura 2.9. Descripción general de un algoritmo de ML supervisado	30
Figura 2.10. Pasos generales del modelo Aprendizaje Automático Supervisado de Regresión	33
Figura 2.11. Pasos generales del modelo Aprendizaje Automático No Supervisado	32
Figura 2.12. Regresión de un árbol de decisión.....	36
Figura 2.13. Clasificación de KNN en dos dimensiones	42
Figura 2.14. Gráfico de límites de decisión para LDA y QDA.....	43
Figura 4.1 a. Metodología para el desarrollo de modelos de ML	59
Figura 4.1 b. Metodología para el desarrollo de modelos de LD.....	60
Figura 4.2. Base de datos proporcionado por el INEEL	66
Figura 4.3. Interfaz de un Jupyter Notebook.....	67
Figura 4.4. Interfaz de Anaconda Navigator.....	68
Figura 4.5. Base de datos capturada por el JN para su procesamiento	70
Figura 4.6. Número de etiquetas por clase.....	72
Figura 4.7. Porcentaje de etiquetas por clase	73

Figura 4.8. Matriz de confusión del algoritmo Quadratic Discriminant Analysis	77
Figura 4.9. Matriz de confusión normalizada del algoritmo Quadratic Discriminant Analysis.....	77
Figura 4.10. Curva ROC del modelo obtenido por el algoritmo Quadratic Discriminant Analysis ...	78
Figura 4.11. Curva PR del modelo obtenido por el algoritmo Quadratic Discriminant Analysis.....	79
Figura 4.12. Matriz de confusión del modelo del algoritmo Gradient Boosting.....	80
Figura 4.13. Matriz de confusión normaliza del modelo del algoritmo Gradient Boosting.....	80
Figura 4.14. Curva ROC del modelo obtenido por el algoritmo Gradient Boosting	81
Figura 4.15. Curva PR del modelo obtenido por el algoritmo Gradient Boosting.....	82
Figura 4.16. Matriz de confusión del modelo del algoritmo K-Nearest Neighbors.....	83
Figura 4.17. Matriz de confusión normalizada del modelo del algoritmo K-Nearest Neighbors	83
Figura 4.18. Curva ROC del modelo obtenido por el algoritmo K-Nearest Neighbors.....	84
Figura 4.19. Curva PR del modelo obtenido por el algoritmo K-Nearest Neighbors	85
Figura 4.20. Matriz de confusión del modelo del algoritmo XGBoost.....	86
Figura 4.21. Matriz de confusión normalizada del modelo del algoritmo XGBoost	86
Figura 4.22. Curva ROC del modelo obtenido por el algoritmo XGBoost.....	87
Figura 4.23. Curva PR del modelo obtenido por el algoritmo XGBoost.....	88
Figura 4.24. Matriz de confusión del modelo optimizado del algoritmo K-Nearest Neighbors.....	91
Figura 4.25. Matriz de confusión normalizada del modelo optimizado del algoritmo K-Nearest Neighbors	91
Figura 4.26. Curva ROC del modelo optimizado obtenido por el algoritmo K-Nearest Neighbors...	92
Figura 4.27. Curva PR del modelo optimizado obtenido por el algoritmo K-Nearest Neighbors	93
Figura 4.28. Matriz de confusión del modelo optimizado del algoritmo XGBoost.....	94
Figura 4.29. Matriz de confusión normalizada del modelo optimizado del algoritmo XGBoost	95
Figura 4.30. Curva ROC del modelo optimizado obtenido por el algoritmo XGBoost.....	95
Figura 4.31. Curva PR del modelo optimizado obtenido por el algoritmo XGBoost	96
Figura 4.32. Esquematación del modelo difuso tipo mamdani	97
Figura 4.33. Ejemplo de una función de pertenencia de tipo gaussiana	99

Figura 4.34. Ventana de diseño para modelos de Lógica Difusa.....	103
Figura 4.35. Parámetros de la variable de entrada H2	104
Figura 4.36. Parámetros de la variable de entrada CH4.....	104
Figura 4.37. Parámetros de la variable de entrada C2H2.....	105
Figura 4.38. Parámetros de la variable de entrada C2H4.....	105
Figura 4.39. Parámetros de la variable de entrada C2H6.....	106
Figura 4.40. Parámetros de la variable de salida Diagnóstico	106
Figura 4.41. Editor de reglas de inferencia en Matlab	107
Figura 4.42. Visualizador de reglas de inferencia en Matlab.....	108
Figura 4.43. Parámetros de la variable de entrada CH4/H2.....	112
Figura 4.44. Parámetros de la variable de entrada C2H2/ C2H4	113
Figura 4.45. Parámetros de la variable de entrada C2H4/ C2H6	113
Figura 4.46. Parámetros de la variable de salida Falla.....	114
Figura 4.47. Editor de reglas de inferencia en Matlab	115
Figura 4.48. Visualizador de reglas de inferencia en Matlab.....	115
Figura 4.49. Parámetros de la variable de entrada CH4/H2.....	120
Figura 4.50. Parámetros de la variable de entrada C2H2/ C2H4	121
Figura 4.51. Parámetros de la variable de entrada C2H4/ C2H6	121
Figura 4.52. Parámetros de la variable de salida Falla.....	122
Figura 4.53. Editor de reglas de inferencia en Matlab	123
Figura 4.54. Visualizador de reglas de inferencia en Matlab.....	123
Figura 5.1. Curva ROC- Macro promedios de cada algoritmo probado.	128
Figura 5.2. Curva PR- Micro promedios de cada algoritmo probado.	129
Figura 5.3. Curva PR- Macro promedios de KNN y XGboost optimizados.....	131
Figura 5.4. Curva PR- Micro promedios de KNN y XGboost optimizados.	131
Figura 5.5. Gráfico de importancia de características de XGBoost.....	132
Figura 5.6. Primer árbol del modelo optimizado de XGBoost.....	133

Figura 5.7. Matriz de confusión obtenida de los valores de predicción versus las etiquetas reales..	135
Figura 5.8. Matriz de confusión normalizada de los valores de predicción versus las etiquetas reales.	136
Figura 5.9. Curva ROC de los valores de predicción versus las etiquetas reales.....	136
Figura 5.10. Curva PR de los valores de predicción versus las etiquetas reales.....	137
Figura 5.10. Curva de ganancia acumulativa de la validación del algoritmo KNN.....	138
Figura 5.11. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD binario.	141
Figura 5.12. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD multifalla.	142
Figura 5.13. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD multifalla 2.	143

ÍNDICE DE TABLAS

Tabla 2.1. División del Transformador de Potencia	12
Tabla 2.2. Interpretación del origen de la avería de acuerdo con Dornenburg.	16
Tabla 2.3. Interpretación de fallas mediante método de Rogers	17
Tabla 2.4. Relación entre contenido de 2-FAL y DP.....	21
Tabla 3.1a. Aportaciones de la revisión de la literatura.....	53
Tabla 3.1b. Aportaciones de la revisión de la literatura.....	54
Tabla 4.1. Análisis de gases disueltos en aceite y otros parámetros de medición.	65
Tabla 4.3. Resultados del análisis exploratorio de la base de datos de entrenamiento.	71
Tabla 4.4. Estadísticos de la base de datos a entrenar.....	72
Tabla 4.5. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo QDA ...	76
Tabla 4.6. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo GB	79
Tabla 4.7. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo KNN ...	82
Tabla 4.8. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo XGB ...	85

Tabla 4.9. Métricas de desempeño de algoritmos seleccionados.....	89
Tabla 4.10. Reporte de clasificación de las métricas precisión, recall y f1 score de optimización KNN. .	90
Tabla 4.11. Reporte de clasificación de las métricas precisión, recall y f1 score de optimización XGB ...	94
Tabla 4.12. Concentraciones de gas (en ppm) del estándar IEEE.....	98
Tabla 4.13. Parámetros de modelación de LD de las variables de entrada.	100
Tabla 4.14. Parámetros de modelación de LD de la variable de salida.....	101
Tabla 4.15. Reglas de inferencia para el modelo de LD de clasificación binaria.	102
Tabla 4.16. Parámetros de modelación de LD multiclase de las variables de entrada.....	109
Tabla 4.17. Parámetros de modelación de LD multiclase de la variable de salida.	110
Tabla 4.18. Reglas de inferencia para el modelo de LD de clasificación multiclase.....	111
Tabla 4.19. Clases de la variable de salida del modelo propuesto.	116
Tabla 4.20. Parámetros de modelación de LD multiclase de las variables de entrada.....	117
Tabla 4.21. Parámetros de modelación de LD multiclase de la variable de salida.	118
Tabla 4.22. Reglas de inferencia para el modelo de LD de clasificación multiclase.....	119
Tabla 5.1. Resultados obtenidos por las diferentes métricas aplicadas a los algoritmos.	127
Tabla 5.2. Resultados obtenidos por las diferentes métricas aplicadas a los algoritmos optimizados.....	130
Tabla 5.3. Reporte de clasificación de las métricas precisión, recall y f1 score del modelo KNN.....	134
Tabla 5.4. Resultados obtenidos por las diferentes métricas aplicadas a los modelos de LD.....	144
Tabla 5.5. Resultados obtenidos por las diferentes métricas aplicadas a los modelos de LD.....	145

CAPÍTULO I. GENERALIDADES

Resumen

En este capítulo se presenta una breve reseña sobre el tema a desarrollar, además se plantea la problemática de estudio, también se define la hipótesis de estudio y posterior a ello se establece el objetivo general y los objetivos específicos. También se presentan las motivaciones para el desarrollo del caso de estudio.

1.1 Introducción

Los transformadores de potencia son equipos primordiales en la distribución de la energía eléctrica. Por ello es de suma importancia brindar el mantenimiento adecuado para evitar rápidos deterioros o bien que este llegue a ser irreparable, ya que esto afecta con gran magnitud aspectos sociales como económicos.

Una compañía eléctrica estará preparada para abordar los retos planteados, sólo si cuenta con una metodología para la gestión óptima de sus activos, es decir, si puede tomar las decisiones correctas y oportunas. En este sentido, se debe valorar el nivel de riesgo para cada unidad. Al valorar el riesgo, se puede establecer una clasificación por mérito y condición de los transformadores, lo que permite la planificación de acciones para el mediano y largo plazo. (Cerón, Orduña, Aponte, & Romero, 2015).

En la actualidad hay más de cien mil transformadores de potencia que operan en los Estados Unidos y más de cuatrocientos mil en todo el mundo (Energy D. (Office of Electricitu Delivery and Energy Reliability), 2014). En México, la compañía de electricidad de Comisión Federal de Electricidad opera más de 60,000 transformadores de diferente tipo y potencia, en las Redes Nacional de Transmisión y Redes Generales de Distribución (CFE, 2019). El transformador es una máquina eficiente y confiable, que, en condiciones normales de operación, está diseñada y construida para tener una vida útil cercana a los 40 años.

El transformador experimenta cambios con el paso del tiempo. En consecuencia, a medida que envejece, aumenta la posibilidad de que presente fallas. De acuerdo con Hartffor Steam Boiler, una de las mayores aseguradoras de transformadores de Estados Unidos, 3 de cada 100 transformadores de una cuantía instalada en la década de 1960, están propensos a fallas, el costo promedio de reemplazo de un transformador de 100 MVA puede alcanzar los US \$2 millones y puede tomar de 18 a 24 meses su construcción (Energy D. (Office of Electricitu Delivery and Energy Reliability), 2014).

Se han empleado diversas estrategias de diagnóstico que hacen uso de técnicas inteligentes para identificar posibles fallos en los transformadores de potencia. A pesar de que estos métodos arrojan resultados positivos, su implementación práctica resulta complicada y enfrentan limitaciones en cuanto a la detección de los fallos. Es por ello por lo que, con los datos históricos obtenidos por

las pruebas normalizadas del estado de salud, se puede alimentar a los modelos basados en técnicas de inteligencia artificial y de este modo se permite determinar la condición del transformador sin que se afecte la continuidad del servicio.

La inteligencia artificial va de la mano de la cuarta revolución industrial, de las cuales parten distintas técnicas que pueden ser aplicadas a esta problemática. Uno de estos temas es el aprendizaje automático (Machine Learning) que utiliza datos sobre las poblaciones para realizar tareas de investigación; así como equipos/componentes del sistema e incluso datos financieros para realizar clúster, clasificación y predicción basadas en modelos estocásticos, en su mayoría apoyados por algoritmos informáticos.

Considerando la ingeniería de mantenimiento, la aplicación del aprendizaje automático se propone para la clasificación de criticidad del equipo, predicciones de regresión de fallas y la estrategia de mantenimiento más avanzada, llamada Prognostic Health Management (PHM), la cual tiene por objetivo definir la vida útil restante del equipo y el estado de salud, en este caso de los transformadores, sobre la base de datos en línea o de testes no destructivos mediante la medición de factores estresantes como la vibración, la tensión, la temperatura, humedad, contenido de gases y otros parámetros físicos que degradan el equipo hasta la falla funcional. (Calixto, 2016).

Teniendo todo eso en cuenta, en este trabajo se propone el desarrollo de un modelo basado inteligencia artificial como lo es el Machine Learning y la Lógica Difusa, con la finalidad de determinar el estado de salud de los transformadores mediante el diagnóstico de fallas y que posteriormente permita tomar las decisiones pertinentes.

Este trabajo se estructura en de la siguiente manera. En primer lugar, se aborda la problemática que será objeto de estudio, seguido de un marco teórico que presenta la información relevante para comprender los aspectos fundamentales del tema. Posteriormente, se realiza una revisión del estado actual de la investigación para obtener una visión general del tema y para destacar la contribución específica que este trabajo realizará al estado de conocimiento existente. Se establecen el objetivo general, así como su desglose específico, y se formula la hipótesis del proyecto. Se plantea y desarrolla la metodología a seguir. Finalmente, se presentan los resultados obtenidos, así como las conclusiones y comentarios finales.

1.2 Descripción de la empresa (colaboración)

El Instituto Nacional de Electricidad y Energías Limpias (INEEL) es el Instituto con el que se colaboró para el desarrollo del proyecto, es un centro de investigación dedicado a dar solución a las necesidades del sector eléctrico y energético. Su misión es apoyar e impulsar el sistema de energía del futuro a través de proyectos de innovación y desarrollo tecnológico, permitiendo ofrecerte soluciones para la modernización y el ahorro económico de su empresa. La propuesta tendrá como usuario principal la Comisión Federal de Electricidad (CFE).

Misión

Impulsar el desarrollo sustentable de la industria energética mediante la innovación.

Visión

Se le acreditan aportaciones únicas a la competitividad de la industria por la formación de ecosistemas de innovación y creación de valor (gobierno, industria y academia) que propician e impulsan la sustentabilidad de la industria energética.

Política

Es compromiso del Instituto Nacional de Electricidad y Energías Limpias satisfacer las necesidades de sus clientes y superar sus expectativas mediante la innovación, la eficacia y la mejora continua de sus procesos, dentro del marco normativo, legal y regulatorio aplicable en materia de gestión de la calidad, desarrollo sustentable, igualdad laboral, gestión del medio ambiente, seguridad y salud en el trabajo.

Objetivos

Según su decreto de creación, y su modificación, se tienen los siguientes objetivos:

- Realizar y promover la investigación científica, el desarrollo experimental y la investigación tecnológica, con la finalidad de resolver los problemas científicos y tecnológicos relacionados con el mejoramiento de la industria eléctrica.
- Contribuir a la difusión e implantación, dentro de la industria eléctrica, de aquellas tecnologías que mejor se adapten al desarrollo económico del país.
- Mantener relaciones efectivas con instituciones similares de otros países y con institutos académicos y de investigación tecnológica en el país.
- Impartir cursos de especialización y actualización de conocimientos en ciencia, tecnología y administración industrial en el área de la industria eléctrica.
- Brindar asesoría a la Comisión Federal de Electricidad (CFE), a la industria de manufacturas eléctricas y a las compañías de ingeniería y de servicios de consultoría relacionadas con la industria eléctrica.
- Proponer a la Secretaría de Energía (SENER) y a la CFE, programas de investigación aplicada y tecnológica, y los correspondientes planes de operación, inversión y financiamiento a corto, mediano y largo plazo.
- Patentar y licenciar las tecnologías desarrolladas y los resultados de la investigación que obtengamos y que resulten procedentes.
- Realizar actos conexos con los anteriores y los demás que se fijen en nuestro estatuto.

1.3 Planteamiento del problema

Los transformadores de potencia son componentes clave para el correcto funcionamiento de las redes de transmisión y distribución. Si bien los transformadores son activos muy confiables, la detección temprana de mecanismos de degradación incipientes es muy importante para prevenir fallas que puedan disminuir su vida útil. La gestión del ciclo de vida de los transformadores de potencia se compone de varias etapas, como la especificación del transformador, el montaje, la puesta en marcha, la operación, el mantenimiento y las operaciones de fin de vida. En particular, para las dos últimas etapas, es de suma importancia contar con herramientas adecuadas para la evaluación del estado de salud de los transformadores de potencia. Estas herramientas son la base para la toma de decisiones de reparación, reacondicionamiento o reemplazo de transformadores, bajo el paraguas de optimización de costos, confiabilidad, disponibilidad y seguridad.

La evaluación del estado o índice de salud (IS) es el proceso de identificar marcadores e índices para determinar y cuantificar el nivel de degradación de los componentes de los transformadores. Las estrategias de IS de transformadores de potencia incluyen pruebas eléctricas y fisicoquímicas exhaustivas, el uso de técnicas en línea y/o fuera de línea, el análisis de parámetros de operación y mantenimiento, y el uso de estrategias basadas en condiciones respaldadas por estándares actuales y conocimiento experto.

Los transformadores utilizan un medio de enfriamiento para reducir el calor generado en el proceso de conversión de carga. El sistema de aislamiento está expuesto a esfuerzos eléctricos, mecánicos y térmicos. Estos fenómenos se consideran normales si fueron considerados por las características de diseño del transformador, en caso contrario se consideran anormales. La gran mayoría de la identificación de las fallas en un transformador se enfocan en el sistema de aislamiento cuya integridad está completamente relacionada con la confiabilidad del transformador. Entre los comportamientos anormales se encuentran las sobrecargas de emergencia, las cenizas de arco, los eventos transitorios, las fallas térmicas, por mencionar algunos. El sistema de aislamiento del transformador se divide en fluido aislante (comúnmente aceite mineral) y aislamiento sólido (papel kraft, cartón prensado y otros materiales).

El aceite juega un papel muy importante proporcionando un aislamiento altamente confiable, funcionando como un refrigerante eficiente, mientras elimina el calor generado en el núcleo y los

devanados durante la operación del transformador. El aceite aislante está hecho de hidrocarburos cuya estructura química se descompone por esfuerzos anormales, produciendo gases disueltos asociados con fallas específicas. Por lo tanto, el análisis de gases disueltos (DGA) es una técnica de diagnóstico ampliamente estudiada para la que ya se dispone de muchas herramientas. Estos se basan en el análisis de cada gas subproducto, su concentración y la interrelación entre ellos. La mayoría de estos métodos se basan en intervalos de relación de gas disuelto que clasifican los transformadores en diferentes fallas. Por lo tanto, los datos de gas cerca de los límites son propensos a malas interpretaciones. Aún más, los métodos clásicos de DGA siempre identifican una falla incluso cuando no está presente, por lo que aún se requiere la evaluación de expertos para determinar con precisión si realmente existe una falla o no. En general, para decidir si remover, reparar o reemplazar un transformador en presencia de fallas térmicas, se requiere determinar la severidad de la falla, por lo que se prefiere un diagnóstico de fallas en transformadores (DFT) más fino. Es por ello por lo que el DFT requiere de métodos automáticos basados en algoritmos de Inteligencia Artificial.

1.4 Hipótesis

El desarrollo del modelo inteligente para el diagnóstico de fallas de transformadores de potencia basado en técnicas de inteligencia artificial logrará determinar fallas presentes en los transformadores, de ese modo, servirá de apoyo a la decisión para su posible mantenimiento o cambio oportuno y así se reducir los daños colaterales que un transformador en mal estado pueda causar.

1.5 Objetivo general

Diseñar y desarrollar un modelo inteligente para el diagnóstico de fallas en transformadores de potencia basado en técnicas de inteligencia artificial utilizando el análisis de gases disueltos en el aceite refrigerante como marcador o indicador para determinar y cuantificar el nivel de degradación (falla) de los componentes de los transformadores.

1.6 Objetivos específicos

- Buscar y recolectar datos de indicadores para determinar el nivel de falla de los componentes de los transformadores.
- Procesar, analizar, y limpiar los datos para utilizar el método de entrenamiento y prueba de algoritmos de machine learning, así como también de lógica difusa.
- Diseñar y construir un modelo inteligente para el diagnóstico de fallas en un transformador de potencia.
- Validar el desempeño del modelo inteligente mediante métricas de evaluación como el porcentaje de precisión, la matriz de confusión y curva ROC.
- Demostrar la capacidad de diagnóstico del modelo inteligente mediante la comparación de otros algoritmos de ML desarrollados y los modelos de Lógica Difusa para tal fin.

1.7 Justificación

Durante todo el ciclo de vida de un transformador de potencia se pueden realizar algunas acciones como: adquirir, mantener, reparar, reubicar, desechar, reemplazar o no hacer nada. Los expertos afirman que el 53% de las fallas de los transformadores se pueden prevenir; p. ej., antes de generar una falla, los esfuerzos térmicos del transformador transforman la composición química del refrigerante del transformador, generando gases disueltos, que pueden ser asociados a fallas, por lo tanto, su detección temprana podría ser clave para evitar una contingencia futura.

Para realizar cualquier acción sobre un transformador de potencia se debe contar con suficiente información y una metodología de actuación eficiente. Cada decisión sobre el transformador tiene un riesgo asociado y cada acción un costo; por lo tanto, resulta clave valorar los riesgos y costos en todos los probables escenarios y así tomar la mejor decisión que conlleve a las afectaciones mínimas.

Es por lo antes mencionado que la inteligencia artificial es una gran alternativa para el diagnóstico del índice de salud de los transformadores, puesto que, dentro de todo el abanico de opciones el Machine Learning puede aprovechar muy bien la base de datos debido a que es apta para la clasificación de criticidad del equipo, predicciones de regresión de fallas y la estrategia de

mantenimiento Prognostic Health Management o bien la vida útil restante del equipo, así como también la Lógica Difusa y dar como resultado un buen modelo de diagnóstico que sirva como apoyo a los expertos en la toma de decisiones.

1.8 Conclusión

En conclusión, el desarrollo de un modelo inteligente basado en técnicas de inteligencia artificial para el diagnóstico de fallas en transformadores de potencia representa una respuesta innovadora y eficiente a los desafíos que enfrenta la gestión de estos equipos cruciales en la distribución de energía eléctrica. La colaboración con el Instituto Nacional de Electricidad y Energías Limpias (INEEL) proporciona un respaldo sólido para este proyecto, alineándolo con la misión de impulsar el desarrollo sustentable de la industria energética.

La justificación de este enfoque radica en la capacidad de la inteligencia artificial, en particular, el machine learning y la lógica difusa, para procesar datos históricos y actuales, identificar patrones para su clasificación y evaluar la condición de los transformadores. Este enfoque ofrece la oportunidad de prevenir fallas, optimizar el mantenimiento, y toma decisiones informadas basadas en riesgos y costos asociados.

Los objetivos específicos, desde la recolección de datos hasta la validación del modelo, proporcionan una metodología sólida para la implementación exitosa del modelo inteligente. La propuesta de un sistema basado en inteligencia artificial no solo contribuirá al sector eléctrico nacional como usuario principal, sino que también se posiciona como una herramienta valiosa en la toma de decisiones para expertos en el sector eléctrico.

En resumen, este trabajo no solo busca avanzar en la investigación y aplicación de inteligencia artificial en el diagnóstico de transformadores, sino que también abre nuevas perspectivas para la gestión eficiente de activos, la prevención de fallas y la optimización continua de la infraestructura eléctrica en pro del desarrollo sostenible y económico.

CAPÍTULO II. MARCO TEÓRICO

Resumen

En este capítulo se describen los conceptos básicos teóricos que se utilizan en este trabajo de investigación, incluyendo la descripción de las características de los transformadores de potencia; además de la descripción básica de las técnicas o herramientas de Inteligencia Artificial y Software que son utilizadas para el desarrollo y validación de los modelos.

2.1 Transformadores de potencia

Un transformador de potencia (Figura 2.1) es una máquina compuesta por varios subsistemas, cuya condición puede ser monitoreada y evaluada de forma independiente. (Cerón, Orduña, Aponte, & Romero, 2015)

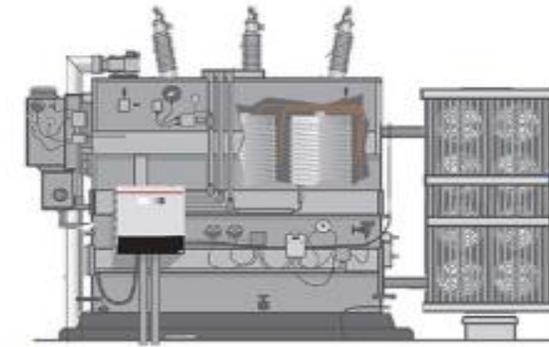


Figura 2.1 Transformador de potencia

A pesar de que estos subsistemas interactúan entre sí y pueden aportar a la falla del sistema general, se debe considerar que algunos subsistemas son de fácil mantenimiento con respecto a otros, a los que el acceso es limitado o incluso prohibitivo durante la normal operación de la máquina; por lo que tales subsistemas resultan ser de mayor criticidad en la determinación de la probabilidad de falla final de una unidad transformadora. Varios trabajos han identificado de forma exitosa los principales subsistemas de un transformador, pero uno de los más completos está desarrollado en IEEE C57.140 (2006), donde el transformador es dividido en 8 subsistemas principales y 25 componentes (Tabla 2.1). (Cerón, Orduña, Aponte, & Romero, 2015).

En la Tabla 2.1 se observa que, de los ocho subsistemas, dos de ellos componen la parte activa de la unidad, es decir, el núcleo y los devanados. Son justamente estos dos subsistemas los de más difícil mantenimiento ya que los mismos conforman la parte interna del equipo, están sumergidos en el aceite, no son reemplazables fácilmente y una intervención implica poner fuera de servicio la unidad, acción de elevado riesgo e indeseable por parte de los propietarios del activo puesto que facilita el ingreso de humedad, implica largos tiempo de fuera de servicio, puede producir pérdidas y vertimientos de aceite, entre otras consecuencias. (Cerón, Orduña, Aponte, & Romero, 2015).

Tabla 2.1. División del Transformador de Potencia (IEEE C57.140, 2006)

Subsistema	Componente	Relación de transformación	Resistencia aislamiento	Resistencia devanado	PF/capacitancia	DGA aceite	Visual	Monitoreo PD	Escaneo infrarrojo	Contenido de agua	Contenido de acidez	Tensión de ruptura
Pasatapa (Bushing)	Condensador				X						X	X
	Aceite				X	X	X			X		
	Escudo corona					X		X				
Devanado	Conductor/devanada	X	X	X	X	X		X				
	Aislamiento	X	X	X	X	X		X			X	X
Núcleo	Núcleo de acero					X						
	Aislamiento		X			X		X	X		X	X
Sistema conservador del aceite	Aceite		X		X	X			X	X		
	Conservador				X	X	X				X	X
	Tubería						X					
Refrigerador	Bomba					X	X					
	Aspas					X	X					
	Placas del radiador						X					
	Placas tubulares						X					
	Carcasa						X					
Cambiador de derivaciones desenergizado (DETC)	Contactos					X						
	Árbol de transmisión						X					
Cambiador de derivaciones de bajo carga (LTC)	Contactos					X						
	Árbol de transmisión						X					
	Circuito de control						X					
	Tablero aislador					X						
	Empaques					X						
Tanque	Empaques					X						
	Tubería					X						
	Estructura de acero					X	X		X			

La situación difiere con los otros subsistemas (pasatapas, sistema conservador de aceite, sistema de refrigeración, cambiadores de derivaciones y parte externa del tanque) ya que estos al estar ubicados en el exterior de la máquina, son susceptibles a mantenimientos más frecuentes y menos riesgosos y costosos, e incluso a la sustitución completa del subsistema (obviamente con la excepción del tanque). Luego, si se realiza un mantenimiento y monitoreo adecuado de los otros

seis subsistemas, se puede evaluar la probabilidad de falla del transformador bajo la premisa de que el fin de vida del transformador se alcanzará cuando fallen alguno de los dos subsistemas núcleo y devanado, los que además tienen como componente crítico al papel aislante. (Cerón, Orduña, Aponte, & Romero, 2015).

2.2 Índice de salud

El índice de salud se obtiene de la combinación de la información resultante de pruebas de diagnóstico que se realizan a un transformador de potencia, por lo tanto, es necesario contar con un eficiente sistema de adquisición y almacenamiento de los datos. Su objetivo, es proporcionar un único índice cuantitativo que representa su estado general de salud. El resultado de este índice varía desde cero (transformadores en óptimas condiciones) hasta uno para (transformadores deteriorados). (Cerón, Echeverry, Aponte, & Romero, 2015).

Algunas de las metodologías identificadas para calcular el índice de salud del transformador, se basan en establecer relaciones lineales entre los resultados de los diferentes ensayos realizados a este equipo. En estas metodologías se plantea que a cada resultado se le debe asignar un valor relativo y a cada prueba un peso específico de acuerdo con su importancia; luego se realiza por cada prueba el producto entre valor relativo y el peso específico y finalmente se deben sumar todos los productos obtenidos; el valor resultante de la sumatoria corresponde a la condición del transformador evaluado. De esta forma, se pueden comparar los valores obtenidos para cada transformador y clasificar su estado en función de la condición técnica. Sin embargo, esta técnica presenta dos desventajas: La primera es que los límites de aceptación entregados por la normatividad actual (CIGRE 445, 2011; IEC 60296, 2003; IEC 60422, 2013; IEC 60599, 2007; IEC 61198, 1993; IEEE C57.104, 2008; IEEE C57.106, 2006; IEEE C57.12.90, 2010; IEEE C57.140, 2006; IEEE C57.152, 2013) difieren entre sí y además contienen un margen de incertidumbre, por lo tanto, aparece una región difusa entre los límites. La segunda desventaja es que la asignación de pesos a las pruebas se basa en la experiencia, lo cual puede diferir entre un experto a otro. Esta desviación en los pesos hace que el índice de salud estimado difiera para una misma condición del transformador. (Cerón, Echeverry, Aponte, & Romero, 2015)

2.2.1 Descargas parciales acústicas (DP acústicas)

La detección acústica de DP se realiza mediante un equipo de prueba diseñado y fabricado por el IIE. El equipo de pruebas consiste en un conjunto de sensores acústicos que miden en la banda de los 150 KHz, un módulo concentrador y una computadora. Cada módulo concentrador maneja cuatro sensores acústicos y pueden ser interconectados tres o más módulos concentradores simultáneamente. Los módulos concentradores capturan las formas de onda de la envolvente de las señales acústicas detectadas. Un programa de control despliega la información de las formas de onda acústicas adquiridas, su frecuencia de repetición, sus tiempos de arribo y mediante un análisis posterior, la localización de las DP dentro del tanque. La figura 2.2 muestra la detección acústica de DP en un transformador. (Azcárraga Ramos, Liñan García, Nava Guzmán, & Ramírez Niño, 2014)



Figura 2.2. Detección de DP acústica en un Transformador

2.2.2 Gases disueltos en el aceite (DGA)

Las sobrecargas, DP y arqueos en el interior del transformador degradan químicamente el aislamiento aceite-papel, generando varios gases que se disuelven en el aceite según la energía asociada a la falla. De acuerdo con esto es posible diagnosticar un tipo particular de falla, midiendo la concentración de determinados gases. El análisis de tendencias en función del tiempo de estas concentraciones, en conjunto con la detección acústica de DP, el análisis fisicoquímico del aceite y la detección de furfurales hacen de la DGA una herramienta de diagnóstico normalizada, rentable y de bajo costo. Los métodos de análisis preferidos para DGA son criterios de concentración máxima, el triángulo de Duval, y métodos Dornenburg y Rogers. (Azcárraga Ramos, Liñan García, Nava Guzmán, & Ramírez Niño, 2014).

2.2.2.1 Dornenburg

En 1970, Dornenburg propone una técnica para diferenciar el origen térmico o eléctrico de las averías; esta se basa en las cuatro relaciones entre concentraciones de gases (Ecuación 1-4). (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

$$R_1 = \text{Razón 1} = \frac{CH_4}{H_2} \quad \text{Ecuación (1)}$$

$$R_2 = \text{Razón 2} = \frac{C_2H_2}{C_2H_4} \quad \text{Ecuación (2)}$$

$$R_3 = \text{Razón 3} = \frac{C_2H_2}{CH_4} \quad \text{Ecuación (3)}$$

$$R_4 = \text{Razón 4} = \frac{C_2H_6}{C_2H_2} \quad \text{Ecuación (4)}$$

Esta técnica se puede aplicar solo si existe una cantidad adecuada de gases disueltos en el aceite, lo que es una desventaja, dado que las relaciones que se obtienen de los análisis no se ajustan a la interpretación del origen de la avería (Tabla 2.2) (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014).

Tabla 2.2. Interpretación del origen de la avería de acuerdo con Dornenburg.

Avería originada por	R1	R2	R3	R4
Descomposición térmica	$<>1.0$	<0.75	<0.3	>0.4
Corona (PD de baja intensidad)	<0.1	No es significativa	<0.3	>0.4
Arco (PD de alta intensidad)	>0.1 y <1.0	>0.75	>0.3	<0.4

2.2.2.2 Rogers

En 1978, Rogers observa que la concentración de cada gas varía con la temperatura de la avería, e introduce una nueva relación entre las concentraciones de los gases (Etileno y Acetileno) que requieren una temperatura más elevada para generarse. Concluye que el etano y el metano no ayudan en la identificación de la avería, y, por ende, las elimina de las relaciones utilizadas para esta técnica (Ecuación 5-7) (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014). En la tabla 2.3 se muestra la tabla de interpretación del método de Rogers.

$$\text{Razón 1} = \frac{CH_4}{H_2} \quad \text{Ecuación (5)}$$

$$\text{Razón 2} = \frac{C_2H_2}{C_2H_4} \quad \text{Ecuación (6)}$$

$$\text{Razón 3} = \frac{C_2H_4}{C_2H_6} \quad \text{Ecuación (7)}$$

Tabla 3.3. Interpretación de fallas mediante método de Rogers

RANGO DE RELACIONES		R1 (CH ₄ /H ₂)	R2 (C ₂ H ₂ /C ₂ H ₄)	R3 (C ₂ H ₄ /C ₂ H ₆)
< 0.1		1	0	0
0.1-1.0		0	1	0
1.0-3.0		2	1	1
> 3.0		2	2	2
CASO	TIPO DE FALLA	CÓDIGO		
1	Descargas parciales de baja intensidad	1	1	0
2	Descargas parciales de alta intensidad	1	1	0
3	Descargas de baja energía, chispeo y arco	0	1-2	1-2
4	Descargas de alta energía arco	0	1	2
5	Falla térmica por temp. inferiores a 150° C	0	0	1
6	Falla térmica por temp. Entre 150°C y 300°C	2	0	0
7	Falla térmica por temp. Entre 300°C y 700°C	2	0	1
8	Falla térmica por temp. Superior a 700°C	2	0	2

2.2.2.3 Duval

En 1974, Duval utiliza tres de los seis gases empleados por Dornenburg para clasificar averías en transformadores, y concluye que el hidrógeno se difunde más rápido en el metal que los otros gases hidrocarburos, distorsionando el diagnóstico. (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

En el 2002, Duval interpreta los resultados del DGA en una forma gráfica y utiliza modelos de laboratorio para simular averías tomando como muestra ciento setenta y nueve casos reales (con el transformador en servicio), identificados por inspección visual, y diecinueve casos simulados en laboratorio. (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

Para la interpretación gráfica, Duval utiliza un triángulo equilátero compuesto por 3 vértices (%CH₄, %C₂H₄ y %C₂H₂). Esta técnica (Ecuación 8-10) propone un sistema de concentraciones en ppm (parte por millón). (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

$$\%C_2H_2 = 100 \frac{X}{X + Y + Z} \quad \text{Ecuación (8)}$$

$$\%CH_4 = 100 \frac{Y}{X + Y + Z} \quad \text{Ecuación (9)}$$

$$\%C_2H_4 = 100 \frac{Z}{X + Y + Z} \quad \text{Ecuación (10)}$$

Siendo: X Concentración en ppm de CH_4 , Y concentración en ppm de C_2H_4 ; Z concentración en ppm de C_2H_2 .

A manera de ejemplo, en la Figura 2.3 se ilustra la forma de detectar una avería en el transformador. Las concentraciones CH_4 , C_2H_4 y C_2H_2 son los ejes coordenados del triángulo equilátero, donde se ha tomado $X = X_p = 60$ ppm, $Y = Y_p = 30$ ppm y $Z = Z_p = 10$ ppm. Se traza para X_p una recta paralela a BC ; para Y_p , una recta paralela a CA , y para Z_p , una recta paralela a AB , siendo la intersección de estas rectas el punto de ubicación de la avería. (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

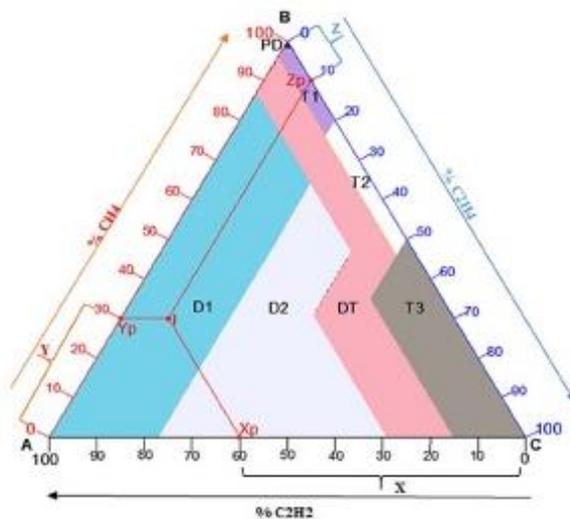


Figura 2.3. Ubicación de avería en el triángulo de Duval

2.2.2.4 Técnica de Gases Claves

La presencia de los gases combustibles depende de la temperatura presente en el aceite del transformador. Cuando se presenta una avería, la temperatura del transformador aumenta y se libera gran cantidad de gases combustibles en la cuba del transformador. La técnica de Gas Clave detecta las averías mediante la medición de los gases individuales en lugar de calcular las proporciones de gas. Esta técnica tuvo sus inicios en 1973 en los laboratorios de Doble, estableciéndose oficialmente en 1974. (Sarria Arias, Guerrero Bello, & Rivas Trujillo, 2014)

2.2.3 Rigidez Dieléctrica

Es la medida de la capacidad del aceite para soportar tensión eléctrica. Sirve para indicar la presencia de agentes tales como agua, suciedad, contaminación de fibras celulósicas o partículas en el aceite. Un valor de tensión de ruptura dieléctrica bajo puede indicar la presencia de una concentración significativa de uno o más de estos agentes (IEEE C57.140, 2006). Sin embargo, un alto valor de tensión de ruptura dieléctrica, necesariamente no indica la ausencia de contaminantes, pero puede ser indicio de concentraciones no tan grandes de estos agentes, por lo tanto, no se deben tomar decisiones a partir de los resultados de esta prueba sin el acompañamiento de otros ensayos. Por ejemplo, la humedad en combinación con el oxígeno y el calor pueden destruir el aislamiento sólido (papel) mucho antes de que la rigidez dieléctrica del aceite se vea comprometida (IEEE C57.106, 2006). (Cerón, Echeverry, Aponte, & Romero, 2015).

2.2.4 Contenido de Humedad

Se mide la cantidad de agua presente en el aceite a través de la titulación colométrica de Karl Fischer. Este método de prueba es conocido por su alto grado de sensibilidad con detecciones hasta de 10gg de agua. El contenido de humedad en un transformador afecta tanto al papel como al aceite, por ejemplo, un alto contenido de humedad puede reducir la resistencia mecánica de la celulosa, así como la rigidez dieléctrica del aceite mineral. (Cerón, Echeverry, Aponte, & Romero, 2015)

La condición del aceite del transformador con respecto al contenido de humedad se divide en tres tipos: bueno, aceptable y malo. Según la norma (IEEE C57.152, 2013), el límite de aceptación para aceites en servicio es máximo 25ppm y para aceites nuevos antes de energización es máximo 10ppm. De acuerdo con diferentes autores un valor aceptable de contenido de humedad es 20ppm. (Cerón, Echeverry, Aponte, & Romero, 2015)

2.2.5 Análisis de Contenido de Furanos

El deterioro de la estructura química del papel como consecuencia de una mayor temperatura de operación del transformador y un alto contenido de humedad disminuye el grado de polimerización (DP) lo cual es proporcional a un mayor contenido de compuestos furánicos disueltos en el aceite, especialmente del compuesto 2-furaldehído cuya proporción es más del 90% comparada con la de los demás furanos. El contenido de 2-furaldehído (2-FAL) se considera el factor más importante en la evaluación de la salud del transformador, ya que evalúa directamente la salud del aislamiento sólido (papel). (Cerón, Echeverry, Aponte, & Romero, 2015)

Algunos estudios se han realizado a partir de datos obtenidos de muestras de laboratorio y transformadores fuera de servicio, ya sea para reparación o investigación. Con esta información se desarrollaron modelos matemáticos que relacionan la concentración del compuesto furánico (2FAL) con el valor de DP. Uno de los modelos es mostrado en la Ecuación 11, que fue desarrollada con base en datos recolectados de transformadores con papel kraft. (Cerón, Echeverry, Aponte, & Romero, 2015)

$$DP = \frac{1.5 - \log_{10}(2FAL_{ppm})}{0.0035} [ppm] \quad \text{Ecuación (11)}$$

Donde DP= grado de polimerización [ppm] y 2FAL= contenido de 2-furaldehído [ppm]

Según la norma IEEE C57.140 (2006), un aceite en buena condición debe tener como máximo un contenido de 2-FAL de 0.1ppm, al aplicar la Ecuación (8) esto equivale a un grado de

polimerización (DP) mayor de 700; lo cual representa un aislamiento sólido (papel) con una resistencia mecánica de la celulosa cercana al 100% de su valor inicial, es decir, el valor medido para papel nuevo (Martínez, 2010). Un aceite con una condición aceptable tendrá un DP entre 700 y 400 (papel con retención de resistencia mecánica cercana al 70%), un aceite con una condición regular tendrá un DP entre 560 y 290 (papel con retención de resistencia mecánica cercana al 50%), un aceite con una condición mala tendrá un DP entre 200 y 320 (papel con retención de resistencia mecánica cercana al 20%) y un aceite con una condición muy mala tendrá un DP menor a 200 (papel con retención de resistencia mecánica cercana a 0%). Los valores equivalentes de 2-FAL en relación con los rangos de DP se muestran en la Tabla 2.4. (Cerón, Echeverry, Aponte, & Romero, 2015)

Tabla 2.4. Relación entre contenido de 2-FAL y DP

Condición	Muy malo	Malo	Regular	Aceptable	Bueno
DP [ppm]	<200	200-320	290 – 560	400 – 700	>700
2-FAL [ppm]	>6.46	6.46 – 2.45	3.13 – 0.35	1.29 – 0.12	<0.12

2.2.6 Diagnóstico de devanados

El circuito equivalente en alta frecuencia del transformador puede ser considerado como una red RLC (Figura 2.4). La inductancia se asocia a los devanados y al comportamiento del núcleo magnético. La resistencia representa las pérdidas en el cobre que dependen de la longitud y el diámetro de los devanados, y también puede ser correlacionada con los contactos del cambiador de derivaciones y con las pérdidas dieléctricas. La capacitancia a tierra o entre devanados tiene que ver con la geometría del arreglo interno del transformador y está definida por los materiales, formas y distancias dieléctricas entre devanados a tierra y entre los devanados de alta y baja tensión. Si la frecuencia aplicada es lo suficientemente alta es posible identificar cambios geométricos en bobinas y entre vueltas, revelando la existencia de fallas incipientes. (Azcárraga Ramos, Liñan García, Nava Guzmán, & Ramírez Niño, 2014)

En la técnica de FRA se aplican voltajes alternos de frecuencia variable con pasos discretos a diferentes arreglos de devanados en el transformador y se realiza la medición de impedancia, admitancia, ángulo de fase o cualquier otro parámetro adecuado, en función de la frecuencia aplicada. La técnica FRA aporta sus resultados de forma gráfica principalmente. Analizando las respuestas en frecuencia de los devanados es posible determinar qué tipo de problema afecta a la integridad electromecánica del transformador. (Azcárraga Ramos, Liñan García, Nava Guzmán, & Ramírez Niño, 2014)

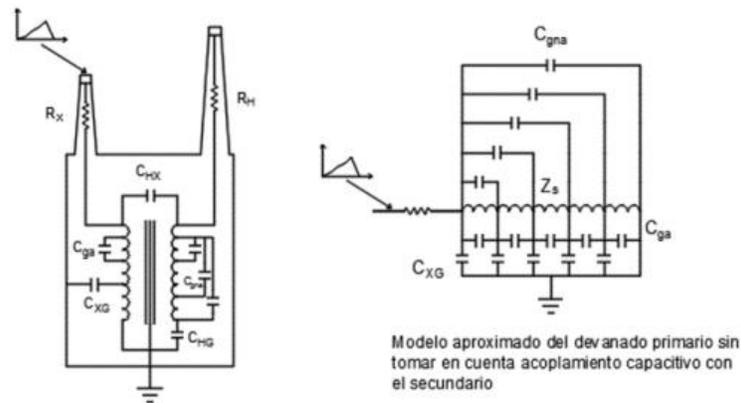


Figura 2.4. Circuito equivalente en alta frecuencia de un transformador

2.3 Monitoreo para la detección y prevención de fallas de transformadores de potencia

Durante su operación, los transformadores de potencia están sujetos a esfuerzos térmicos, eléctricos y mecánicos, los cuales degradan el sistema de aislamiento aceite/papel. Las causas principales de la degradación del sistema aislante aceite/papel son temperatura excesiva, la presencia de oxígeno y humedad combinados con los esfuerzos eléctricos, los cuales actúan como aceleradores del proceso de degradación. Las causas secundarias que aceleran el envejecimiento del sistema aislante son los esfuerzos mecánicos (originados por la excitación del núcleo y cortos circuitos) y los productos de descomposición del aceite (ácidos y lodos). (López Tagle, Equihua Tapia, Linán García, Pascacio de los Santos, & Nuñez Dominguez, 2000)

El proceso de degradación del sistema aislante aceite/papel evoluciona gradualmente hasta presentarse la falla en el transformador de potencia, que en ocasiones puede ser catastrófica. La degradación de transformadores de potencia presenta ciertos síntomas que indican la presencia de problemas. La detección oportuna de una degradación incipiente puede ser la diferencia entre someter el transformador a una reparación mayor o sustituir a una pieza dañada. La detección de degradación incipiente se efectúa mediante el monitoreo de ciertos parámetros clave que sirven para diagnosticar la condición del sistema aceite/papel. El monitoreo de equipo de potencia puede realizarse en línea o fuera de línea. La tendencia mundial en el ámbito de detección de degradación incipiente en equipo de potencia es la aplicación de técnicas de monitoreo y diagnóstico en línea. (López Tagle, Equihua Tapia, Linán García, Pascacio de los Santos, & Nuñez Dominguez, 2000)

2.3.1 Monitoreo en Línea

Un sistema de monitoreo en línea consta principalmente de sensores, sistemas de adquisición de datos, manejo de información y métodos para evaluar la condición de los transformadores de potencia. El monitoreo es la base para realizar el diagnóstico de la condición de los transformadores de potencia. El monitoreo en línea permite diagnosticar problemas incipientes en transformadores de potencia de manera efectiva. (López Tagle, Equihua Tapia, Linán García, Pascacio de los Santos, & Nuñez Dominguez, 2000)

Los parámetros importantes por monitorear continuamente en un transformador de potencia son la carga, las condiciones de operación y la generación de gases disueltos en el aceite aislante.

De acuerdo con (López Tagle, Equihua Tapia, Linán García, Pascacio de los Santos, & Nuñez Dominguez, 2000) el monitoreo de la carga y de las condiciones de operación consiste básicamente en la medición de las tensiones y corrientes a la frecuencia de operación y en la medición de temperaturas. Las tensiones pueden medirse usando los dispositivos de potencial, y la corriente de carga se mide con los transformadores de corriente instalados en las boquillas o mediante dispositivos ubicados en otros puntos de la subestación.

El estado que guardan los pasos de enfriamiento se mide a través de dispositivos con salidas digitales que indican el estado en que se encuentran (encendido o apagado). Esto permite

correlacionar la actividad de los pasos de enfriamiento con la carga transportada en función del tiempo. La medición de corrientes de bombas permite detectar problemas en estos accesorios. Cualquier incremento de la corriente demandada por ellos es un indicio de la existencia de problemas. (López Tagle, Equihua Tapia, Linán García, Pascacio de los Santos, & Nuñez Dominguez, 2000)

2.4 Inteligencia Artificial

De acuerdo con (Sossa Azuela) la Inteligencia Artificial (IA) es, hoy en día, una disciplina con un nivel de madurez muy alto. No se puede poner en tela de duda su potencial, aplicabilidad e impacto en nuestra sociedad. Prácticamente todos en nuestras casas, oficinas de trabajo, negocios, escuelas, etcétera, sin darnos cuenta, utilizamos algún dispositivo “inteligente” que nos hace la vida más fácil, extiende nuestras capacidades, nos libera de actividades engorrosas, nos da seguridad. En resumen, nos brinda una mejor calidad de vida. Los avances logrados hasta el momento, en esta área del conocimiento, permiten que un gran número de personas, hogares y empresas, cuenten con sistemas capaces de auxiliarlos en la toma de decisiones atinadas, y en situaciones que antes hubiera sido muy difícil, incluso imposible de lograr.

Según (Haugeland, 1985), la inteligencia artificial es “la interesante tarea de lograr que las computadoras piensen... máquinas con mente, en su amplio sentido literal). Otra definición, según (Luger y Stubblefield, 1993) la inteligencia artificial es “la rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente”. Así, la IA puede entenderse como la parte de la ciencia que se ocupa del diseño de sistemas de computación inteligente, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento y la resolución de problemas.

Las tecnologías basadas en la IA ya están siendo utilizadas para ayudar a los humanos a beneficiarse de mejoras significativas y disfrutar de una mayor eficiencia en casi todos los ámbitos de la vida. Pero el gran crecimiento de la IA también nos obliga a estar atentos para prevenir y analizar las posibles desventajas directas o indirectas que pueda generar la proliferación de la IA.

La IA se puede aplicar en casi todas las situaciones (Rouhianen, 2018). Éstas son sólo algunas de las aplicaciones técnicas de la IA que están creciendo rápidamente en la actualidad:

- Reconocimiento de imágenes estáticas, clasificación y etiquetado: estas herramientas son útiles para una amplia gama de industrias.
- Mejoras del desempeño de la estrategia algorítmica comercial: ya ha sido implementada de diversas maneras en el sector financiero.
- Procesamiento eficiente y escalable de datos de pacientes: esto ayudará a que la atención médica sea más efectiva y eficiente.
- Mantenimiento predictivo: otra herramienta ampliamente aplicable en diferentes sectores industriales.
- Detección y clasificación de objetos: puede verse en la industria de vehículos autónomos, aunque también tiene potencial para muchos otros campos.
- Distribución de contenido en las redes sociales: se trata principalmente de una herramienta de marketing utilizada en las redes sociales, pero también puede usarse para crear conciencia entre las organizaciones sin ánimo de lucro o para difundir información rápidamente como servicio público.
- Protección contra amenazas de seguridad cibernética: es una herramienta importante para los bancos y los sistemas que envían y reciben pagos en línea.

2.4.1 Lógica Difusa

De acuerdo con (Delgado, Paz, & Tupia, 2020) La lógica difusa, perteneciente al área de la Inteligencia Artificial, fue investigada inicialmente por Lofti Zadeh en 1965 en su artículo “Fuzzy Sets”. Conceptualizada en un inicio como “conjunto difuso”, este artículo matemático presenta su definición como “una clase con diversos grados de pertenencia”.

Este marco matemático permite realizar un tratamiento sistemático a la imprecisión e incertidumbre producidas, principalmente, por variables con un alto índice de ambigüedad. Según los conjuntos clásicos, el clima o es frío o es caliente. Sin embargo, con lógica difusa el clima puede ser caliente y frío a la vez a un cierto grado de pertenencia en ambos conjuntos. Además, Zadeh menciona que “este marco de trabajo provee un medio natural para lidiar con problemas en

los que la fuente de imprecisión es la ausencia de criterios definidos claramente de pertenencia de clases en vez de la presencia de variables aleatorias”

Otros autores como (Ponce Cruz, 2010) definen que la lógica difusa es un conjunto de principios matemáticos basados en grados de membresía o pertenencia, cuya función es modelar información. Este modelado se hace con base en reglas lingüísticas que aproximan una función mediante la relación de entradas y salidas del sistema (composición). Una de las ventajas del sistema de control basado en lógica difusa es que no es necesario conocer un modelo matemático del sistema real, pues se puede ver como una caja negra a la cual se le proporcionan entradas, y a través del sistema esta planta generará la salida deseada

De acuerdo con (Ponce Cruz, 2010), la anatomía básica de un controlador difuso consta de lo siguiente:

1. Reglas. Estas son reglas que dictan la acción de control que se va a tomar. Éstas se derivan de un experto. Dichas reglas tienen la estructura de relaciones. La lógica difusa se basa en relaciones, las cuales se determinan por medio de cálculo de reglas “SI-ENTONCES” (con las cuales se puede modelar aspectos cualitativos del conocimiento humano, así como los procesos de razonamiento sin la necesidad.
2. Difusificador. Es el nexo entre las entradas reales y difusas. Todas las entradas necesitan ser mapeadas a una forma en que las reglas puedan utilizarlas.
3. Desdifusificador. Toma un valor difuso de las reglas y genera una salida real.

La lógica difusa consta de tres etapas para obtener el resultado deseado. La primera etapa se basa en un proceso donde las variables tienen un grado de incertidumbre metalingüístico. Por lo tanto, el rango de valores (universo de discurso) de cada variable puede clasificarse por conjuntos difusos. Cuando los sensores miden las variables, los valores pasan a un proceso de fusificación que consiste en pasar dichos valores a un rango de pertenencia entre cero (0) y uno (1). Se busca determinar en qué grado el valor que se está adquiriendo pertenece a un conjunto difuso. Los conjuntos difusos son caracterizados mediante funciones de membresía, las cuales están sintonizadas al punto de operación adecuado para el funcionamiento del sistema. En la segunda etapa se proponen reglas lingüísticas (inferencia) que servirán de guía para que el sistema se comporte de manera más adecuada, idónea o deseada según el modelo de referencia o los objetivos

del usuario. El grado de pertenencia de cada una de las variables se evalúa en un conjunto de reglas de inferencia. Dichas reglas de inferencia fueron determinadas con ayuda de un experto. El conjunto de reglas de inferencia determina una consecuencia, es decir, asigna un grado de pertenencia a un conjunto difuso que caracteriza a las salidas. Una vez obtenidas las consecuencias, la tercera etapa es un proceso para determinar los valores óptimos de salida, conocido como defusificación, y que consiste en pasar el grado de pertenencia, proveniente de la consecuencia de la regla de inferencia, a un valor nítido o real. Para hacer eso, previamente se sintonizaron funciones de membresía de cada una de las salidas con el fin de obtener un valor cuantificable. Al final, el control entregará valores nítidos o reales, consecuencia de las reglas lingüísticas previamente estructuradas, con lo cual este sistema interpretará las órdenes y realizará las acciones pertinentes. (Ponce, 2010) En la Figura 2.5 se muestra el diagrama esquemático del controlador difuso.

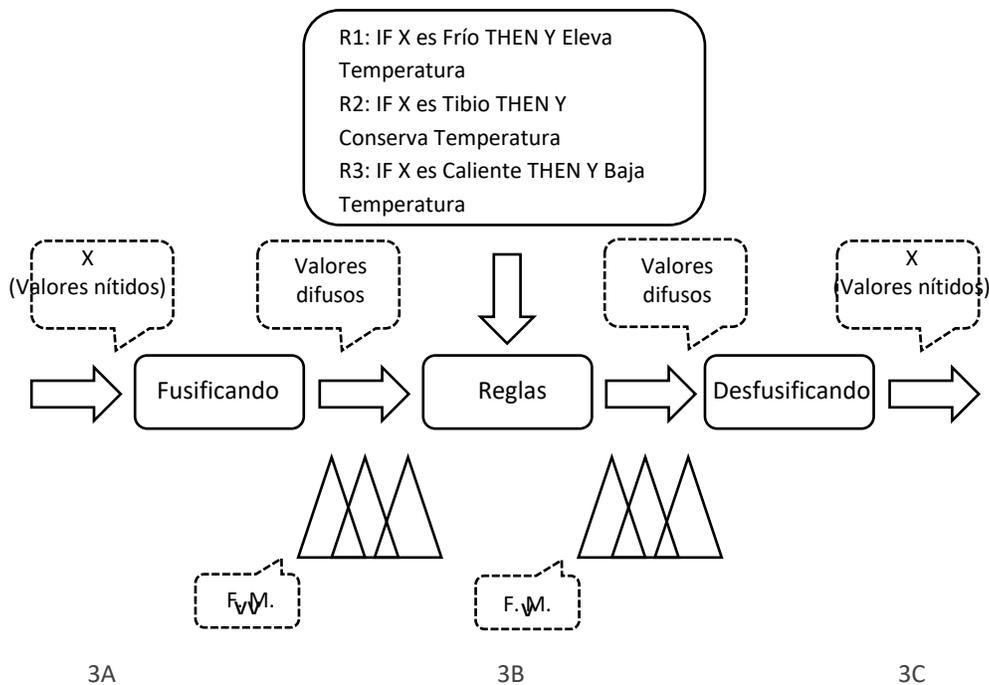


Figura 2.5. Esquematización de la Lógica Difusa

2.5 Aprendizaje automático

Según (Bobadilla , 2020) el aprendizaje automático o Machine Learning (ML) es un subconjunto de la IA que hace que los ordenadores “aprendan” a partir de los datos. en vez de programar, paso a paso, cada solución específica para cada necesidad planteada, tal y como se realiza en el enfoque de la programación convencional, el área de machine learning está dedicada al desarrollo de algoritmos genéricos que pueden extraer patrones de diferentes tipos de datos. De esta manera, un programa de machine learning destinado, por ejemplo, a clasificar números escritos a mano, no va a diferir sustancialmente de un programa destinado a la clasificación de las imágenes de sales de tráfico: ambos se basarán en la existencia de algún tipo de algoritmo de machine learning que clasifique datos etiquetados.

Con el objetivo de poder abordar cualquier tarea específica, el ingeniero de datos debe conocer algunos conceptos importantes de machine learning, así como diferentes opciones existentes, las medidas de calidad más utilizadas, etc. los conocimientos básicos incluyen la identificación de las tareas, empezando por la clasificación de los problemas de machine learning (Figura 2.8) en alguno de los siguientes tipos:

- Aprendizaje supervisado
 - Regresión
 - Clasificación
- Aprendizaje no supervisado
 - Clustering (agrupamiento)
 - Reducción de dimensiones
- Aprendizaje semi-supervisado
- Aprendizaje por refuerzo

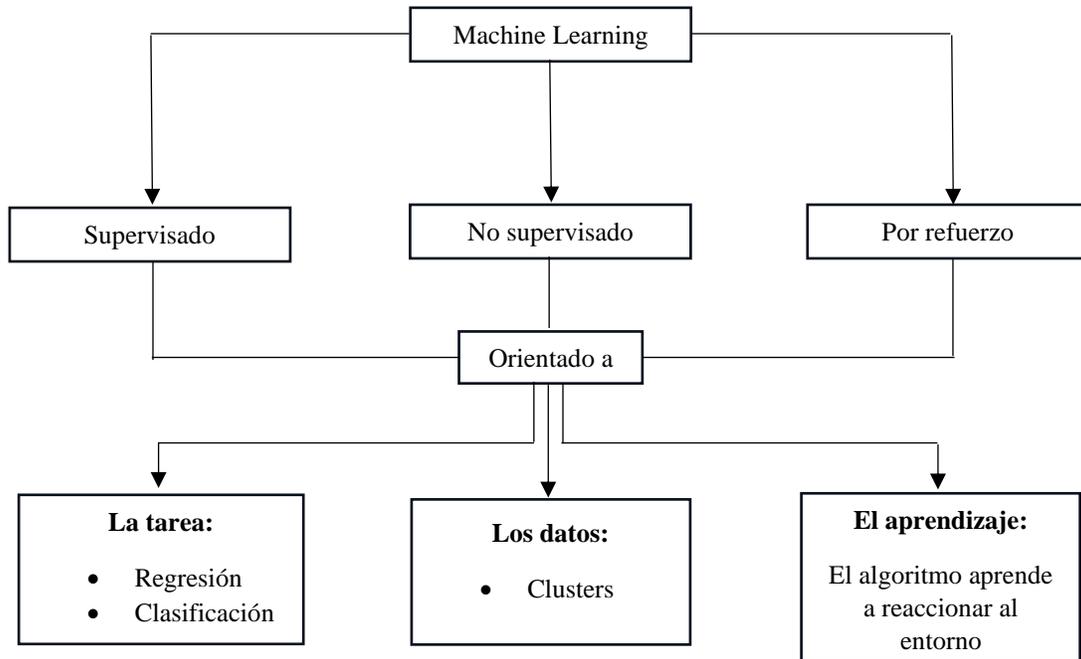


Figura 2.8. Clasificación de Machine Learning

2.6 Aprendizaje automático supervisado

De acuerdo con (Beunza Buini, Puertas Sanz, & Condés Moreno, 2019) el aprendizaje supervisado se refiere a un tipo de modelos de machine learning que se entrenan con un conjunto de ejemplos en los que los resultados de salida (outcome, eventos o labels) son conocidos. Los modelos aprenden de esos resultados conocidos y realizan ajustes en sus parámetros interiores para adaptarse a los datos de entrada. Una vez el modelo es entrenado y los parámetros internos son coherentes con los datos de entrada y los resultados de la batería de datos de entrenamiento, el modelo podrá realizar predicciones adecuadas ante nuevos datos no procesados previamente y de los que desconozcamos el outcome, evento o label (Figura 2.9).

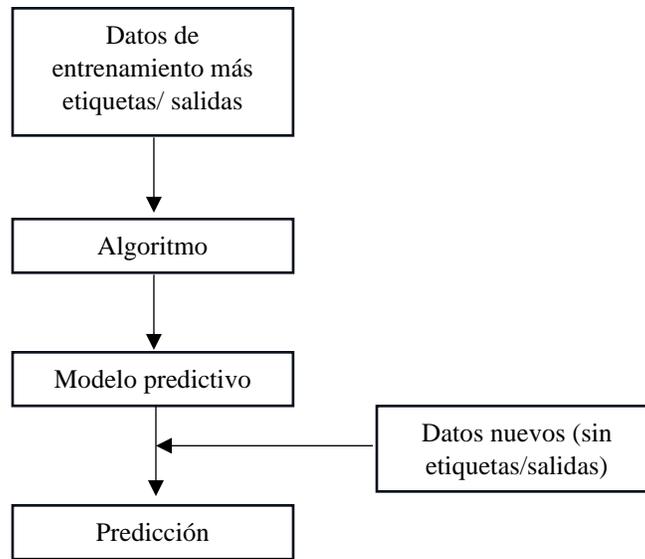


Figura 2.9. Descripción general de un algoritmo de ML supervisado

El aprendizaje supervisado en machine learning se aplica cuando cada dato o conjunto de datos de entrada (muestra) tiene asociada una etiqueta. Por ejemplo: un conjunto de imágenes en las que cada una de ellas contiene algún tipo de metadato (habitualmente un etiqueta o conjunto de etiquetas): ((pict0001, bmp, “perro”), (pict0002.bmp, “pájaro”). Partiendo de este conjunto de datos se pueden usar diferentes algoritmos de clasificación de machine learning con el objetivo de “entrenar” un modelo y poder, al acabar el entrenamiento, predecir la etiqueta correspondiente a una nueva imagen (no incluida en el conjunto de datos original); este es un problema de clasificación. (Bobadilla , 2020)

2.6.1 Aprendizaje Automático Supervisado de Clasificación

Este tipo de algoritmos parten de datos “etiquetados” (labels), donde la variable dependiente (p. ej. Mortalidad) es conocida y es categórica.

Una vez entrenado el modelo predictivo con los datos de entrenamiento etiquetados según el diagnóstico, se puede aplicar dicho modelo predictivo a nuevos datos en los que no se conoce su etiqueta, de manera que el modelo permite realizar una predicción, en este caso, sobre el diagnóstico de neumonía en una radiografía de tórax. Otro ejemplo sencillo sería un algoritmo de

predicción de mortalidad de pacientes ingresados en UCI (unidad de cuidados intensivos), donde la variable “fallecido” (sí/no) clasifica a los pacientes entre los que han fallecido y los que no. De este modo, al entrenar el algoritmo, disponemos de la información de a qué grupo pertenece cada sujeto. (Beunza Buini, Puertas Sanz, & Condés Moreno, 2019).

2.6.2 Aprendizaje Automático Supervisado de Regresión (SMLR)

El Aprendizaje Automático Supervisado de Regresión (SMLR) tiene el objetivo principal de predecir y pronosticar valores futuros de datos de variables dependientes basados en las variables independientes que son características de equipo/componente del conjunto de datos pre-observados. Por lo tanto, se aplica para predecir el valor de la variable de proceso, Vida útil restante (RUL), Estado de salud (SoH) y otros parámetros utilizados en el campo de mantenimiento. La principal ventaja de este enfoque es predecir los valores de los parámetros automáticamente sobre la base de los datos históricos actuales. Por lo tanto, los modelos SMLR permiten ahorrar la enorme cantidad de tiempo dedicado a esas actividades, así como para vincular los datos del sensor a los modelos PHM para predecir el equipo y componente RUL y SoH. Los modelos SMLR más comunes son los siguientes:

- Linear Regresión;
- Ridge and Lasso Regresión;
- Stepwise Linear Regresión;
- Gaussian Regresión;
- Decision Tree Regresión;
- Support Vector Machine Regresión,
- Neural Network Regresión (NNR).

Las etapas generales del proceso del SMLR se describen en la figura 2.10.

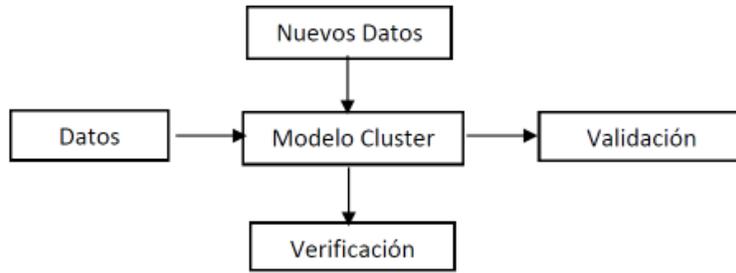


Figura 2.11. Pasos generales del modelo Aprendizaje Automático No Supervisado

2.7 Aprendizaje no Supervisado

El Aprendizaje Automático No Supervisado tiene como objetivo definir un patrón en el conjunto de datos sin conocimiento previo de las características de los datos. Por lo tanto, la primera comprensión de su conjunto de datos puede comenzar aplicando los métodos de Aprendizaje Automático No Supervisado para entender cómo se puede organizar su conjunto de datos y si hay un patrón de tal conjunto de datos basado en sus variables independientes. (Calixto, 2016)

Los conceptos detrás del Aprendizaje Automático No Supervisado es agrupar un conjunto de datos sin conocer la clasificación previa o cualquier información sobre los datos. Para agrupar los datos, el Aprendizaje Automático No Supervisado modela el conjunto de datos e intenta organizarlo en un clúster. El paso posterior verifica el resultado y, finalmente, Si el resultado es satisfactorio, el nuevo conjunto de datos se puede utilizar el modelo definido sobre la base del conjunto de datos anterior y luego el modelo se valida. Los pasos generales del proceso de Aprendizaje Automático No Supervisado se describen en la Figura 2.11.

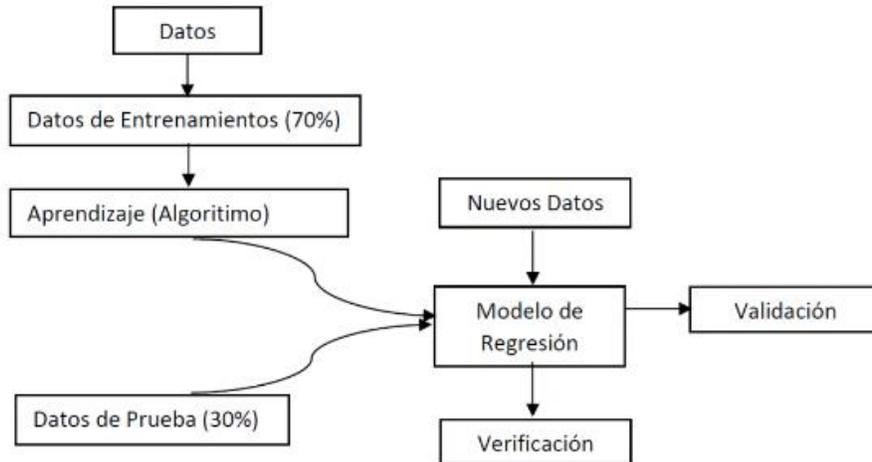


Figura 2.10. Pasos generales del modelo Aprendizaje Automático Supervisado de Regresión

Considerando la ingeniería del mantenimiento, el tipo de datos relacionados con el equipo abarca las características físicas, así como el rendimiento, el costo de las operaciones, el costo del mantenimiento preventivo, el mantenimiento correctivo y el costo de las piezas de repuesto. Por lo tanto, al definir algunas de estas variables, es posible agrupar equipos con características similares. De hecho, la organización de datos basada en sus características comunes es el concepto detrás de los datos agrupados que es el resultado de tales métodos de Aprendizaje Automático No Supervisado como muestra la figura 2.11

Los modelos de Aprendizaje Automático No Supervisados más comunes utilizados para la agrupación de los datos son:

- Principal Component Analysis
- Multidimensional Scaling
- K-Means
- Gaussian Mixture
- Hierarchical Clustering
- Neural Network Self-Organized Map

2.8 Algoritmos de aprendizaje supervisado

Como se ha mencionado anteriormente, los algoritmos de aprendizaje supervisado son un tipo de algoritmo de aprendizaje automático que se utilizan para entrenar modelos predictivos a partir de datos etiquetados. Estos algoritmos se denominan “supervisados” porque se les proporciona un conjunto de datos de entrenamiento que incluye ejemplos con etiquetas o resultados conocidos, lo que permite al algoritmo aprender a hacer predicciones o bien clasificaciones basadas en esos ejemplos.

El proceso del aprendizaje supervisado se puede resumir en 4 pasos:

1. Conjunto de datos de entrenamiento: se proporciona un conjunto de datos para su entrenamiento, cada ejemplo contiene características como lo son las variables de entrada, así como su etiqueta propia o bien su valor o variable de salida.
2. Entrenamiento del modelo: el algoritmo toma los datos de entrenamiento y crea un modelo matemático que relacione las características de entrada con las etiquetas o variables de salida, con el objetivo de encontrar un modelo que sea capaz de hacer predicciones precisas.
3. Evaluación del modelo: una vez entrenado el modelo, se prueba con un conjunto de datos no antes visto durante el entrenamiento, de este modo se mide la capacidad del modelo para generalizar o clasificar a nuevos datos y hacer predicciones precisas.
4. Predicción: una vez realizado el entrenamiento y la evaluación el modelo se puede utilizar para hacer predicciones de nuevos conjuntos de datos no etiquetados o con la variable de salida. Donde, el modelo toma las características de entrada y genera las predicciones o clasificaciones de acuerdo con lo aprendido durante el entrenamiento.

2.8.1 Máquinas de soporte vectorial (Support Vector Machines o SVM)

Las máquinas de soporte de vectores (SVM) son un conjunto de métodos de aprendizaje supervisado que se utilizan para la clasificación, la regresión y la detección de valores atípicos. (scikitlearn, 2019) SVM son especialmente eficaces en la clasificación de datos en conjuntos de alta dimensionalidad.

Las ventajas de las máquinas de vectores de soporte son:

- Eficaz en espacios de grandes dimensiones.
- Sigue siendo eficaz en los casos en los que el número de dimensiones es mayor que el número de muestras.
- Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), por lo que también es eficiente en memoria.

Las desventajas de las máquinas de vectores de soporte incluyen:

- Si la cantidad de características es mucho mayor que la cantidad de muestras, evitar el ajuste excesivo al elegir funciones del kernel y el término de regularización es crucial.
- Las SVM no proporcionan estimaciones de probabilidad directamente, sino que se calculan mediante una costosa validación cruzada quíntuple.

Las SVM se pueden utilizar para problemas de clasificación binaria, pero también se pueden extender para problemas de clasificación multiclase utilizando enfoques como “one vs rest” (uno contra el resto) o “one vs one” (uno contra uno). (scikitlearn, 2019)

Las SVM son poderosas y versátiles, pero la elección del kernel y la configuración de los hiperparámetros pueden afectar significativamente su rendimiento. Es importante ajustar adecuadamente una SVM y comprender las características de los datos y las necesidades del problema antes de aplicarla. También es útil para problemas en los que el número de características es alto en comparación con el número de ejemplos de entrenamiento. (scikitlearn, 2019)

2.8.2 Decision Trees

Los árboles de decisión (DT) son un método de aprendizaje supervisado no paramétrico que se utiliza para clasificación y regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo aprendiendo reglas de decisión simples inferidas de las características de los datos. Un árbol puede verse como una aproximación constante por partes (scikitlearn, 2019).

En la Figura 2.12 se puede observar un ejemplo, los árboles de decisión aprenden de los datos a aproximarse a una curva sinusoidal con un conjunto de reglas de decisión si-entonces-si no. Cuanto

más profundo sea el árbol, más complejas serán las reglas de decisión y más ajustado será el modelo.

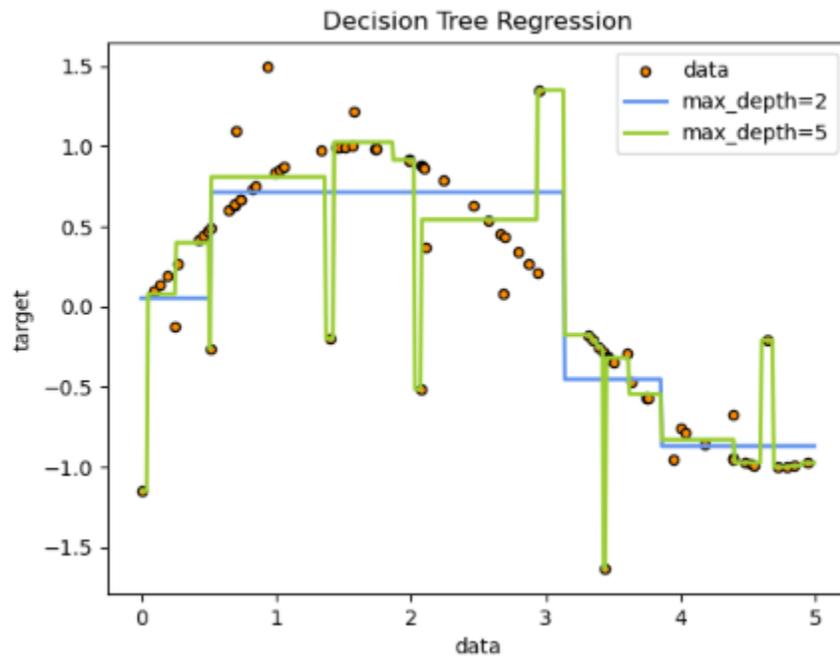


Figura 2.12. Regresión de un árbol de decisión

Algunas de las ventajas de los árboles de decisión son:

- Sencillo de entender e interpretar. Se pueden visualizar árboles
- Requiere poca preparación de datos. Otras técnicas a menudo requieren la normalización de datos, es necesario crear variables ficticias y eliminar valores en blanco. Algunas combinaciones de árboles y algoritmos admiten valores faltantes.
- Capaz de manejar datos tanto numéricos como categóricos.
- Capaz de manejar problemas de múltiples salidas
- Utiliza un modelo de caja blanca. Si una situación dada es observable en un modelo, la explicación de la condición se explica fácilmente mediante la lógica booleana. Por el contrario, en un modelo de caja negra (por ejemplo, en una red neuronal artificial), los resultados pueden ser más difíciles de interpretar.

- Es posible validar un modelo mediante pruebas estadísticas. Eso permite dar cuenta de la confiabilidad del modelo.
- Funciona bien incluso si sus supuestos son de alguna manera violados por el modelo real a partir del cual se generaron los datos.

Las ventajas de los árboles de decisión incluyen:

- Se pueden crear árboles demasiado complejos que no generalizan bien los datos. a esto se le llama sobreajuste. Para evitar este problema son necesarios mecanismos como la poda, establecer el número mínimo de muestras requeridas en un nodo de hoja o establecer la profundidad máxima del árbol.
- Los árboles de decisión pueden ser inestables porque pequeñas variaciones en los datos pueden generar un árbol completamente diferente.
- Se crean árboles sesgados si algunas clases dominan. Por lo tanto, se recomienda equilibrar el conjunto de datos antes de ajustarlo al árbol de decisión.

2.8.3 Random Forest

El random forest es un algoritmo de machine learning de uso común registrado por Leo Breiman y Adele Cutler, que combina la salida de múltiples árboles de decisión para alcanzar un solo resultado. Su facilidad de uso y flexibilidad han impulsado su adopción, ya que maneja problemas de clasificación y regresión. Además, mientras que los árboles de decisión consideran todas las posibles divisiones de características, los bosques aleatorios solo seleccionan un subconjunto de esas características. (IBM, 2019)

El bosque aleatorio (Random Forest) es uno de los algoritmos de aprendizaje automático más utilizados. Los árboles de decisión utilizados para construir el bosque aleatorio pueden tener una precisión de clasificación baja o correlaciones altas, lo que afecta el rendimiento integral del bosque aleatorio. (Zhigang , y otros, 2014)

Los algoritmos de random forest tienen tres hiperparámetros principales, que deben configurarse antes del entrenamiento (IBM, 2019):

1. Tamaño del nodo
2. Cantidad de árboles
3. Cantidad de características muestreadas

Dependiendo del tipo de problema, la determinación de la predicción variará. Para una tarea de regresión, se promediarán los árboles de decisión individuales, y para una tarea de clasificación, un voto mayoritario, es decir, la variable categórica más frecuente, arrojará la clase predicha. (IBM, 2019)

Ventajas del Random Forest:

- Los árboles de decisión corren el riesgo de sobreajustarse, ya que tienden a ajustar todas las muestras dentro de los datos de entrenamiento. Sin embargo, cuando hay una gran cantidad de árboles de decisión en un random forest, el clasificador no se ajustará demasiado al modelo, ya que el promedio de árboles no correlacionados reduce la varianza general y el error de predicción.
- Facilita la evaluación de la importancia o contribución de las variables al modelo.
- El agrupamiento de características también convierte al clasificador de random forest en una herramienta eficaz para estimar los valores perdidos, ya que mantiene la precisión cuando falta una parte de los datos.

Desventajas:

- Pueden manejar grandes conjuntos de datos y proporcionar predicciones más precisas, sin embargo, pueden ser lentos para procesar los datos, ya que están computando datos para cada árbol de decisión individual.
- Debido a que procesan conjuntos de datos más grandes, requerirán más recursos para almacenar esos datos.
- La predicción de un único árbol de decisiones es más fácil de interpretar en comparación con un bosque de ellos.

2.8.4 Gradient Boosting

Un modelo Gradient Boosting está formado por un conjunto de árboles de decisión individuales, entrenados de forma secuencial, de forma que cada nuevo árbol trata de mejorar los errores de los árboles anteriores. La predicción de una nueva observación se obtiene combinando las predicciones de todos los árboles individuales que forman el modelo (Rodrigo, 2020).

Ventajas:

- Son capaces de seleccionar predictores más relevantes de forma automática.
- Los árboles pueden, en teoría, manejar tanto predictores numéricos como categóricos sin tener que crear variables dummy.
- Al ser métodos no paramétricos, no requieren que los datos sigan una distribución específica.
- Por lo general, necesitan menos limpieza y preprocesamiento de datos en comparación con otros métodos de aprendizaje estadístico. Por ejemplo, no requieren estandarización.
- Son menos susceptibles a ser influenciados por valores atípicos.
- Son muy útiles en la exploración de datos, permiten identificar de forma rápido y eficiente las variables (predictores) más importantes.
- Son adecuados para conjuntos de datos con un gran número de observaciones.

Desventajas:

- La combinación de múltiples árboles reduce la capacidad de interpretación en comparación con modelos basados en un solo árbol.
- Al tratar con predictores continuos, se puede perder parte de la información al categorizarlos durante la división de los nodos.
- No son capaces de extrapolar fuera del rango observado en los datos de entrenamiento.

2.8.5 XGboost

De acuerdo con (SageMaker, 2023) XGBoost (potenciación del gradiente eXtreme) es una implementación de código abierto popular y eficiente del algoritmo de árboles aumentados de gradientes. La potenciación de gradientes es un algoritmo de aprendizaje supervisado que intenta predecir de forma apropiada una variable de destino mediante la combinación de un conjunto de estimaciones a partir de un conjunto de modelos más simples y débiles. El algoritmo XGBoost funciona bien en competencias de aprendizaje automático debido a su manejo robusto de una variedad de tipos de datos, relaciones, distribuciones y la variedad de hiperparámetros que puede ajustar. Se puede usar XGBoost para problemas de regresión, de clasificación (binaria y multiclase) y de ranking.

2.8.6 Naive Bayes

Naive Bayes es un algoritmo de aprendizaje automático supervisado que se utiliza principalmente en problemas de clasificación y minería de texto. El algoritmo se basa en el teorema de Bayes y en una suposición “ingenua” o “naive” acerca de la independencia condicional de las características.

Naive Bayes (NB) es uno de los algoritmos de aprendizaje supervisado más eficientes y efectivos. Simplifica considerablemente el aprendizaje mediante el supuesto de independencia de los atributos y, no obstante, compite en la práctica con clasificadores más sofisticados. Ha sido considerado además como uno de los 10 algoritmos de minería de datos más influyentes por varias razones. Es fácil de construir y no necesita de esquemas iterativos de estimación de parámetros. Ello implica que puede ser aplicado a grandes bases de datos. Es también fácil de interpretar por usuarios no familiarizados con los clasificadores (Pereira Toledo, López Cabrera, & Quintero Domínguez, 2017).

Naive Bayes aprende las estimaciones de la probabilidad de una clase. De hecho, lo que se aprende es la distribución de probabilidad condicional de los valores de una clase dados los valores de los atributos. No obstante, NB solo es capaz de aprender distribuciones de probabilidad condicional muy simples (Pereira Toledo, López Cabrera, & Quintero Domínguez, 2017).

2.8.7 K Nearest Neighbor

K-Nearest Neighbor (KNN) es uno de los algoritmos supervisados de machine learning basado en las similitudes y ofrece en algunos contextos un rendimiento interesante, asimismo, es una generalización para las reglas del vecino más cercano (Lévano Rodríguez & Cerdán León, 2022). KNN es un algoritmo simple, robusto y versátil para clasificación y se ha utilizado para diferentes tipos de problemas como: reconocimiento de patrones, clasificación de modelos, categorización de texto, bioinformática incluso en la medicina entre otros; también es un clasificador no paramétrico y de aprendizaje perezoso; al ser no paramétrico significa que está libre de suposiciones sobre las propiedades de los datos subyacentes por lo que no es necesario tener conocimientos previos sobre los datos: y es de aprendizaje perezoso porque cualquier generalización de los datos de entrenamiento se posponen hasta que los datos de la prueba se presentan al sistema (Ehsani & Drablos, 2020).

El algoritmo KNN almacena todos los datos disponibles y clasifica un nuevo punto de datos según la similitud. Esto significa que cuando aparecen nuevos datos, se pueden clasificar fácilmente en una categoría de conjunto de pozos utilizando el algoritmo KNN. En la fase de entrenamiento simplemente almacena el conjunto de datos y cuando obtiene nuevos datos, los clasifica en una categoría que es muy similar a los nuevos datos (Raschka, 2018).

En la Figura 2.13 se muestran dos paneles, en el de la izquierda los datos de entrenamiento son mostrados en puntos azules, y un punto de consulta que se quiere saber marcado con un signo de interrogación. En el lado derecho, se encuentran las etiquetas de las clases, y el círculo punteado indica el vecino más cercano de acuerdo con el punto marcado, asumiendo una métrica de distancia euclidiana. En este caso, el algoritmo predice que la etiqueta de la clase de acuerdo con el punto más cercano es al conjunto de entrenamiento marcado como la clase 0.

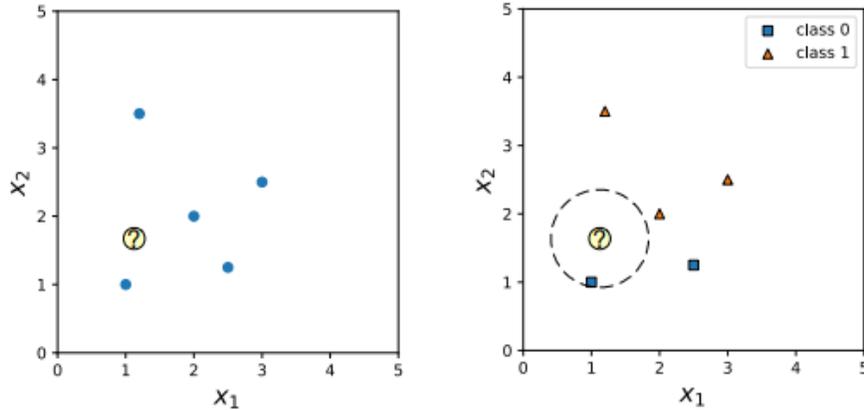


Figura 2.13. Clasificación de KNN en dos dimensiones

2.8.8 Linear Discriminant Analysis

El Análisis Discriminante Lineal (LDA) o discriminante de Fisher es una técnica común utilizada para la reducción y clasificación de dimensionalidad. A diferencia del Análisis de Componentes Principales (PCA), que busca maximizar la varianza total de los datos, el LDA se centra en maximizar la separación entre las clases (Mohanty, 2013).

El Discriminante Lineal (LDA) no es solo una herramienta de reducción de dimensiones, sino también un método de clasificación sólido.

El LDA se utiliza como herramienta para clasificación, reducción de dimensiones y visualización de datos. Ha existido desde hace bastante tiempo. A pesar de su simplicidad, LDA a menudo produce resultados de clasificación sólidos, decentes e interpretables. Cuando se abordan problemas de clasificación del mundo real, LDA suele ser el método de evaluación comparativa antes de emplear otros más complicados y flexibles (Xiaozhou, 2020).

Se trata de un algoritmo de aprendizaje supervisado, lo que significa que utiliza etiquetas de clases. Es más adecuado para la separación de clases. Utiliza matrices dispersión dentro de clase y entre clases.

2.8.9 Quadratic Discriminant Analysis

El Análisis Discriminante Cuadrático (QDA) es una técnica de aprendizaje automático y estadística utilizada en problemas de clasificación. Es similar al Análisis Discriminante Lineal (LDA), pero difiere en su suposición sobre la matriz de covarianza de las clases. Además, permite que las clases tengan matrices de covarianza diferentes. Esto significa que el QDA puede capturar relaciones más complejas entre las características de las clases. Esta diferencia se puede observar en la Figura 2.14, donde la fila inferior demuestra que el LDA sólo puede aprender límites lineales, mientras que el QDA puede aprender límites cuadráticos y, por lo tanto, es más flexible.

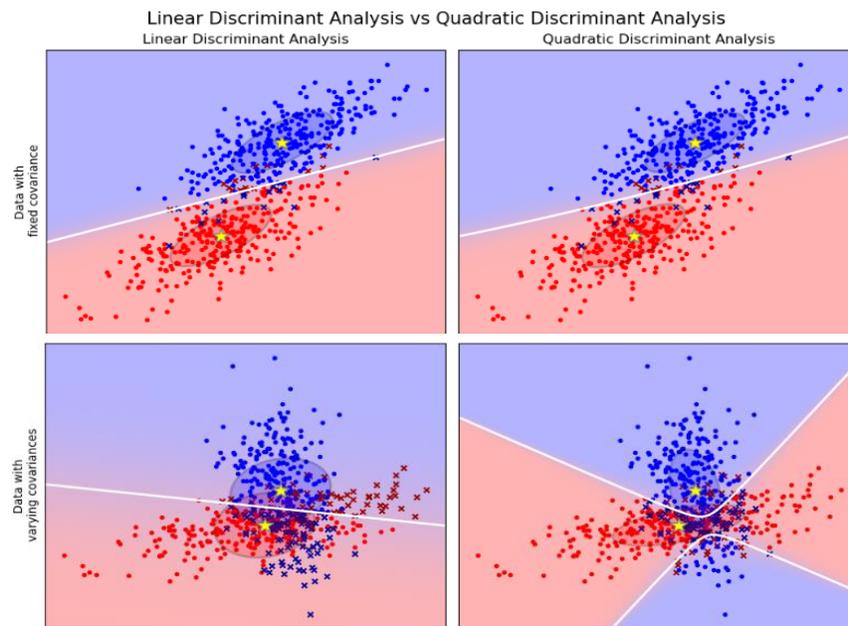


Figura 2.14. Gráfico de límites de decisión para LDA y QDA

2.9 Hiperparámetros

Los hiperparámetros son parámetros o bien configuraciones ajustables que se establecen antes del proceso de entrenamiento de un modelo generado por un algoritmo de machine learning. No obstante, estos parámetros no se aprenden automáticamente de la base de datos, a diferencia de los parámetros propios del modelo que se ajustan durante la fase de entrenamiento.

Los hiperparámetros son indispensables para optimizar el rendimiento del modelo y adaptarlo a un aprendizaje para resolver un problema específico.

Algunos de los hiperparámetros más comunes son:

- **Tasa de Aprendizaje (Learning rate):** controla el tamaño de los pasos que se toman durante el proceso de optimización (por ejemplo, en el descenso de gradiente). Un valor más alto puede hacer que el entrenamiento sea más rápido, pero un valor muy alto puede hacer que el modelo no converja.
- **Número de épocas (Epochs):** representa cuántas veces se entrena un modelo en el conjunto de datos completo. Es un hiperparámetro importante en el entrenamiento de redes neuronales y se utiliza para controlar el tiempo y los recursos de entrenamiento.
- **Profundidad del Árbol (Tree Depth):** En algoritmos de árboles de decisión como el Random Forest o Gradient Boosting, se puede controlar la profundidad máxima del árbol (max Depth). Sin embargo, un árbol más profundo puede llevar al sobreajuste.
- **Número de Árboles (Number of Trees):** En algoritmos de conjunto como Random Forest o Gradient Boosting, se puede especificar cuántos árboles se utilizarán en el conjunto.
- **Número de vecinos (Neighbors):** En algoritmos de KNN, este hiperparámetro controla la cantidad de vecinos que se consideran al realizar una predicción.
- **Kernel:** En algoritmos como Máquinas de Soporte Vectorial (SVM), se puede seleccionar un kernel específico, como lineal, polinómico o radial.

La selección adecuada de hiperparámetros es esencial para optimizar el rendimiento del modelo. Esto se puede hacer mediante técnicas de búsqueda de hiperparámetros, como búsqueda en cuadrícula (grid search) o búsqueda aleatoria (random search), que exploran diferentes combinaciones de valores de hiperparámetros para encontrar la configuración de hiperparámetros óptima.

2.10 Métricas de desempeño

Las métricas de desempeño, también conocidas como métricas de evaluación o métricas de rendimiento, son medidas utilizadas para evaluar qué tan bien funciona un modelo de aprendizaje automático en una tarea específica.

Para evaluar el desempeño de un algoritmo de aprendizaje automático existen una serie de métodos y fórmulas que califican en distintos aspectos su rendimiento al completar la tarea asignada (Buitrago Ariza, 2023). Estas métricas se ajustan a diferentes tipos de conjuntos de datos, es decir, existen métricas tanto para datos balanceados como no balanceados, para clasificación y para regresión, entre otros, y es posible utilizarlas para medir el rendimiento de cualquier modelo, sin embargo, las que mejor se ajusten a cierto tipo de datos son las que tendrán una mejor evaluación.

A los modelos se les debe evaluar la calidad con la que predicen problemas de clasificación o bien de regresión. Según (Contreras, Fuentes, & Rodríguez , 2020) la evaluación de los algoritmos puede hacerse por cuatro métricas diferentes: Exactitud, precisión (especificidad) Recall (sensibilidad) y medición de puntaje F1. No obstante, estas métricas proporcionan información cuantitativa sobre la calidad de las predicciones del modelo y ayudan a determinar si el modelo es adecuado para el problema en cuestión. Algunas de las métricas de desempeño más comunes incluyen:

- Precisión (Accuracy): esta métrica mide la proporción de predicciones correctas sobre el total de predicciones realizadas por el modelo.
- Recuperación (Recall o Sensitivity): La recuperación mide la proporción de verdaderos positivos (instancias positivas correctamente clasificadas) sobre el total de instancias positivas en los datos. Es especialmente importante cuando la identificación de casos positivos es crítica, como en las aplicaciones médicas.
- Especificidad (Specificity): Esta métrica mide la proporción de verdaderos negativos (instancias negativas correctamente clasificadas) sobre el total de instancias negativas en los datos. Es relevante en problemas donde la detección de falsos positivos es costosa o problemática.
- F1-Score: Es una métrica que combina precisión y recuperación en un solo valor. Es útil cuando se necesita un equilibrio entre ambas técnicas. Se calcula como la media armónica de precisión y recuperación.
- Precisión Balanceada (Balanced Accuracy): es una métrica útil en problemas de clasificación desequilibrados y proporciona una evaluación más equitativa del rendimiento del modelo en todas las clases.

- **MCC (Matthews Correlation Coefficient):** El MCC es una métrica que tiene en cuenta los cuatro valores de la matriz de confusión (verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos). Proporciona una medida de la calidad de las predicciones, considerando los desequilibrios en las clases.
- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** La curva ROC y el área bajo la curva ROC (AUC) se utilizan comúnmente para evaluar modelos de clasificación binaria. Cuanto mayor sea el AUC, mejor será el modelo para distinguir entre las clases.
- **RMSE (Root Mean Squared Error):** Esta es una métrica común en problemas de regresión. Mide la raíz del error cuadrático medio entre las predicciones del modelo y los valores reales. Cuanto menor sea el RMSE, mejor será el rendimiento del modelo.
- **MAE (Mean Absolute Error):** El MAE es otra métrica utilizada en problemas de regresión que mide el promedio de las diferencias absolutas entre las predicciones y los valores reales. Es menos sensible a valores atípicos que el RMSE.
- **R-cuadrado (R-squared):** Esta métrica evalúa la capacidad del modelo para explicar la variabilidad en los datos. Un valor más cercano a 1 indica un buen ajuste del modelo.
- **Coeficiente de similitud de Jaccard:** es una métrica útil para medir la similitud entre conjuntos, ya que es independiente del tamaño absoluto de los conjuntos. Es especialmente relevante en aplicaciones donde se desea comparar la similitud entre elementos sin tener en cuenta su cantidad.

2.11 Conclusión

El monitoreo y mantenimiento efectivo de transformadores de potencia son esenciales para garantizar su funcionamiento óptimo y prevenir fallas. La complejidad de estos sistemas, compuestos por diversos subsistemas, demanda un enfoque integral. La combinación de métodos tradicionales, como el análisis de gases disueltos en el aceite y la detección de descargas parciales, junto con tecnologías emergentes, como la inteligencia artificial y el aprendizaje automático, permite obtener índices de salud con mayor precisión y tomar mejores decisiones. La aplicación de la lógica difusa en el análisis de datos demuestra el papel crucial de la tecnología avanzada en el monitoreo de transformadores. Además, la elección adecuada de hiperparámetros y métricas de desempeño en algoritmos de machine learning potencia la capacidad predictiva. En resumen, la convergencia de métodos tradicionales (basados en el conocimiento del dominio) con las nuevas tecnologías emergentes (basadas en IA) establece un marco robusto para monitorear el estado de salud y garantizar la fiabilidad y longevidad de los transformadores de potencia.

CAPÍTULO III. ESTADO DEL ARTE

Resumen

En este capítulo se realiza una revisión de la literatura, en la cual se reúne información de artículos científicos en donde se han aplicado técnicas de inteligencia artificial para el diagnóstico de fallas e índices de salud de transformadores de potencia. Como primera parte se presenta una breve introducción, posteriormente una breve descripción de los artículos por técnica de IA aplicada y al final se muestra en una tabla el concentrado de los artículos revisados.

3.1 Introducción

Actualmente el uso de la inteligencia artificial es más prominente, tanto que sus técnicas se han vuelto más conocidas debido a la adaptabilidad a problemas o sistemas complejos en presencia de incertidumbre y también debido a los resultados satisfactorios que logran obtenerse.

Un caso más de aplicación se ve representado en el diagnóstico de fallas de los transformadores de potencia. Los transformadores de potencia son equipos de gran importancia y su disponibilidad es fundamental para la seguridad y continuidad del servicio eléctrico en los procesos industriales. Existen diferentes métodos de diagnóstico que utilizan técnicas inteligentes para la detección de fallos en los transformadores de potencia.

Si bien, se genera una gran cantidad de datos a través del monitoreo, mantenimiento, reparación y diagnóstico de los transformadores de energía, sin embargo, todos estos datos no pueden preindicar el tipo exacto y la probabilidad de falla del transformador. Para superar dicha problemática en esta revisión del estado del arte se presentan distintas metodologías referentes a técnicas de inteligencia artificial que tienen como objetivo la modelación basada en datos históricos de distintos transformadores con la finalidad de obtener un modelo inteligente capaz de predecir el tipo de falla que puede presentar dicho equipo, y ser un gran aliado para la toma de decisiones en el diagnóstico de estos mismos.

3.2 Lógica difusa para la detección de fallas

Las pruebas realizadas para el monitoreo de los transformadores se pueden llevar a cabo en línea y fuera de línea, donde se pueden aplicar distintas técnicas para determinar el estado del transformador, sin embargo, se requieren de distintos parámetros para determinar alguna falla. Es por lo que (Mileta & Zlatan, 2017) analizan la aplicación del modelo mamdani para crear un sistema de diagnóstico de fallas basado en el estado actual del transformador de potencia. El estudio presenta dos casos, con un controlador único y cinco independientes; en el primer caso, las entradas del controlador son resultados de pruebas de transformadores en línea y fuera de línea como lo son la antigüedad, temperatura de sobrecalentamiento, frecuencia, análisis de respuesta, temperatura de aislamiento, análisis de gas disuelto e índice de polarización; en el segundo caso, además de las entradas existentes incluye mediciones previas. Los resultados obtenidos muestran

una efectividad aceptable en la detección de diferentes fallas y podría servir como una buena orientación en el monitoreo de la condición del transformador de potencia.

En otro trabajo, (Fernández Blanco, Hernández, & Corrales Barrios, 2021) proponen un método para el diagnóstico de fallos en un transformador de 40MVA utilizando la lógica difusa a partir del análisis de los gases disueltos en el aceite. La propuesta es simple, de fácil implementación y posee una buena precisión en la detección de fallos múltiples. Con muestras de gases disueltos fidedignos garantiza una tasa total de exactitud en la detección de fallos incipientes de 91.6%, mientras que el método del triángulo de Duval es 74.1%, el método de IEC es 52.2% y el método de Rogers es 42.8%, lo que demuestra su superioridad frente a los métodos de diagnósticos convencionales.

3.3 Machine Learning para el diagnóstico de transformadores

El uso exitoso de herramientas de Machine Learning (ML) puede encontrar múltiples aplicaciones en la industria, como proporcionar formas rápidas de analizar nuevos datos de transmisión desde sensores en línea, evaluando la importancia de las variables individuales en el contexto de evaluación de la condición de transformador y también la necesidad o adecuación de la imputación de datos en el tan ampliamente problema común de la falta de datos. Es por ello por lo que, (Cheim, 2018) presenta un trabajo que consiste en entrenar 12 algoritmos de Machine Learning con datos reales de 1,000 transformadores que fueron analizados individualmente por expertos humanos. Cada transformador en la base de datos fue calificado con una tarjeta ‘verde’, ‘amarilla’ o ‘roja’ dependiendo de los datos. El trabajo describe los principales pasos hacia el entrenamiento de los algoritmos de ML y el sorprendente resultado producido por esos algoritmos cuando se solicita para analizar 200 casos no vistos durante el entrenamiento.

Otros autores presentan un trabajo similar pero enfocado en los generadores de energía ya que la detección rápida de fallas mejora la eficiencia del sistema y reduce los costos operativos. (Quintanilha, Elias , & B. da Silva, 2021) proponen un marco diferente a lo convencional, en el que se utilizan algoritmos de aprendizaje automático (ML) para detectar y clasificar varios tipos de fallas en un sistema de generación de energía de embarcaciones posicionadas dinámicamente.

Primero, realizan un análisis de componentes principales para extraer información relevante de los datos etiquetados. Luego un algoritmo de bosque aleatorio aprende patrones ocultos del comportamiento defectuoso para inferir la detección de fallas a partir de datos no etiquetados. Los resultados sobre la detección y clasificación de fallas para el enfoque propuesto mostraron una mejora significativa en la precisión y la velocidad en comparación con los resultados de los métodos basados en reglas en una base de datos completa.

3.4 Redes neuronales artificiales en la detección de fallas

Los datos obtenidos por la aplicación de diferentes técnicas de medición son bastos para poder desarrollar modelos de simulación que empleen la inteligencia artificial. Para este caso se presenta una herramienta de diagnóstico basa en metodologías de inteligencia artificial como las redes neuronales probabilísticas, utilizadas en la detección de fallas en transformadores. (Cadena, Cadena, & Pérez Londoño, 2008) utilizan los resultados entregados por ensayos realizados sobre el aceite de un transformador, a través del análisis de gases disueltos (AGD), se obtienen los conjuntos de datos para el entrenamiento y posterior prueba para la red neuronal. Los resultados obtenidos se comparan con lo calculados mediante metodologías dadas por el estándar IEEE C57, permitiendo obtener con ello un criterio de validez sobre la metodología artificial implementada

3.5 Algoritmos genéticos en la estimación de parámetros de transformadores

(Pérez B., Matos Alfonso, & Fernández, 2009) presentan una técnica basada en Algoritmos Genéticos para la estimación de los parámetros y validación del modelo de la temperatura del aceite superior en transformadores de potencia. Para tal fin se emplean datos de un sistema de monitoreo y diagnóstico en línea instalado en un transformador de 100 MVA 230/115/24 kV OA/FA/FOA de la Subestación Barquisimeto de ENELBAR, Venezuela desde el año 2003. El objetivo del trabajo fue comparar la reducción del error entre el modelo y las mediciones reales de la temperatura del aceite superior cuando sus parámetros son estimados por algoritmos genéticos y mínimos cuadrados. La estimación de parámetros por algoritmos genéticos evidencia mejores

resultados del modelo, lo cual incrementa su desempeño como herramienta de diagnóstico para transformadores de potencia.

3.6 Trabajos de inteligencia artificial relacionados con el diagnóstico de fallas en transformadores

La aplicación de las técnicas de Inteligencia Artificial se ha explorado relativamente poco, sin embargo, se pudo ver que se han aplicado diferentes técnicas para el mismo fin. Los estudios presentados se centran en el desarrollo de un modelo inteligente de simulación capaz de clasificar el estado de los transformadores mediante las fallas reportadas, o bien determinar el índice de salud que permita ser un apoyo para los expertos en la toma de decisiones. Los trabajos que más destacan son los desarrollados con las técnicas de Machine Learning presentados por (Cheim, 2018) y (Quintanilha, Elias , & B. da Silva, 2021) con el desarrollo de sus algoritmos de aprendizaje automático para detectar y clasificar varios tipos de fallas, así como también los de Lógica Difusa presentados por (Mileta & Zlatan, 2017) y (Álvarez, Ramirez, & Fernández, 2011).

Estos estudios (Tabla 3.1a y 3.1b) sirven como referencia para el caso de estudio en esta investigación que además será una gran aportación a este estado del arte con las diferentes técnicas ya mencionadas como lo son el Machine Learning.

Tabla 3.1a. Aportaciones de la revisión de la literatura

Título	Autores	Obtención de datos	Aportación	Técnica
Analysis of artificial intelligence expert systems for power transformer condition monitoring and diagnostics	(Zarkovic & Stojkovic, 2017)	Datos obtenidos a través del monitoreo, mantenimiento, reparación y diagnóstico de transformadores de potencia.	Un modelo de simulación diseñado para caracterizar la condición de funcionamiento y determinar la urgencia de la intervención a los transformadores con posibilidad de indicar la probabilidad de un tipo específico de falla	Lógica Difusa
Machine Learning Tools in Support of Transformer Diagnostics	(Cheim L. 2018)	Datos reales obtenidos de 1000 transformadores que fueron analizados individualmente por expertos humanos	Modelo inteligente capaz de clasificar el grado de falla de los transformadores.	Machine Learning Algorithms
Aplicación de Redes Neuronales Probabilísticas en la detección de fallas incipientes en transformadores	(Cadena, Cadena, & Pérez, 2008)	Datos obtenidos mediante ensayos realizados sobre el aceite de un transformador, a través del análisis de gases disueltos (AGD)	Una herramienta de diagnóstico basa en metodologías de inteligencia artificial, para la detección de fallas de transformadores.	Redes Neuronales Artificiales
Parameter estimation and validation of power transformers top oil temperature model by applying genetic algorithms	(Pérez, Matos & Fernández. 2009)	Se emplean datos de un sistema de monitoreo y diagnóstico en línea instalado en un transformador de la subestación de ENELBAR, Venezuela	Un modelo de simulación para la estimación de valores de temperatura del aceite superior, en comparativa con los datos reales.	Algoritmos Genéticos
A fuzzy logic method for diagnosis incipient defaults in a 40 MVA transformer	(Fernández, Hernández & Corrales, 2021)	Datos históricos de pruebas con muestras de gases disueltos.	Método para el diagnóstico de fallos en un transformador de 40 MVA utilizando la lógica difusa a partir del análisis de los gases disueltos en aceite.	Lógica Difusa

Tabla 3.1b. Aportaciones de la revisión de la literatura

Título	Autores	Obtención de datos	Aportación	Técnica
Análisis de gases disueltos para monitoreo y diagnóstico de transformadores de fuerza en servicio	(Crespo S., 2015)	Muestreo de datos reales con el medidor portátil Transport X (KERLMANDLTD)	Análisis de los gases disueltos para el diagnóstico de fallas, permitiendo determinar el estado del transformador y reducir los índices de falla.	Análisis de Gases Disueltos (AGD)
Sistema de Monitoreo en Tiempo Real para el Diagnóstico de Transformadores de Potencia en una Empresa de Energía Eléctrica	(Pérez, Torrez, Fernández & Fernández, 2012)	Datos reales muestreados	Proponen un sistema de monitoreo en tiempo real para el diagnóstico de los transformadores de potencia principales de la Empresa ENELBAR-CORPOELEC-Venezuela	Análisis de Gases Disueltos en aceite (AGD) entre otros
Condition Monitoring and Fault Diagnosis of Generator in Power Networks	(Yang, Li, Ye, & Song, 2019)	Datos reales muestreados	Metodología de procesamiento de señales jerárquicas para el monitoreo de la condición del generador y el diagnóstico de fallas basada en datos de forma de onda eléctrica sin procesar en redes eléctricas, que a menudo se pueden medir mediante sensores de forma de onda estratégicamente	Técnicas de monitoreo y algoritmo jerárquico
A fault detector/classifier for closed-ring power generator using machine learning	(Quintanilha, otros, 2021)	Datos históricos	Algoritmos de aprendizaje automático para detectar y clasificar varios tipos de fallas en un sistema de generación de energía de embarcaciones posicionadas dinámicamente.	Machine Learning
Diagnóstico de Transformador de Potencia utilizando Lógica Difusa	(Sánchez, Aranda & Maldonado, 2016)	Datos históricos	Crear un sistema de diagnóstico para la comparativa con los resultados de métodos normalizados.	Lógica Difusa

3.7 Conclusión

La aplicación de técnicas de inteligencia artificial en el diagnóstico de fallas en transformadores de potencia se presenta como un campo de investigación y desarrollo prometedor. A través de enfoques como lógica difusa, machine learning, redes neuronales artificiales y algoritmos genéticos, se ha buscado modelar y predecir eficientemente el comportamiento de estos dispositivos críticos para garantizar la seguridad y continuidad de un buen servicio eléctrico.

La diversidad de métodos y enfoques refleja la complejidad inherente a los sistemas de transformadores, así como la necesidad de utilizar enfoques avanzados para aprovechar la gran cantidad de datos generados durante la operación y monitoreo de estos equipos.

La lógica difusa ha demostrado su utilidad al considerar múltiples parámetros para la detección de fallas, mientras que los algoritmos de machine learning han destacado en la clasificación eficiente de transformadores en diferentes estados de salud.

En este panorama, se destaca la necesidad de seguir avanzando en la aplicación de la inteligencia artificial para desarrollar modelos más precisos y robustos. La combinación de diversas técnicas y la exploración de nuevos enfoques pueden llevar a soluciones más integrales y efectivas en el diagnóstico de fallas en transformadores de potencia. Este progreso contribuirá significativamente a la mejora de la confiabilidad y la eficiencia de los sistemas eléctricos, así como a la toma de decisiones informada en el mantenimiento y operación de transformadores.

CAPÍTULO IV. METODOLOGÍA

Resumen

En este capítulo se describe la metodología, los procedimientos, análisis y la clasificación de las variables; con la información presentada anteriormente se propone el desarrollo de varios modelos inteligentes que permitan diagnosticar fallas en un transformador de potencia y de ese modo se puedan tomar las mejores decisiones para abordar el diagnóstico.

4.1 Metodología

Actualmente las técnicas de IA han incursionado en el desarrollo de modelos inteligentes para el diagnóstico de fallas de los transformadores de potencia, así como se mostró en el estado del arte. Esto, debido a que se han obtenido buenos resultados en las predicciones, además de la modelación de la incertidumbre y subjetividad, no obstante, si bien es un gran complemento a la toma de decisiones, en México no se ha abordado mucho sobre el tema. Lo anterior puede derivar de la falta de expertos para llevar a cabo las pruebas pertinentes y necesarias como parte de la inspección y diagnóstico de los equipos y a su vez de la obtención de datos, y de ese modo el desarrollo de modelos inteligentes que sirvan al diagnóstico de fallas sin requerir tantas pruebas físicas.

Así mismo cuando se emplea ML con el fin de permitir que un sistema aprenda a partir de datos y se obtengan resultados de predicción satisfactorias, resulta fundamental adherirse a un proceso claramente definido. Esto garantiza una orientación efectiva durante el desarrollo y proporciona una guía para abordar situaciones en las que los resultados no alcancen las expectativas. Siendo así, el desarrollo de este trabajo de investigación se basa en una metodología básica de un proceso de ML.

De acuerdo con (GERON, 2017), la metodología se desarrolla en siete etapas que tienen como objetivo principal transformar una problemática en un sistema funcional. En este contexto, se ha aplicado esta metodología para desarrollar un sistema de diagnóstico de fallas en transformadores de potencia. En los siguientes puntos se detallan las etapas de la metodología:

1. Análisis General: En esta etapa, se enfoca en comprender los objetivos del proyecto, delimitar la problemática de interés, buscar posibles soluciones y generar una propuesta de solución.
2. Obtención de Datos: Aquí se determina y recopilan los datos necesarios para abordar la problemática.
3. Exploración: En este punto, se exploran los datos para obtener información. Esto se logra mediante un análisis de atributos, la aplicación de estadísticas, la visualización de datos y la identificación de posibles transformaciones.

4. Procesamiento: Esta etapa comprende la limpieza de datos, la selección y escalado de atributos, y la aplicación de ingeniería de características, con el propósito de extraer conocimiento y mejorar la comprensión de los datos.
5. Elección algorítmica: En esta fase, se seleccionan algoritmos pertinentes para abordar la problemática y se evalúa su rendimiento utilizando diversas métricas. Este análisis permite elegir el algoritmo más adecuado y confiable.
6. Ajuste del modelo: En este punto se calibran los parámetros del modelo para asegurarse de que sea la mejor opción para resolver el problema. Para ello, se implementa un algoritmo de optimización y validación cruzada, se entrena y evalúa el modelo.
7. Obtención del modelo e implementación: La última etapa consiste en convertir el modelo en un sistema completo capaz de interactuar con expertos y usuarios, permitiendo su aplicación en situaciones prácticas.

Por otra parte, el modelo de lógica difusa (LD) permite representar el sistema bajo consideración de las variables de entrada y de salida, por medio de los conjuntos difusos representados en términos lingüísticos. La metodología de LD se basa en un modelo tipo mamdani, compuesto por tres etapas: la primera etapa es la fusificación que es la representación gráfica de las variables de entrada por medio de conjuntos difusos; la segunda, es el mecanismo de inferencia el cual consiste de reglas de inferencia del modelo resultante a partir de los conjuntos difusos de las variables de entrada; finalmente la tercera etapa se trata de la defusificación, la cual es la representación gráfica de las variables de salida a través de los conjuntos difusos.

Estas metodologías (Figura 4.1 a y 4.1 b) tienen como objetivo llevar a cabo el desarrollo de modelos inteligentes que permitan predecir el tipo de falla de un transformador de potencia.

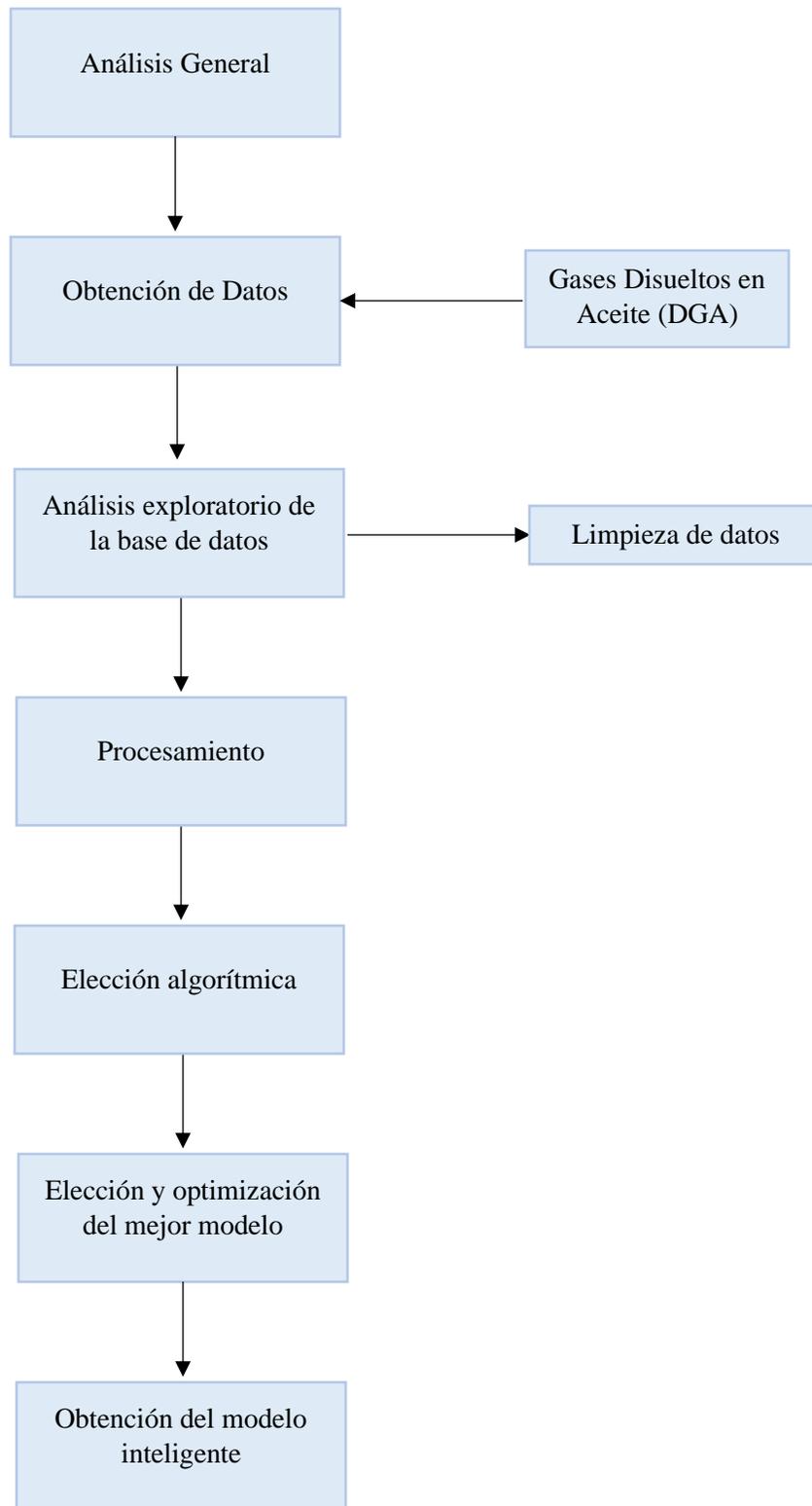


Figura 4.1 a. Metodología para el desarrollo de modelos de ML.

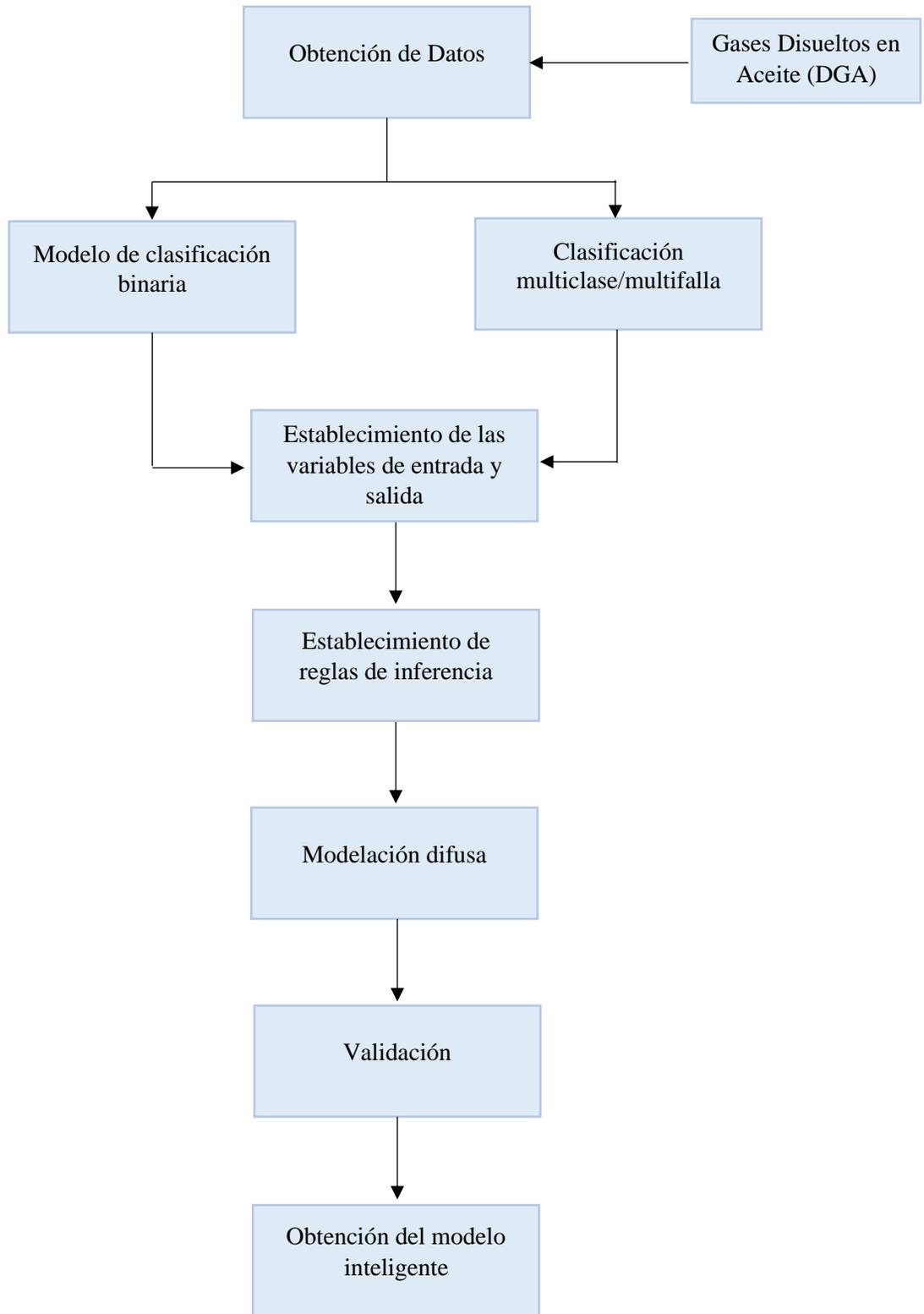


Figura 4.1 b. Metodología para el desarrollo de modelos de LD

4.2. Análisis General

En esta fase se delimitan tanto los alcances como los límites del problema en función de su aplicabilidad final, es decir, lo que se espera que el modelo sea capaz de resolver. Además, se brinda un marco de referencia para visualizar el espacio de soluciones que se pretenden encontrar. Esto incluye si la solución al problema asistirá o bien será de apoyo a la toma de decisiones, o si esta se espera que sea el reemplazo de alguna otra herramienta actualmente usada o bien que funja ambos roles.

4.2.1. Definición del problema bajo su dominio

El problema a resolver es la predicción de fallas en transformadores de potencia mediante la obtención de gases disueltos en aceite, es decir, mediante los datos obtenidos por el DGA se pueden hacer inferencias y predecir el comportamiento que los transformadores tendrán, esto categorizado a un tipo de falla en específico y el cual determinará si este requerirá de algún tipo de mantenimiento o en el peor de los casos un cambio de equipo, sin embargo, estas serían decisiones que le conciernen a los expertos.

Los gases disueltos en aceite determinan ciertas concentraciones de gases específicos, estos a su vez son interpretados sintónicamente por los expertos utilizando herramientas clásicas que no brindan una respuesta precisa, sin embargo, son de utilidad para hacer inferencias. Es por ello, que optar por modelos inteligentes capaces de ser entrenados y validados y finalmente sean capaces de hacer clasificaciones más precisas se ve como una buena alternativa para su desarrollo.

De acuerdo con los expertos, se requiere de un modelo inteligente de machine learning que sea capaz de clasificar las características de entrada (concentraciones de gases) y etiquetarlas a una variable de salida que indique un tipo de falla correspondiente dentro de un conjunto de posibilidades de esta misma variable.

4.2.2. Determinación del tipo de problema

La configuración de ciertas características (concentraciones de los gases) da como resultado el origen de una falla. Se ha descrito antes, que este tipo de arquitectura en la base de datos corresponde a una de clasificación, donde los diferentes arreglos de características se asocian a una etiqueta particular, en este caso una variable de salida denominada diagnóstico de falla. No obstante, la variable de salida puede ser solo una de un conjunto de posibilidades, es decir, se trata de una variable categórica.

Al tratarse de un problema de clasificación, este debe ser evaluado no solo con las métricas propias de los algoritmos de clasificación, ya que, la precisión, recall, f1 score, entre otros, evalúan más a detalle las clases pertenecientes en la base de datos, es decir, permite comprender el comportamiento de los modelos por medio de evaluar clase por clase donde se involucran valores como la tasa de falsos positivos, falsos negativos, así como también métricas que conjuguen a ambos como lo son la especificidad y la sensibilidad.

4.2.3. Generación de supuestos

La clasificación con algoritmos de aprendizaje automático implica varios supuestos y consideraciones que deben cumplirse o tenerse en cuenta para obtener resultados efectivos y significativos. A continuación, se presentan algunos supuestos y consideraciones que son clave para la clasificación con algoritmos de ML:

- **Datos etiquetados:** La clasificación supervisada requiere que cada instancia del conjunto de datos para la fase de entrenamiento esté etiquetada, es decir que cada caso en el conjunto de datos tenga una etiqueta o clase conocida.
- **Independencia de características:** se asume que las instancias de la base de datos son independientes entre sí. Esto significa que las predicciones para una instancia o caso no dependen de las predicciones para otras instancias.
- **Homogeneidad de características:** se asume que las características utilizadas para la clasificación son relevantes y estas tienen el mismo significado consistente en todos los casos.

- Representatividad: El conjunto de datos de entrenamiento debe ser representativos y balanceados de la población que se aplicará al modelo. Si los datos de entrenamiento no son representativos o balanceados, es probable que el modelo no generalice bien a nuevos datos.

4.2.4. Propuesta

Por lo tanto, la propuesta se define como: Un modelo inteligente basado en técnicas de inteligencia artificial que sea capaz de usarse en cualquier computadora y con facilidad de uso, que pueda clasificar valores de concentraciones de gases disueltos en aceite en fallas que suelen tener los transformadores de potencia con una precisión aceptable.

4.3. Obtención de datos

En esta fase además de obtenerse la base de datos, también se conocen las características de los datos, así como su preparación previa antes de ser usada, para generar todos los desarrollos requeridos en otras áreas tanto de software u organizacionales para la adquisición y posterior procesamiento de estos datos. Esto incluye la creación de bases de datos relacionales o no relacionales; pero no se limita a ello, sino también tener métodos efectivos de captura de datos, en casos donde se generan de forma manual o se encuentran en papel.

4.3.1. Determinación de datos necesarios

Es preferente tener tantos datos como sea posible, además de tenerlos en un archivo tabular donde se desglosen las filas de una base de datos grande la cual permita un mejor entrenamiento para cualquier algoritmo que se vaya a utilizar. Sin embargo, también se requiere obtener la mayor cantidad de características, es decir, las columnas de las variables de entrada para determinar cuáles de estas influyen en la etiqueta de salida. Es posible que en fases subsecuentes se reduzcan los conjuntos de atributos mediante técnicas para elegir los más importantes y significativos.

4.3.2. Obtención de la base de datos

En los problemas de clasificación, siempre es preferible que la base de datos esté balanceada, es decir, que cada conjunto de clase sea la misma cantidad, sin embargo, esto no siempre es posible, por ejemplo: en la medicina es habitual tener más datos de ausencia de enfermedad que de enfermedades o bien para este caso, es posible que un tipo de falla sea la dominante en los transformadores.

El desarrollo de la base de datos (BD) es realizado por el Instituto Nacional de Electricidad y Energías Limpias (INEEL), siendo esto parte del desarrollo de proyectos que la institución realiza como parte del objetivo de su misión que es impulsar el desarrollo sustentable de la industria energética mediante la innovación.

La BD fue creada con los datos obtenidos de diferentes fuentes bibliográficas donde se aplicó la prueba química de análisis de gases disueltos en aceite que realizan los expertos a los transformadores de potencia como parte del monitoreo y mantenimiento de los transformadores. Mediante esta técnica (AGD) se obtienen valores en partes por millón de cinco diferentes gases. Esta BD consta de 741 muestras de pruebas de análisis de gases disueltos en transformadores de potencia.

El análisis de gases disueltos (AGD) es la técnica más elemental, económica y precisa para diagnosticar las fallas del transformador. Estas fallas se detectan analizando la concentración de gases disueltos tales como Hidrógeno (H_2), Metano (CH_4), Etano (C_2H_6), Etileno (C_2H_4) y Acetileno (C_2H_2). Las técnicas de interpretación de estas fallas se han establecido en diversos estándares de asociaciones de ingeniería relevantes, como IEEE e IEC (IEC, 1978), (IEEE, 1991). Se considera que los resultados de este análisis junto con su interrelación con otras variables pueden ser de gran utilidad para determinar el diagnóstico de un transformador. Para el entrenamiento del modelo de determinación del diagnóstico basado en AGD, específicamente se requieren los datos que se describen en la Tabla 4.1.

Tabla 4.1. Análisis de gases disueltos en aceite y otros parámetros de medición.

No.	Dato de entrada	Unidad	Descripción
1	Hidrógeno (H ₂)		
2	Metano (CH ₄)		
3	Etano (C ₂ H ₆)	ppm	Número real positivo con 2 decimales. Valor máximo 60,000
4	Etileno (C ₂ H ₄)		
5	Acetileno (C ₂ H ₂)		

En la Tabla 4.1 se muestran las cinco variables de entrada que contienen la base de datos, siendo estas las concentraciones en partes por millón (ppm) que cada gas libera en las descargas parciales de los transformadores. Además, cuenta con una variable de salida la cual se denomina multiclase ya que para cada uno de los 741 casos se obtiene uno de los 10 tipos de fallas, como se muestra en la Tabla 4.2.

Tabla 4.2. Codificación del diagnóstico de fallas de los transformadores de la variable de salida multiclase.

Diagnóstico de falla	Nomenclatura
Normal	Condición normal
Descargas parciales	PD
Descargas de baja energía	D1
Descargas de alta energía	D2
Falla térmica de menos de 300°C con sobrecalentamiento	T1 O
Falla térmica de menos de 300°C con carbonización	T1 C
Falla térmica entre 300°C y 700°C con carbonización	T2 C
Falla térmica de más de 700°C con carbonización	T3 C
Falla térmica de más de 700°C que solo involucra aceite	T3 H
Stray gassing	S

En la Figura 4.2 se presenta una muestra de lo que es la base de datos obtenida por el INEEL en formato Excel. Las primeras cinco columnas representan a las variables de entrada que son las concentraciones de cada uno de los cinco gases que se obtienen en la prueba AGD; la columna de “Pentágonos” es un prediagnóstico sin embargo no se tomará como parte de la modelación; en la columna “Diagnóstico final” se muestra la variable de salida que puede ser una de las clases que se mostró anteriormente en la Tabla 4.2 en la última columna “Diagnóstico de la fuente bibliográfica” se presentan los diagnósticos que se tienen como resultado de acuerdo con la revisión de la literatura o bien el estado del arte.

A	B	C	D	E	F	G	H
H2	CH4	C2H2	C2H4	C2H6	Pentágonos	Diagnóstico final	Diagnóstico de la fuente bibliográfica
40000	400	6	600	70	PD	PD	Low/Middle-temperature overheating
29	204	0.1	17	264	T1 O	T1 O	Overheated iron sheets
555	1050	29	3520	489	T3-H	T3-H	Burned selector in oil
754	2647	6	2590	1127	T2 C	T2 C	Carbonized leads
2070	31879	55	38192	1127	T3-H	T3-H	Carbonized paper
6	46	0	9	12	T1 C	Condición normal	Carbonized winding turns
10000	1000	0	0	200	PD	PD	PD
0	2000	0	0	700	T1 O	T1 O	O
15000	1500	0	0	0	PD	PD	PD
12000	1000	0	0	500	S	S	S
300	150	0	75	0	T3 C	T3 C	C
6000	600	0	0	400	S	S	S
29	204	0	17	264	T1 O	T1 O	Sobrecalentamiento de laminaciones del marco inferior
0	416	0	0	1216	T1 O	T1 O	Sobrecalentamiento en terminal
555	1050	29	3520	489	T3-H	T3-H	Selector quemado en aceite encontrado por inspección
12452	104269	1231	206910	36794	T3-H	T3-H	ración de C2H4 mayor a 200,000 ppm sin falla, debido a que el problema térmico es solo e
130	844	14	351	1430	T1 O	T1 O	T1 O
784	529	317	356	488	S	S	S
654	0	313	5	0	D1	D1	D1

Figura 4.2. Base de datos proporcionado por el INEEL

Sin embargo, para el desarrollo del modelo inteligente solo se usaron las concentraciones de los gases que son las primeras cinco columnas y la columna del diagnóstico final, dejando descartadas la columna de pentágonos y diagnóstico de la fuente bibliográfica ya que se consideran que no tienen gran relevancia para el modelo.

4.3.3. Entorno de trabajo

El entorno se refiere a toda la configuración del equipo de cómputo donde se desarrolló el trabajo, es decir, desde la instalación de los softwares y módulos necesarios para hacer Machine Learning. Además, el entorno donde se realizó para poder facilitar su distribución, la integración de documentos en código y comentarios de forma legible, así como también contenido multimedia y

poder ser compartidos fácilmente. También, un entorno más sencillo en el que se pueda hacer uso de gráficos

4.3.3.1. Jupyter Notebooks

Proyecto Jupyter es de código abierto, fue creado en 2011 para trabajar interactivamente en las comunidades científicas y computacionales, sin ánimos de lucro. Sus creadores observaron que los científicos de datos tenían problemas para compartir descripciones detalladas y entendibles de su código a sus colegas (Argumedo, 2019).

Estos son documentos digitales basados en la web que sirven para escribir código, documentarlo, correrlo, visualizar datos y exportarlos sin salir del mismo entorno. Es una herramienta poderosa para aplicar cualquier metodología tanto de ML como de ciencia de datos (Argumedo, 2019).

En la Figura 4.3 se puede observar la interfaz un Jupyter Notebook, donde se observa que es posible insertar hipervínculos, imágenes, así como también videos. El código se ejecuta individualmente por celdas y muestra su resultado al instante, dichos resultados pueden ser numéricos, tabulares, figuras o bien gráficos. Se usó JN para la creación de los modelos, para las distintas fases posteriores de la metodología, con el propósito de que se puedan seguir paso a paso para un mejor entendimiento.

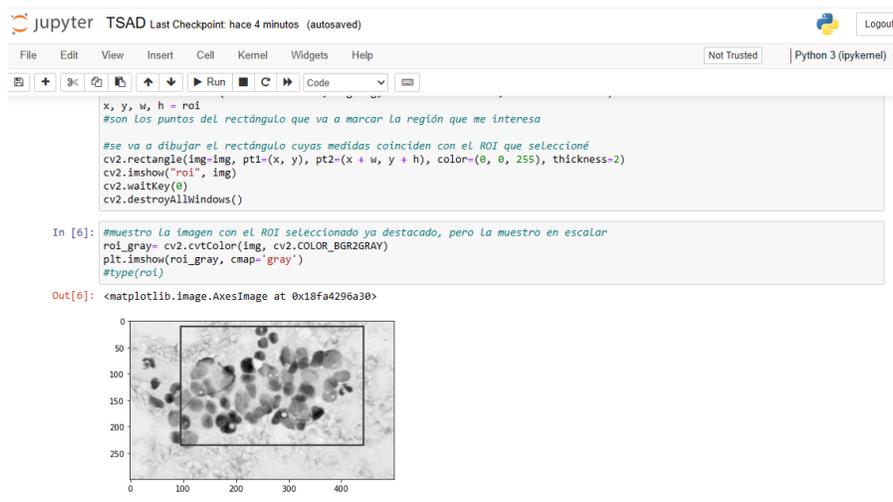


Figura 4.3. Interfaz de un Jupyter Notebook

4.3.3.2. Configuración

Para la interpretación de Python se utilizó la distribución de Anaconda, ya que es de código abierto, y especialmente compilada para su uso en cómputo científico. Posee la mayoría de los módulos necesarios para iniciar un análisis de datos o un experimento de ML (Argumedo, 2019). Para su instalación es necesario ingresar al sitio web de Anaconda (<https://www.anaconda.com/distribution>) e instalar la versión de 64 bits.

En la Figura 4.4 se puede observar la interfaz de Anaconda Navigator, donde se pueden gestionar varios intérpretes virtuales.

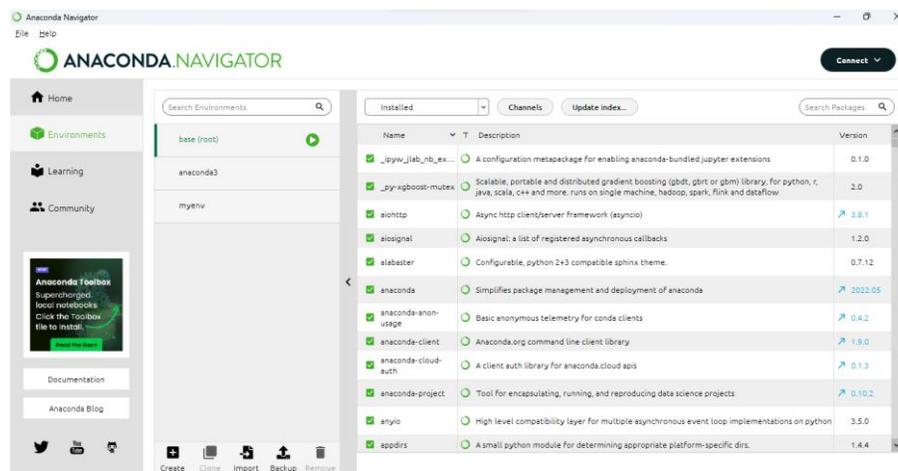


Figura 4.4. Interfaz de Anaconda Navigator

Como se estableció anteriormente, Anaconda tiene una forma sencilla de instalar y configurar los módulos de python relevantes para la creación del entorno de desarrollo, a continuación, se presentan algunos de los módulos:

- Jupyter: Permite el uso de documentos digitales
- Pandas: Sirve para manipular datos tabulares
- Numpy: Módulo para cómputo matemático
- Matplotlib: Permite realizar gráficos y figuras
- Sklearn: Módulo de ML y Ciencia de Datos

4.3.4. Datos

De este modo se obtuvo una carpeta dentro donde se encuentra la base de datos con sus respectivas clases, que posteriormente se analizaran para determinar sus características. Esto servirá para continuar con las siguientes etapas y generará código capaz de ser revisado, probado y replicado.

4.4. Exploración

La fase de exploración hace hincapié al conocimiento general de la base de datos, es decir, se refiere a explorar la BD para extraer la información necesaria para poder prepararla o bien convertirla en elementos que permitan a los algoritmos extraer información y aprender de ellos.

En esta fase se estudia cada atributo y sus características como pueden ser: el tipo de dato (entero, flotante, booleano), datos y valores faltantes, así como el porcentaje de cada clase, entre otros.

Al término de esta fase, se estará familiarizado con la base de datos, además, esto permitió diseñar los algoritmos y que estos tuvieran mayor facilidad de procesarlos y no tener problemas al momento de obtener resultados.

4.4.1. Análisis de atributos

En Machine Learning se utilizan datos tabulares para realizar el entrenamiento y aprendizaje de algoritmos, en donde se tienen filas y columnas: cada fila representa un caso o una muestra y las columnas en datos tabulares se dividen en dos: en atributos que son las características o bien las variables de entrada y en etiquetas que representa la variable de salida o bien la clase o etiqueta a la que pertenece cada muestra.

4.4.1.1. Atributos para la clasificación

En el apartado *obtención de la base de datos* se mostró la base de datos que fue proporcionada por el INEEL, dicha BD consta de 741 casos de fallas de transformadores. La base de datos que se

agrega a phyton para conocer sus características básicas y posteriormente ser procesada es de 656, dejando así 85 casos, que serán utilizados para probar el rendimiento de predicción del modelo.

La BD se encuentra en un archivo de Excel, sin embargo, para ser cargada al Jupyter Notebook se pasó a un formato csv, ya que este es un tipo de formato de archivo que es ampliamente utilizado para representar datos tabulares, hojas de cálculo entre otros. En un archivo csv, los datos se organizan en filas y columnas, y cada valor de celda está separado por comas (u otro delimitador, como punto y coma). También, el archivo creado en el Jupyter Notebook debe estar en la misma carpeta donde se encuentra la base de datos en formato csv.

Después de haber cargado el documento se ejecutan los códigos correspondientes para establecer la base de datos. En la Figura 4.5 se presenta el data frame (marco de datos) que el notebook capturó.

	H2	CH4	C2H2	C2H4	C2H6	Diagnostico
0	0.0	35.0	0.0	19.0	13.0	0
1	7.0	4.0	0.0	16.0	0.0	0
2	0.0	5.0	0.0	0.0	0.0	0
3	8.0	5.0	0.0	6.0	0.0	0
4	0.0	0.0	0.0	0.0	6.0	0
...
652	60.0	340.0	10.0	391.0	43.0	8
653	816.0	187.0	999.0	250.0	17.0	2
654	201.0	780.0	76.0	1508.0	217.0	8
655	509.0	1121.0	32.0	1444.0	298.0	8
656	9478.0	4066.0	1298.0	663.0	351.0	2

657 rows × 6 columns

Figura 4.5. Base de datos capturada por el JN para su procesamiento

4.4.1.2. Conteo de clases y categorías

En apartados anteriores se mostró las características que comprende la base de datos. En la Tabla 4.3 se presenta la configuración de las etiquetas o bien la variable de salida, ya que al ser multiclase esta debe ser clasificada de manera simple, esto se realiza manualmente en el archivo de Excel que fue proporcionado por el INEEL y antes de ser convertido a csv. De esta forma, se evitan algunos pasos de la ingeniería de características y se obtiene una BD más limpia para ser procesada por los algoritmos.

Tabla 4.3. Resultados del análisis exploratorio de la base de datos de entrenamiento.

Clase	Categoría	Tipo de falla	Cantidad
0	Normal	Condición Normal	221
1	PD	Descargas parciales	17
2	D1	Descargas de baja energía	69
3	D2	descargas de alta energía	69
4	T1 O	Falla térmica de menos de 300°C con sobrecalentamiento	86
5	T1 C	Falla térmica de menos de 300°C con carbonización	3
6	T2 C	Falla térmica entre 300°C y 700°C con carbonización	18
7	T3 C	Falla térmica de más de 700°C con carbonización	19
8	T3 H	Falla térmica de más de 700°C que solo involucra aceite	83
9	S	Stray gassing	72

4.4.2. Aplicación de Estadísticas

En este apartado se explora un poco más las características numéricas de los datos que componen a la base de datos, como, por ejemplo: valores mínimos, media, entre otros. También la correlación entre los atributos para determinar si se trabaja bajo un sistema lineal o no.

4.4.2.1. Estadística descriptiva a la BD

Una vez cargada la BD en el jupyter notebook, es posible codificar y obtener las estadísticas necesarias para conocer mejor la estructura de la base de datos. Además, con estos estadísticos se puede empezar a ver qué tan equilibrada o desbalanceada se encuentra la BD.

En la Tabla 4.4 se presenta un concentrado de información estadística extraída de la BD que será entrenada por los algoritmos de ML. La primera columna representa el estadístico, las siguientes cinco columnas representan los atributos o bien características de la BD, en este caso se visualiza la nomenclatura del gas correspondiente. La última columna corresponde a la etiqueta o bien la variable de salida nombrada como “Diagnóstico”.

Tabla 4.4. Estadísticos de la base de datos a entrenar.

	H2	CH4	C2H2	C2H4	C2H6	Diagnostico
count	657.000000	657.000000	657.000000	657.000000	657.000000	657.000000
mean	1117.817473	678.001155	381.986254	667.731335	293.599265	3.461187
std	5170.519241	3257.141559	2861.877081	3249.579283	1697.235820	3.316914
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.960000	3.850000	0.000000	4.100000	0.800000	0.000000
50%	41.000000	62.000000	0.100000	21.310000	22.330000	3.000000
75%	290.000000	255.000000	29.000000	197.000000	155.000000	7.000000
max	92600.000000	61000.000000	57000.000000	42100.000000	29990.000000	9.000000

En la Figura 4.6 se presenta un gráfico de barras que representa el número de etiquetas por cada clase, del mismo modo se puede observar en la Figura 4.7 un gráfico de pastel que representa los porcentajes que ocupa cada clase del total de muestras en la base de datos.

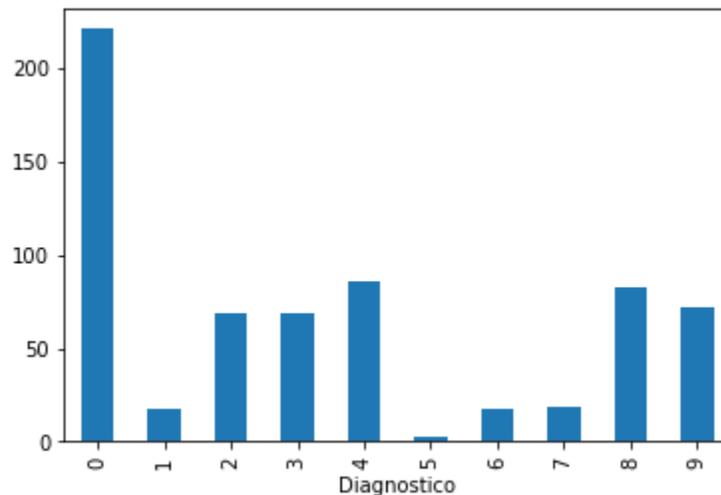


Figura 4.6. Número de etiquetas por clase

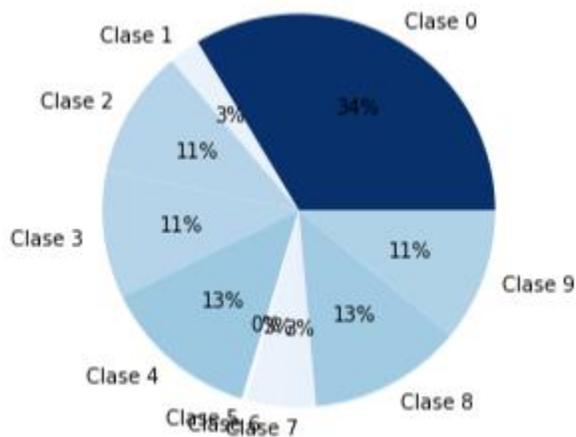


Figura 4.7. Porcentaje de etiquetas por clase

4.4.3. Información

Esta fase del proceso ha generado información pertinente que ayudará posteriormente al procesamiento de los datos en su entrenamiento. De este modo, se sabe cuántos datos se tienen y a que clase pertenecen cada uno, además, de información estadística de la base de datos.

4.5. Procesamiento

El procesamiento es una fase que difiere entre el Machine Learning tradicional y el Deep Learning. Puesto que en el primer caso se enfoca en datos tabulares que suelen ser abundantes, mientras que en el segundo caso se trabaja con menos datos; lo que conlleva realizar una fase de procesamiento la cual se enfoca en incrementar la base de datos en general, desde sus clases que la componen y con otros detalles técnicos, donde también se realiza una limpieza de datos: donde prácticamente consiste en rellenar valores faltantes de la base de datos utilizando técnicas de imputación de datos, estandarización y normalización y por supuesto la remoción de valores inusuales o que estos no impacten significativamente al resultado final. De igual forma la ingeniería de características que sirve para la descomposición de atributos, transformaciones (logarítmicas, potencias etc.), generar atributos compuestos con datos correlaciones y no correlacionados con el fin de hacerle más fácil a los algoritmos su clasificación.

Por lo tanto, de acuerdo con la estructura de la base de datos de este estudio se considera pertinente la omisión del procesamiento, ya que, la BD no omite ningún valor de sus atributos en ninguno de los casos, las etiquetas son establecidas numéricamente antes de ser cargadas al notebook por lo que no se necesita de la descomposición y transformación de las clases.

4.6. Elección algorítmica

En esta fase se lleva a cabo la prueba de diferentes algoritmos de machine learning que se ajusten a la BD. Como se vio anteriormente, se trata de un problema de clasificación y se trata de una base de datos desbalanceada, es decir, que el número de casos de alguna clase predomina en comparación de las demás clases. Sin embargo, no es necesario buscar la mejor solución en todo el espectro de modelos, sino más bien, en las que mejor se ajustan a la BD. Por lo tanto, se prueba con algoritmos de clasificación y que además son aptos para base de datos desbalanceadas o desequilibradas, así como también las métricas de desempeño adecuadas para este tipo de problema.

4.6.1 Prueba de algoritmos

Los algoritmos de clasificación para datos desbalanceados que se probaron para este estudio son los siguientes:

- Support Vector Machine (SVC)
- Naive Bayes
- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Decision Tree
- Random Forest
- Gradient Boosting
- XGBoost
- K Nearest Neighbors (KNN)
- Neural Networks

4.6.2 Métricas de desempeño

Después de haber cargado la base de datos y ejecutado la codificación correspondiente para el entrenamiento y prueba de los algoritmos seleccionados, se requiere evaluar el desempeño o bien la exactitud de clasificación obtenida por los diferentes modelos probados. Las métricas seleccionadas que se presentan a continuación son las más comunes y rigurosos para evaluar modelos de clasificación multiclase, además cada una de ellas tiene un objetivo específico de análisis.

- Accuracy score (Exactitud)
- Balanced accuracy (Exactitud balanceada)
- Coeficiente de correlación de Matthews
- Kapa de Cohen
- Coeficiente de Jaccard
- Precisión
- Recall
- F1 score
- Matriz de confusión
- Curva ROC
- Curva PR

Además, todos los algoritmos se prueban con los siguientes parámetros de entrenamiento y prueba:

test size = 0.2; train size = 0.80; random state = 42

4.6.3 Evaluación del algoritmo Quadratic Discriminant Analysis

Para este algoritmo se obtuvo una exactitud resultante de 0.71 que representa la proporción de predicciones correctas frente al total. Se sabe que la BD se encuentra desbalanceada debido a que

sus clases tienen diferentes tamaños. De este modo, la métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.64.

Al tratarse de una clasificación, es muy probable que existan tanto verdaderos positivos como falsos negativos, es por ello por lo que métricas como la precisión, recall y f1 score pueden medir eso. La precisión mide la proporción de verdaderos positivos entre los casos clasificados como positivos; el recall (sensibilidad o tasa de verdaderos positivos) mide la proporción de verdaderos positivos entre todos los casos positivos reales; mientras que el f1 score combina estas dos últimas en una sola métrica. Estas métricas juntas brindan un estado de resultados donde presenta cada una de las métricas para cada clase. El resultado obtenido para cada clase se presenta en la tabla

Tabla 4.5. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo QDA.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.82	0.5	0.73	0.73	0.78	0	0.3	0.67	0.81	0.65
Recall	0.64	0.67	0.79	0.79	0.82	0	0.75	0.5	0.76	0.79
F1 score	0.72	0.57	0.76	0.76	0.8	0	0.43	0.57	0.79	0.71

4.6.3.1. Matriz de confusión

La matriz de confusión es una métrica de evaluación muy utilizada en problemas de clasificación multiclase, debido a que en esta se proporciona una visión detallada del rendimiento del modelo para cada clase.

En la Figura 4.8 se muestra la matriz de confusión obtenida por esta métrica. En donde, la diagonal principal (de izquierda a derecha) representa los casos correctamente clasificados; fuera de la diagonal principal, los elementos representan los casos incorrectamente clasificados. La fila correspondiente a la clase real indica los casos verdaderos para esa clase, mientras que la columna correspondiente a la clase predicha indica cómo se clasificaron esos casos.

En la Figura 4.9 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

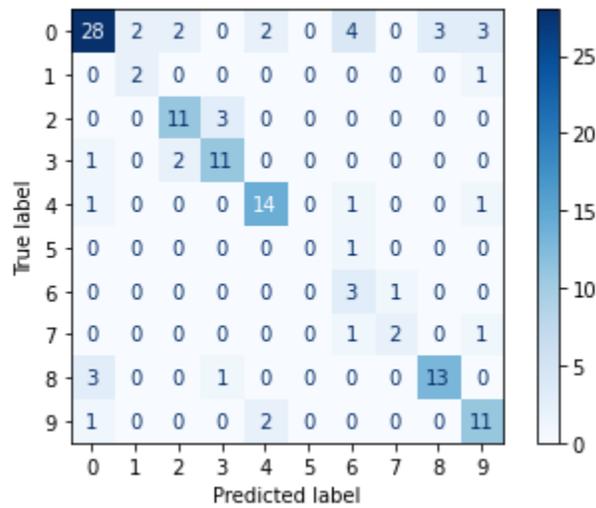


Figura 4.8. Matriz de confusión del algoritmo Quadratic Discriminant Analysis

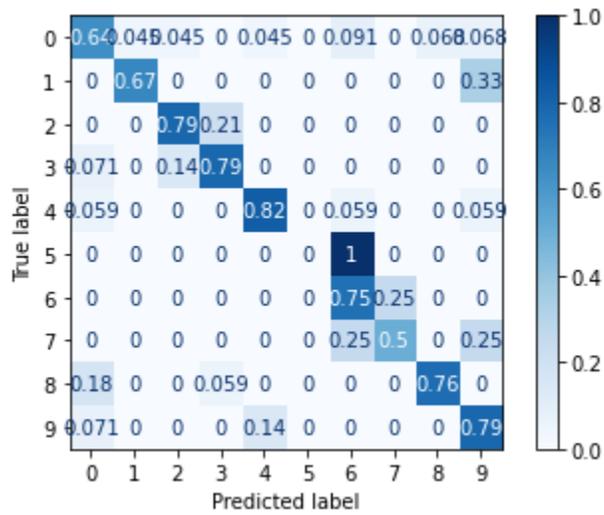


Figura 4.9. Matriz de confusión normalizada del algoritmo Quadratic Discriminant Analysis

4.6.3.2. Curva ROC

La curva ROC (Receiver Operating Characteristic) es una representación gráfica del rendimiento de un modelo de clasificación en función de diversos umbrales de decisión. En la Figura 4.10 se presenta el gráfico de la Curva ROC del modelo obtenido por el algoritmo Quadratic Discriminant Analysis. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y

representa la tasa de verdaderos positivos. El punto más cercano al vértice superior izquierdo de la curva representa el punto de operación óptimo del modelo. Este punto tiene una alta sensibilidad y una baja tasa de falsos positivos. El área bajo la curva (AUC-ROC) mide la capacidad del modelo para distinguir entre clases positivas y negativas. Un AUC-ROC más alto indica un mejor rendimiento del modelo. Por otro lado, la diagonal marcada desde el punto (0,0) al (1,1) representa la curva de un clasificador aleatorio, es decir, para que un modelo de clasificación se considere bueno debería estar por encima de esta diagonal.

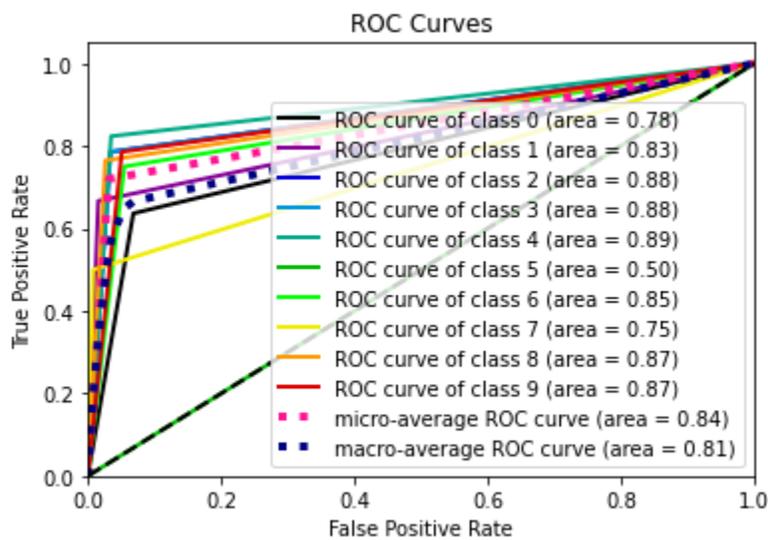


Figura 4.10. Curva ROC del modelo obtenido por el algoritmo Quadratic Discriminant Analysis

4.6.3.3. Curva PR

La Curva PR (Precision-Recall) es otra métrica de evaluación de modelos. En la Figura 4.11 se presenta el gráfico de la Curva PR del modelo obtenido por el algoritmo Quadratic Discriminant Analysis. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión. El área bajo la curva mide la capacidad del modelo para equilibrar precisión y recuperación. Al igual que con la curva ROC, un modelo ideal tendría una curva PR que se elevara rápidamente hacia el vértice superior derecho.

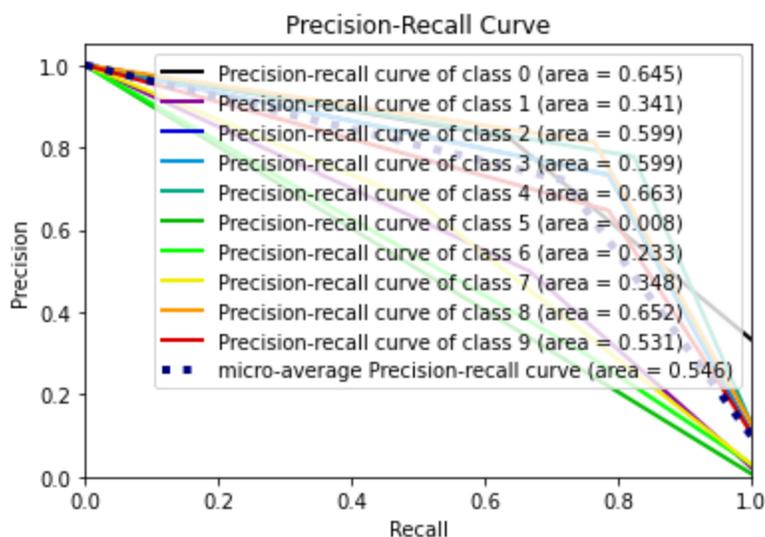


Figura 4.11. Curva PR del modelo obtenido por el algoritmo Quadratic Discriminant Analysis

4.6.4 Evaluación del algoritmo Gradient Boosting

Para este algoritmo se probó con los hiperparámetros: `max_depth = 5`, `n_estimators = 50` y `random_state = 42` se obtuvo una exactitud resultante de 0.82 que representa la proporción de predicciones correctas frente al total. La métrica `Balanced Accuracy` tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.63.

Las métricas como la precisión, recall y f1 score para este modelo se muestran en la Tabla 4.6.

Tabla 4.6. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo GB.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.84	1	0.86	0.81	0.76	0	0.6	1	0.84	0.9
Recall	0.93	0.33	0.86	0.93	0.76	0	0.75	0.25	0.94	0.64
F1 score	0.88	0.5	0.86	0.87	0.76	0	0.67	0.4	0.89	0.75

4.6.4.1. Matriz de confusión

En la Figura 4.12 se muestra la matriz de confusión obtenida por esta métrica. En la Figura 4.13 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

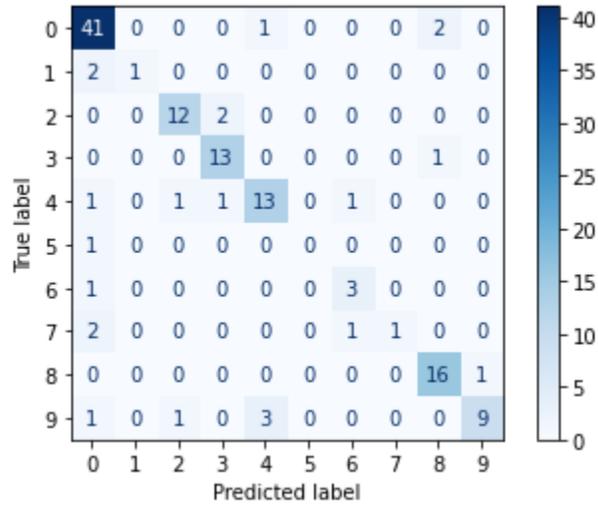


Figura 4.12. Matriz de confusión del modelo del algoritmo Gradient Boosting

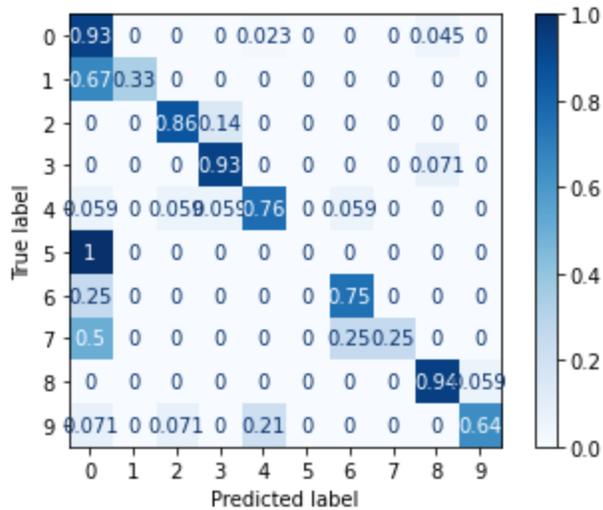


Figura 4.13. Matriz de confusión normalizada del modelo del algoritmo Gradient Boosting

4.6.4.2. Curva ROC

En la Figura 4.14 se presenta el gráfico de la Curva ROC del modelo obtenido por el algoritmo Gradient Boosting. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

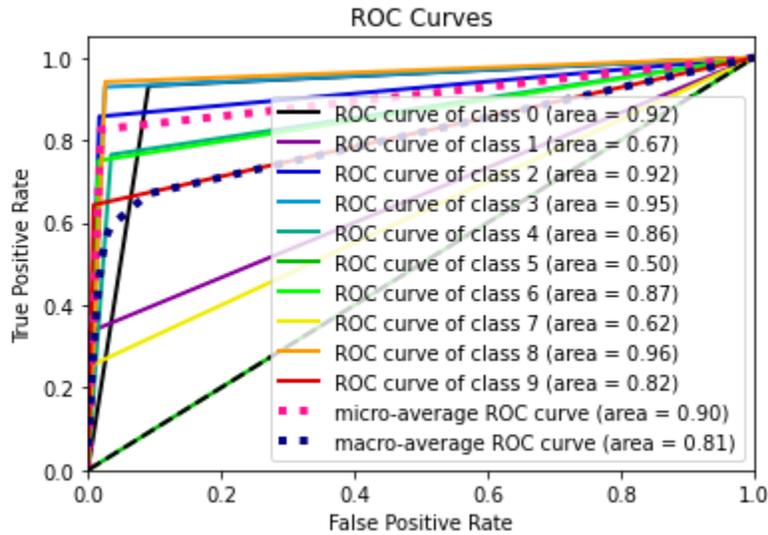


Figura 4.14. Curva ROC del modelo obtenido por el algoritmo Gradient Boosting

4.6.4.3. Curva PR

En la Figura 4.15 se presenta el gráfico de la Curva PR del modelo obtenido por el algoritmo Gradient Boosting. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

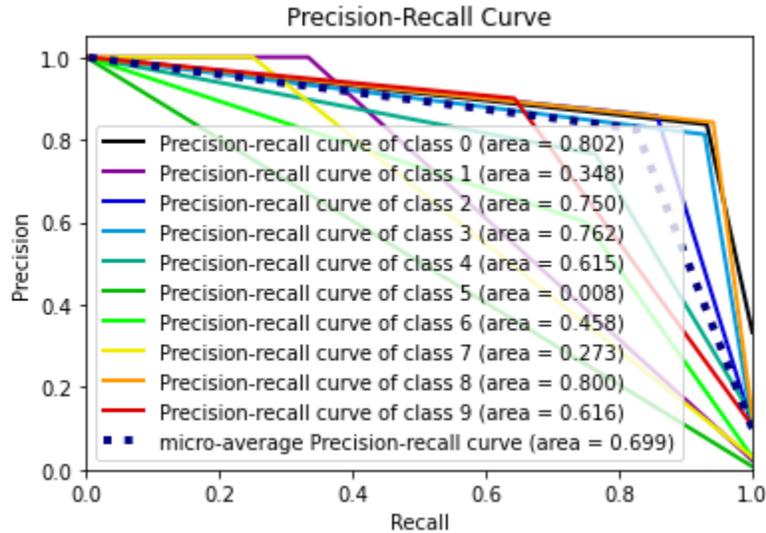


Figura 4.15. Curva PR del modelo obtenido por el algoritmo Gradient Boosting

4.6.5 Evaluación del algoritmo K-Nearest Neighbors

Para este algoritmo se probó con $n_neighbors = 10$ se obtuvo una exactitud resultante de 0.75 que representa la proporción de predicciones correctas frente al total. La métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.67.

Las métricas como la precisión, recall y f1 score para este modelo se muestran en la Tabla 4.7

Tabla 4.7. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo KNN.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precisión	0.74	0.5	0.9	0.92	0.88	0	0.43	0.6	0.75	0.73
Recall	0.77	0.67	0.64	0.86	0.82	0	0.75	0.75	0.88	0.57
F1 score	0.76	0.57	0.75	0.89	0.85	0	0.55	0.67	0.81	0.064

4.6.5.1. Matriz de confusión

En la Figura 4.16 se muestra la matriz de confusión obtenida por esta métrica. En la Figura 4.17 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

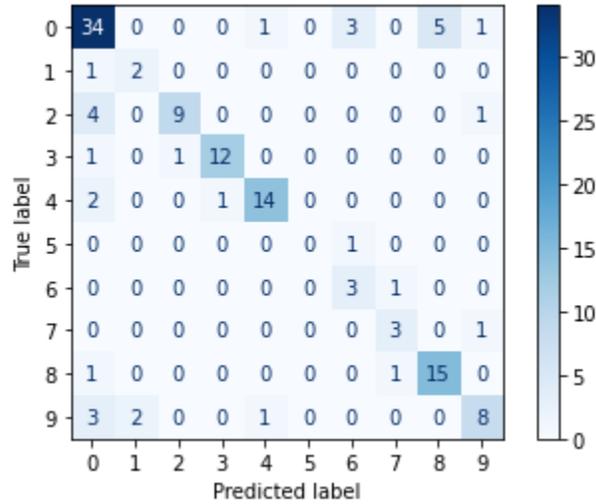


Figura 4.16. Matriz de confusión del modelo del algoritmo K-Nearest Neighbors

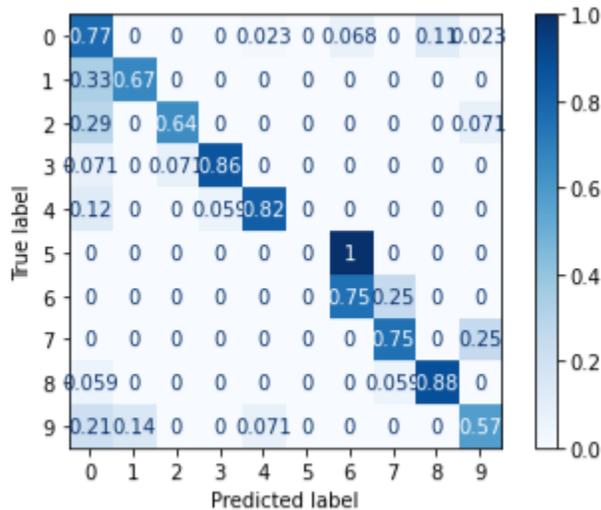


Figura 4.17. Matriz de confusión normalizada del modelo del algoritmo K-Nearest Neighbors

4.6.5.2. Curva ROC

En la Figura 4.18 se presenta el gráfico de la Curva ROC del modelo obtenido por el algoritmo K-Nearest Neighbors. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

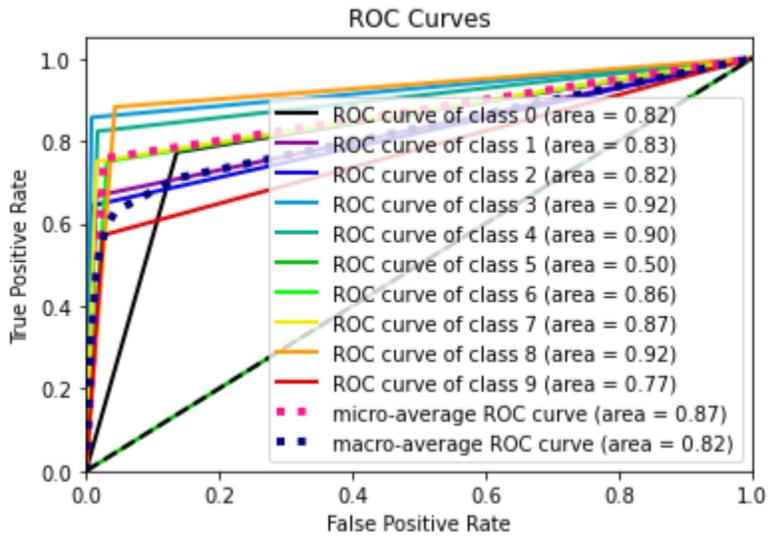


Figura 4.18. Curva ROC del modelo obtenido por el algoritmo K-Nearest Neighbors

4.6.5.3. Curva PR

En la Figura 4.19 se presenta el gráfico de la Curva PR del modelo obtenido por el algoritmo K-Nearest Neighbors. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

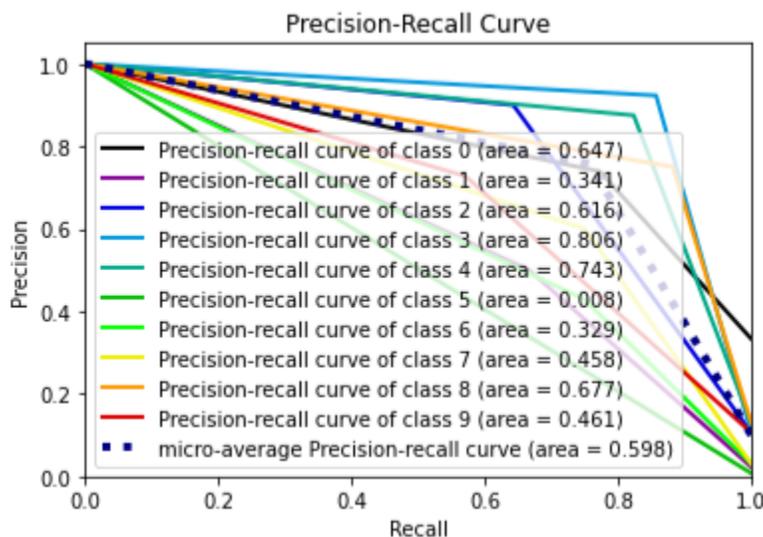


Figura 4.19. Curva PR del modelo obtenido por el algoritmo K-Nearest Neighbors

4.6.6 Evaluación del algoritmo XGBoost

Para este algoritmo se probó con $\text{max_depth} = 5$ y $\text{n_estimators} = 50$ se obtuvo una exactitud resultante de 0.84 que representa la proporción de predicciones correctas frente al total. La métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.66.

Las métricas como la precisión, recall y f1 score para este modelo se muestran en la Tabla 4.8

Tabla 4.8. Reporte de clasificación de las métricas precisión, recall y f1 score para el algoritmo XGB.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.89	1	0.92	0.8	0.82	0	0.5	0.5	0.84	0.92
Recall	0.93	0.33	0.86	0.86	0.82	0	0.75	0.25	0.94	0.86
F1 score	0.91	0.5	0.89	0.83	0.82	0	0.6	0.33	0.89	0.89

4.6.6.1. Matriz de confusión

En la Figura 4.20 se muestra la matriz de confusión obtenida por esta métrica. En la Figura 4.21 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

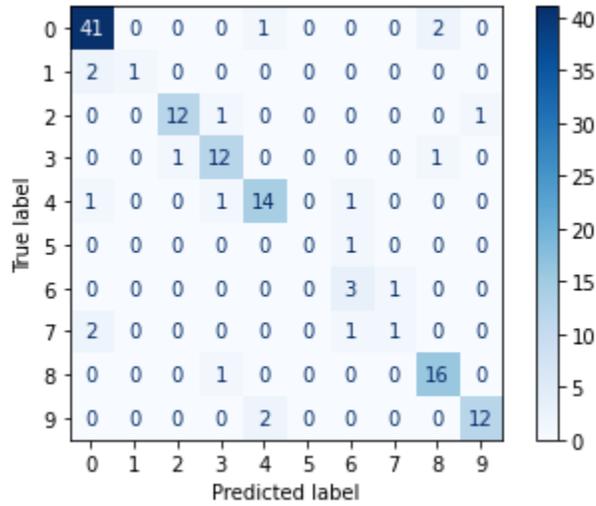


Figura 4.20. Matriz de confusión del modelo del algoritmo XGBoost

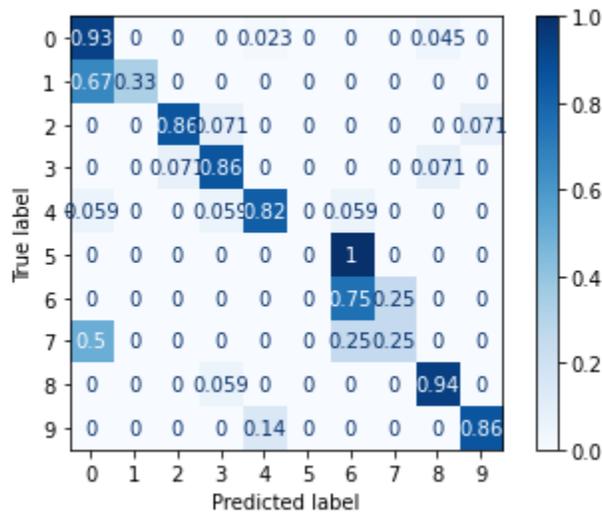


Figura 4.21. Matriz de confusión normalizada del modelo del algoritmo XGBoost

4.6.6.2. Curva ROC

En la Figura 4.22 se presenta el gráfico de la Curva ROC del modelo obtenido por el algoritmo XGBoost. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

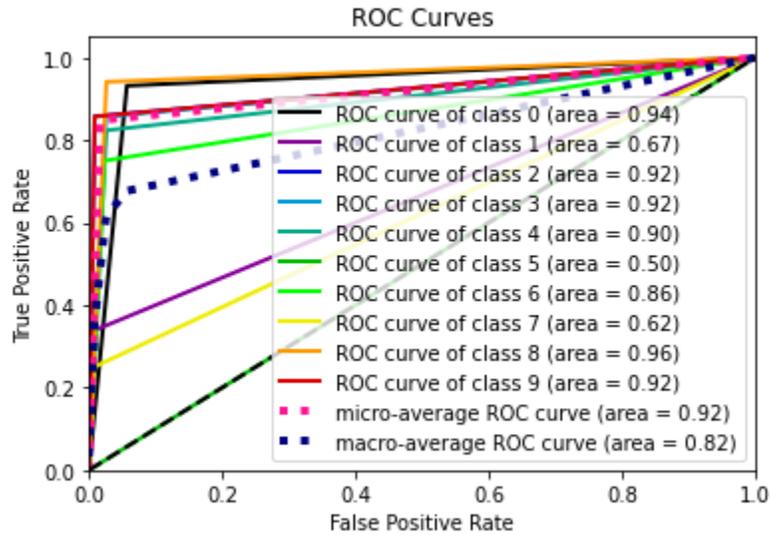


Figura 4.22. Curva ROC del modelo obtenido por el algoritmo XGBoost

4.6.6.3. Curva PR

En la Figura 4.23 se presenta el gráfico de la Curva PR del modelo obtenido por el algoritmo XGBoost, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

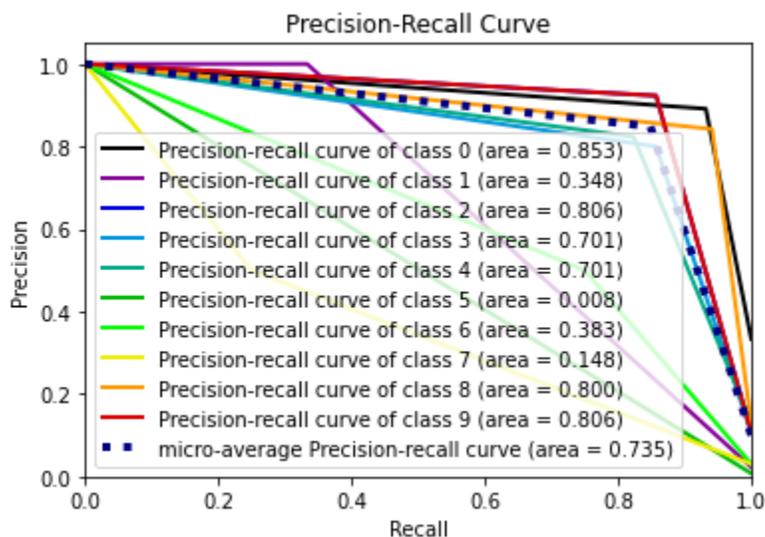


Figura 4.23. Curva PR del modelo obtenido por el algoritmo XGBoost

4.6.7 Elección para la optimización

Las evaluaciones presentadas anteriormente fueron de las mejores que se obtuvieron de los 10 algoritmos probados (ver resto de algoritmos y métricas en el capítulo de resultados). Dentro de estos, el que tuvo la mejor exactitud es el algoritmo XGboost, con 0.84 de exactitud. Sin embargo, esta métrica no considera una exactitud equitativa al tratarse un problema de clasificación multiclase, por otro lado, la métrica de exactitud balanceada si, la cual obtuvo un 0.66 de exactitud. Por otra parte, el algoritmo de K-Nearest Neighbors no fue el que obtuvo la mayor exactitud, siendo esta de 0.75, sin embargo, fue el que mayor exactitud balanceada obtuvo siendo esta igual con 0.67 y superando por poco al algoritmo XGBoost.

En la Tabla 4.9 se presentan los resultados obtenidos de las métricas de exactitud y exactitud balanceada, al igual que los coeficientes de Matthews, Jaccard y Capa de Cohen que no fueron presentados anteriormente pero que miden una exactitud general sin poner énfasis en la clasificación multiclase.

Tabla 4.9. Métricas de desempeño de algoritmos seleccionados.

Algoritmo	Accuracy score	Balanced Accuracy	CC Matthews	Kappa de Cohen	Coef. de Jaccard
Quadratic Discriminant Analysis	0.71	0.64	0.669	0.66	0.56
Gradient Boosting Classifier	0.82	0.63	0.786	0.784	0.703
K-Nearest Neighbors	0.75	0.67	0.704	0.703	0.609
XGBoost	0.84	0.66	0.814	0.813	0.736

Por lo tanto, se decide optimizar ambos modelos: K-Nearest Neighbors y XGBoost, y posteriormente evaluarlos con las mismas métricas y ver cuál de estos es el que mejor exactitud balanceada proporciona. Finalmente, el mejor será probado con los datos que se apartaron previamente de la base de datos, y que no fueron introducidos en la parte de entrenamiento y prueba del algoritmo. Estos últimos resultados se presentan en el capítulo de resultados.

4.6.8 Optimización del mejor modelo

Para la optimización de los modelos seleccionados se requiere encontrar la combinación de los mejores hiperparámetros. Como se vio anteriormente, los hiperparámetros son configuraciones externas al modelo que afectan su rendimiento y que se deben ajustar para optimizar el rendimiento. En Python, existen técnicas como GridSearchCV, la cual realiza una búsqueda exhaustiva de un conjunto predefinido de hiperparámetros, en lugar de ajustarlos manualmente. Esta técnica prueba todas las combinaciones posibles de valores de hiperparámetros utilizando validación cruzada para evaluar el rendimiento del modelo dentro de un rango especificado.

4.6.8.1 Optimización de K-Nearest Neighbors

La optimización dio como resultado que los mejores hiperparámetros para este algoritmo son:
parameters = {metric = euclidean, n_neighbors = 3}

Por lo tanto, una vez encontrado los hiperparámetros de optimización se procede a probar el algoritmo y evaluarlo con las métricas correspondientes. Este algoritmo optimizado obtuvo una

exactitud resultante de 0.77 que representa la proporción de predicciones correctas frente al total. La métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.70.

Las métricas como la precisión, recall y f1 score para este modelo se muestran en la Tabla 4.10

Tabla 4.10. Reporte de clasificación de las métricas precisión, recall y f1 score de optimización KNN.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.71	1	1	0.81	0.82	0	0.6	0.6	0.83	0.8
Recall	0.82	1	0.5	0.93	0.82	0	0.75	0.75	0.88	0.57
F1 score	0.76	1	0.67	0.87	0.82	0	0.67	0.67	0.86	0.67

4.6.8.1.1. Matriz de confusión

En la Figura 4.24 se muestra la matriz de confusión obtenida por esta métrica. En la Figura 4.25 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

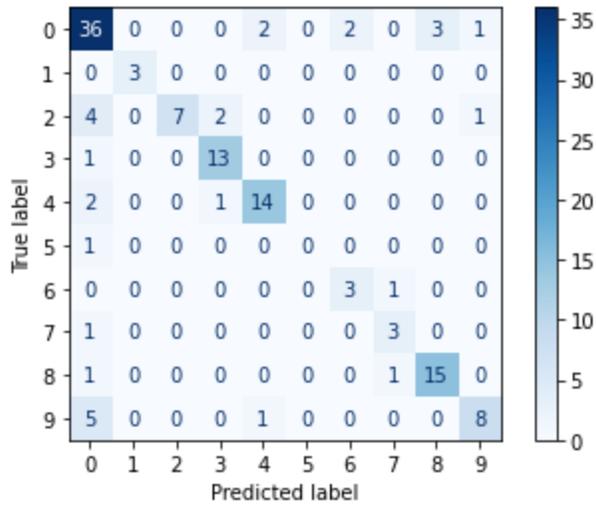


Figura 4.24. Matriz de confusión del modelo optimizado del algoritmo K-Nearest Neighbors

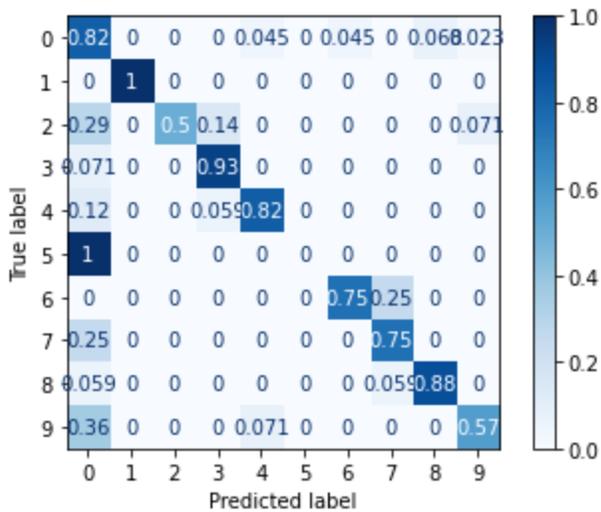


Figura 4.25. Matriz de confusión normalizada del modelo optimizado del algoritmo K-Nearest Neighbors

4.6.8.1.2. Curva ROC

En la Figura 4.26 se presenta el gráfico de la Curva ROC del modelo optimizado obtenido por el algoritmo K-Nearest Neighbors. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

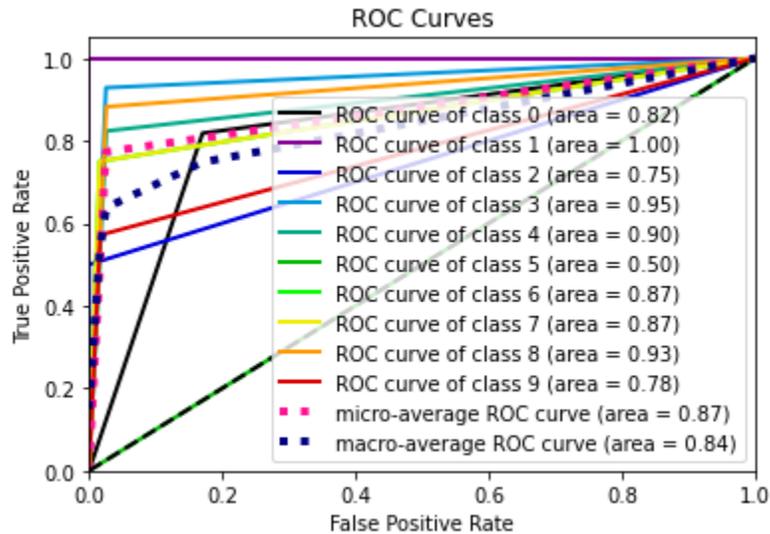


Figura 4.26. Curva ROC del modelo optimizado obtenido por el algoritmo K-Nearest Neighbors

4.6.8.1.3. Curva PR

En la Figura 4.27 se presenta el gráfico de la Curva PR del modelo obtenido por el algoritmo K-Nearest Neighbors. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

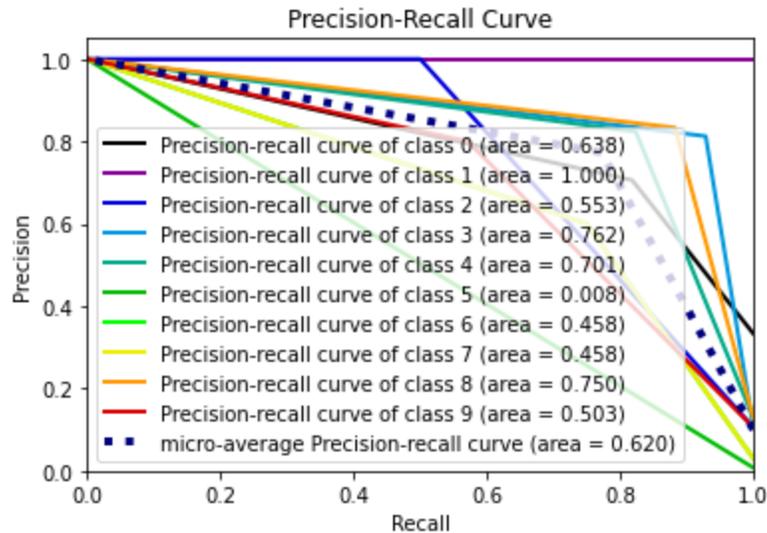


Figura 4.27. Curva PR del modelo optimizado obtenido por el algoritmo K-Nearest Neighbors

4.6.8.2 Optimización de XGBoost

La optimización dio como resultado que los mejores hiperparámetros para este algoritmo son: `parameters = {max_depth = none, n_estimators = 722}`

Por lo tanto, una vez encontrado los hiperparámetros de optimización se procede a probar el algoritmo y evaluarlo con las métricas correspondientes. Este algoritmo optimizado obtuvo una exactitud resultante de 0.84 que representa la proporción de predicciones correctas frente al total. La métrica *Balanced Accuracy* tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.65.

Las métricas como la precisión, recall y f1 score para este modelo se muestran en la Tabla 4.11

Tabla 4.11. Reporte de clasificación de las métricas precisión, recall y f1 score de optimización XGB.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.87	1	0.92	0.8	0.82	0	0.5	0.33	0.89	0.92
Recall	0.93	0.33	0.86	0.86	0.82	0	0.75	0.25	0.94	0.79
F1 score	0.9	0.5	0.89	0.83	0.82	0	0.6	0.29	0.91	0.85

4.6.8.2.1. Matriz de confusión

En la Figura 4.28 se muestra la matriz de confusión obtenida por esta métrica. En la Figura 4.29 se presenta la matriz de confusión normalizada, donde se presenta la proporción de las predicciones del modelo.

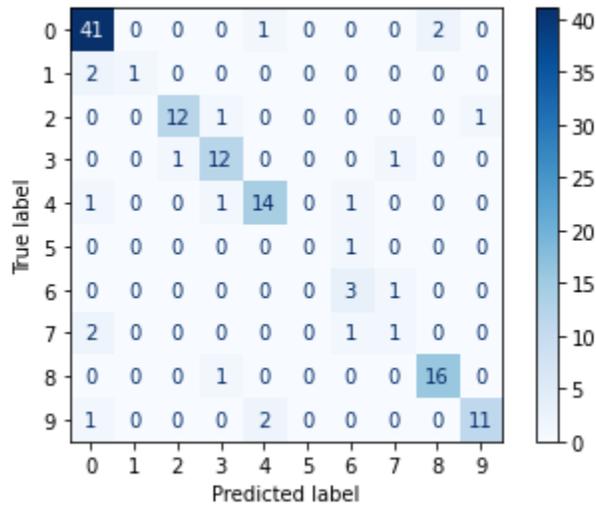


Figura 4.28. Matriz de confusión del modelo optimizado del algoritmo XGBoost

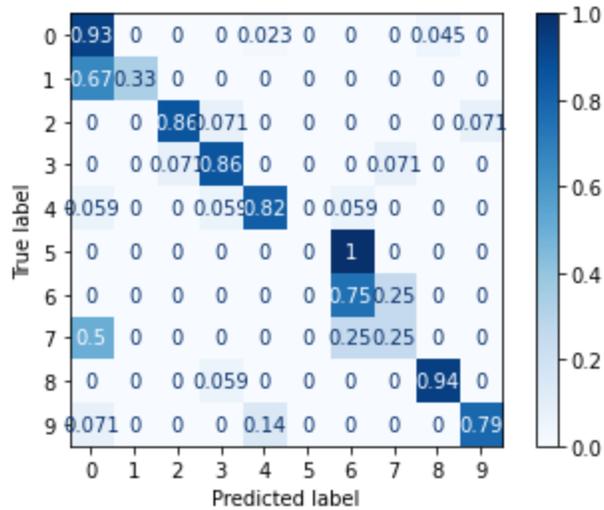


Figura 4.29. Matriz de confusión normalizada del modelo optimizado del algoritmo XGBoost

4.6.8.1.2. Curva ROC

En la Figura 4.30 se presenta el gráfico de la Curva ROC del modelo optimizado obtenido por el algoritmo XGBoost. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

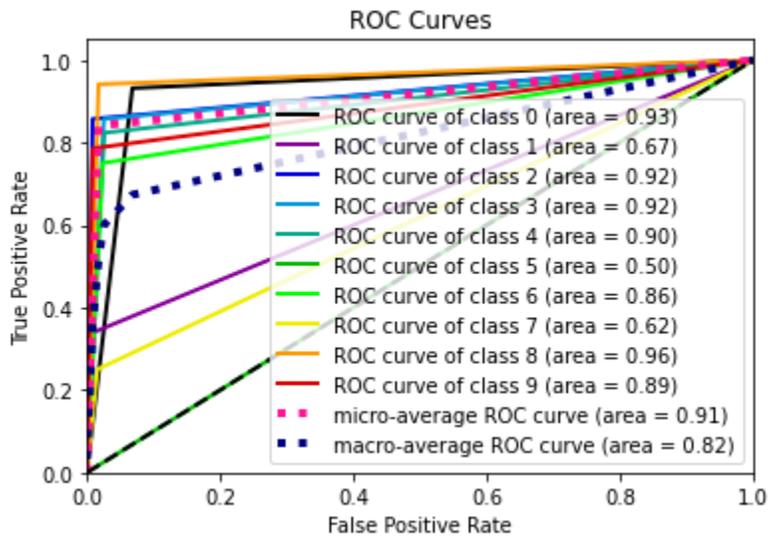


Figura 4.30. Curva ROC del modelo optimizado obtenido por el algoritmo XGBoost

4.6.8.2.3. Curva PR

En la Figura 4.31 se presenta el gráfico de la Curva PR del modelo optimizado obtenido por el algoritmo XGBoost. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

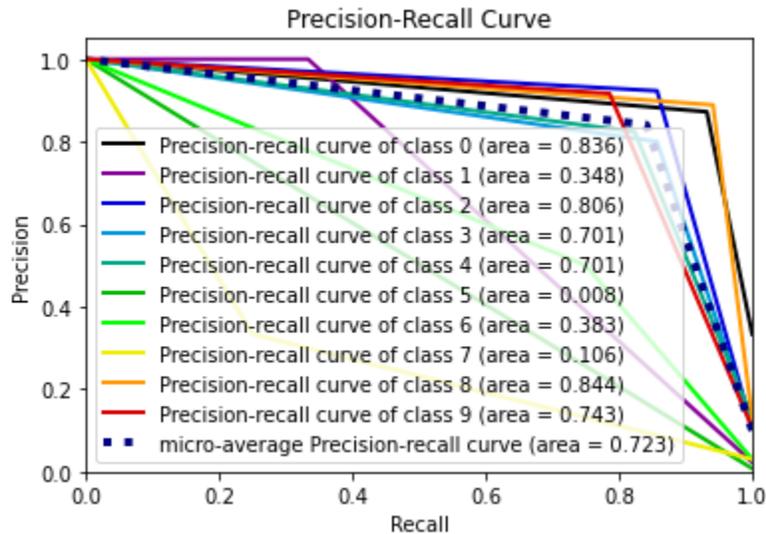


Figura 4.31. Curva PR del modelo optimizado obtenido por el algoritmo XGBoost

Después de haber optimizado los mejores modelos, se lleva a cabo la prueba de las predicciones con el 10% de los datos. Los alcances obtenidos se presentan en el capítulo de resultados.

4.7. Modelación Lógica Difusa

De acuerdo con la revisión del estado del arte, se han desarrollado otras técnicas basadas en relaciones de concentración de gases, entre ellas, la lógica difusa para el diagnóstico de fallas en transformadores. Estas técnicas son de naturaleza heurística y pueden variar de una utilidad a otra. El diagnóstico confiable de fallas utilizando los métodos de diagnóstico enumerados anteriormente, a veces resulta difícil, si no imposible, particularmente cuando dos o más fallas están presentes simultáneamente en un transformador. En consecuencia, existe la necesidad de desarrollar una técnica que sea capaz de determinar el índice de salud de los transformadores.

Este trabajo de investigación propone un modelo global que tiene como fundamento dos modelos de lógica difusa para determinar el índice de salud del aceite de transformadores, donde se utilizan las concentraciones de los cinco gases disueltos en el aceite del transformador para proporcionar una evaluación de este mismo mediante fallas. Este modelo propuesto en la revisión literaria ha demostrado ser más conveniente que los métodos de interpretación de DGA convencionales como se muestra en el estado del arte.

En esta sección se detalla la metodología para llevar a cabo los modelos de simulación de LD propuestos. Para el caso de estudio, los modelos de LD se desarrollan bajo una estructura de un modelo de tipo Mamdani. La esquematización de este modelo se presenta en la Figura 4.32.

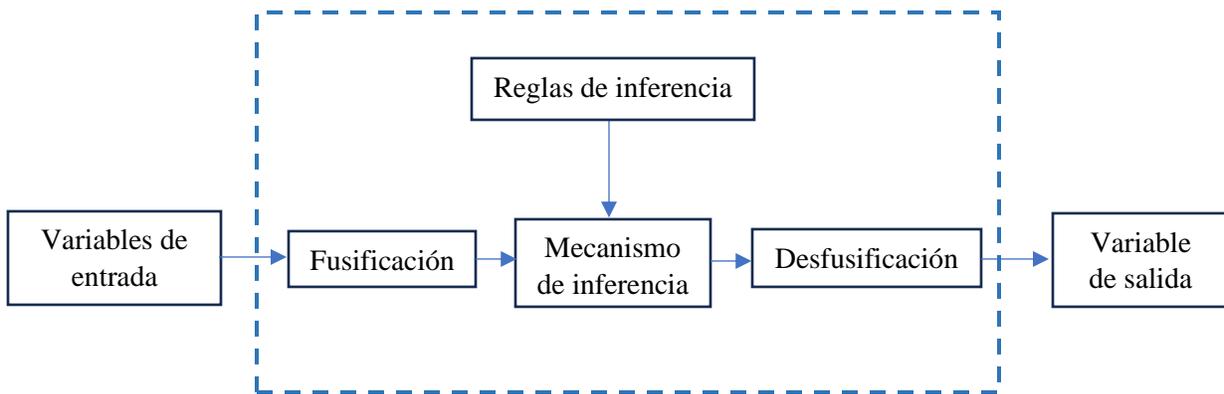


Figura 4.32. Esquematización del modelo difuso tipo mamdani

Los modelos de lógica difusa permiten representar el sistema de estudio considerando las variables de entrada y salida a través de conjuntos difusos representados por términos lingüísticos. El modelo se desarrolla en el software Matlab con la herramienta Fuzzy, la cual es la interfaz que permite la creación del modelo inteligente.

Sin embargo, haciendo la revisión del estado del arte, existen investigaciones con variación en los intervalos, unos dan más y otros dan menos intervalos. En otros estudios utilizan las relaciones de los gases como lo son el método de Rogers o Dornenburg. Por lo tanto, para esta investigación se proponen dos modelos de LD: el primero se trata de un modelo de LD binario, donde el modelo sea capaz de clasificar si el transformador se encuentra en una condición normal o bien si este sufre de una falla sin especificar de qué tipo de falla se trate; el segundo modelo se trata de una

salida multiclase, donde por medio del primer modelo se corrobora que el transformador presenta una falla y por lo tanto se requiere saber a qué tipo de falla corresponde.

4.7.1. Modelo de lógica difusa de clasificación binaria

Este modelo se desarrolla con las concentraciones puras de los diferentes gases, es decir, con las concentraciones en partes por millón tal y como fue proporcionada la base de datos. Se pretende analizar las setecientas cuarenta y una muestras de aceite de transformadores recolectadas por el Instituto Nacional de Electricidad y Energías Limpias (INEEL).

El análisis de gases disueltos es el método más utilizado para determinar el índice de salud del aceite del transformador y para identificar fallas incipientes presentes en los transformadores. El DGA utiliza las concentraciones de varios gases disueltos. De acuerdo con el estándar IEEE57.104-2008, las concentraciones de gas se dividen en cuatro rangos, como se muestra en la Tabla 4.12, donde los rangos 1, 2, 3 y 4 indican una salud del aceite excelente, buena, mala y muy mala respectivamente. Estos rangos se utilizan para crear los conjuntos difusos del primer modelo.

Tabla 4.12. Concentraciones de gas (en ppm) del estándar IEEE

Gas	Número de rango			
	1	2	3	4
H ₂	0 - 100	101 - 700	701 - 1800	> 1800
CH ₄	0 - 120	121 - 400	401 - 1000	> 1000
C ₂ H ₂	0 - 1	2 - 9	10 - 35	> 35
C ₂ H ₄	0 - 50	51 - 100	101 - 200	> 200
C ₂ H ₆	0 - 65	66 - 100	101 - 150	> 150

4.7.1.1 Establecimiento de los parámetros de las variables de entrada

En el apartado *Obtención de la base de datos* se muestran las variables de entrada, así como también la variable de salida multiclase.

Siguiendo con el siguiente paso para el desarrollo del modelo de LD, de los cinco gases disueltos enumerados anteriormente en cuatro grupos (o conjuntos borrosos/difusos) se establecen en orden de 1 a 4 respectivamente como: concentración baja, media, alta y muy alta.

Una función de pertenencia o por sus siglas en inglés (Membership Function) es una curva que define como una determinada concentración de gas disuelto se asigna a un grado de pertenencia, entre 0 y 1, de cualquiera de los cuatro conjuntos difusos. Una MF puede tomar varias formas, por ejemplo, triangular, gaussiana, sigmoideal o trapezoidal como se puede observar en la Figura 4.33.

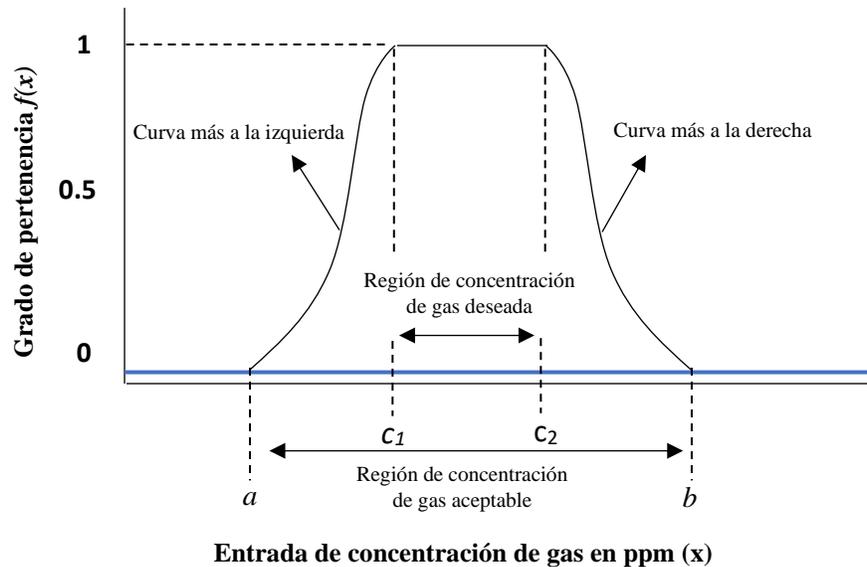


Figura 4.33. Ejemplo de una función de pertenencia de tipo gaussiana

Es así, que el resultado del modelo de lógica difusa del índice de salud del aceite cubre el rango de 0 a 1 y se dividen en cuatro funciones de membresía de tipo trapezoidal, ya que la MF trapezoidal permite establecer un valor de pertenencia constante igual a 1 en un rango definido y no solo en un valor puntual. De este modo, los parámetros para la modelación del modelo de lógica difusa de clasificación binaria se presentan en la Tabla 4.13.

Tabla 4.13. Parámetros de modelación de LD de las variables de entrada.

Variable de entrada	Variabes lingüísticas	Función de membresía	Intervalo
H₂	Baja	Trapezoidal	[0, 0, 100, 101]
	Media		[100, 101, 700, 701]
	Alta		[700, 701, 1800, 1801]
	Muy alta		[1800, 1801, 3000, 3000]
CH₄	Baja	Trapezoidal	[0, 0, 120, 121]
	Media		[120, 121, 400, 401]
	Alta		[400, 401, 1000, 1001]
	Muy alta		[1000, 1001, 3000, 3000]
C₂H₂	Baja	Trapezoidal	[0, 0, 1, 2]
	Media		[1, 2, 9, 10]
	Alta		[9, 10, 35, 36]
	Muy alta		[35, 36, 200, 200]
C₂H₄	Baja	Trapezoidal	[0, 0, 50, 51]
	Media		[50, 51, 100, 101]
	Alta		[100, 101, 200, 201]
	Muy alta		[199, 200, 1000, 1000]
C₂H₆	Baja	Trapezoidal	[0, 0, 65, 66]
	Media		[65, 66, 100, 101]
	Alta		[100, 101, 150, 151]
	Muy alta		[150, 151, 500, 500]

4.7.1.2 Establecimiento de los parámetros de la variable de salida

El establecimiento de la variable de salida también cubre el rango de 0 a 1 y se dividen en dos funciones de membresía de tipo triangular, ya que la MF triangular permite establecer un valor de pertenencia en un valor puntual. De este modo, los parámetros para la modelación de la variable de salida se presentan en la Tabla 4.14.

Tabla 4.14. Parámetros de modelación de LD de la variable de salida.

Variable de salida	Variables lingüísticas	Función de membresía	Intervalo
Diagnóstico	Condición normal	Triangular	[1, 1, 1]
	Falla		[3, 3, 3]

De este modo, dentro de los intervalos, el valor con pertenencia igual a 1 son: para la condición normal igual a 1 y para la falla igual a 3.

4.7.1.3 Reglas de inferencia

Para este primer modelo se obtienen 268 reglas de inferencia, las que posteriormente se alimentan al modelo en Matlab. Algunas de las reglas se presentan en la Tabla 4.15, y el resto se encuentran en el *Anexo 1*.

Tabla 4.15. Reglas de inferencia para el modelo de LD de clasificación binaria.

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Baja	Baja	Baja	Baja	Baja	1	Condición normal
Media	Media	Baja	Alta	Muy alta	3	Falla
Media	Alta	Alta	Muy alta	Muy alta	3	Falla
Baja	Media	Baja	Muy alta	Media	3	Falla
Baja	Baja	Media	Muy alta	Media	3	Falla
Baja	Media	Media	Muy alta	Media	3	Falla
Baja	Baja	Baja	Alta	Alta	3	Falla
Media	Muy alta	Baja	Muy alta	Muy alta	3	Falla
Baja	Baja	Baja	Muy alta	Muy alta	3	Falla
Alta	Alta	Muy alta	Alta	Muy alta	3	Falla
Muy alta	Alta	Alta	Muy alta	Muy alta	3	Falla
Baja	Baja	Muy alta	Muy alta	Alta	3	Falla
Baja	Alta	Alta	Alta	Media	3	Falla
Alta	Media	Muy alta	Media	Media	3	Falla
Media	Alta	Muy alta	Media	Alta	3	Falla
Muy alta	Muy alta	Baja	Baja	Muy alta	3	Falla
Muy alta	Muy alta	Media	Baja	Muy alta	3	Falla
Media	Baja	Muy alta	Muy alta	Baja	3	Falla
Muy alta	Muy alta	Baja	Muy alta	Muy alta	3	Falla
Alta	Muy alta	Alta	Muy alta	Muy alta	3	Falla
Muy alta	Muy alta	Media	Muy alta	Muy alta	3	Falla
Media	Alta	Alta	Muy alta	Muy alta	3	Falla
Media	Muy alta	Media	Muy alta	Muy alta	3	Falla
Muy alta	Media	Baja	Baja	Baja	3	Falla
Muy alta	Alta	Baja	Baja	Muy alta	3	Falla
Baja	Baja	Baja	Alta	Media	3	Falla

4.7.1.4 Modelación en Matlab

Una vez establecidas las variables lingüísticas y funciones de membresía se modelan en Matlab. Para ello, se introduce la palabra *Fuzzy* en la ventana de comandos, donde posteriormente se abrirá una nueva ventana de diseño con un modelo tipo mamdani de manera predeterminada.

La Figura 4.34 muestra la ventana de diseño de lógica difusa, en la cual se puede observar a la izquierda la opción para agregar las variables de entrada, en el centro el tipo mamdani y a la derecha la variable de salida.

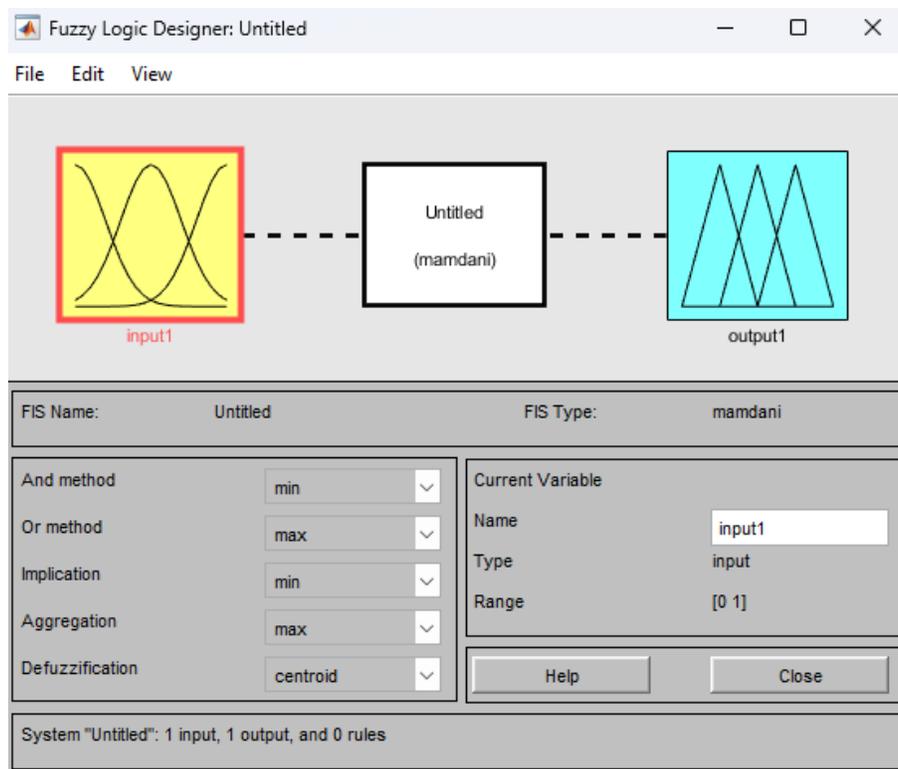


Figura 4.34. Ventana de diseño para modelos de Lógica Difusa

Una vez ya establecidas las variables de entrada y salida con sus respectivas variables lingüísticas y conjuntos difusos se procede a alimentar el modelo. En las Figuras 4.35-4.40 se muestra cómo se establece cada variable de entrada y salida con sus respectivos parámetros de modelación.

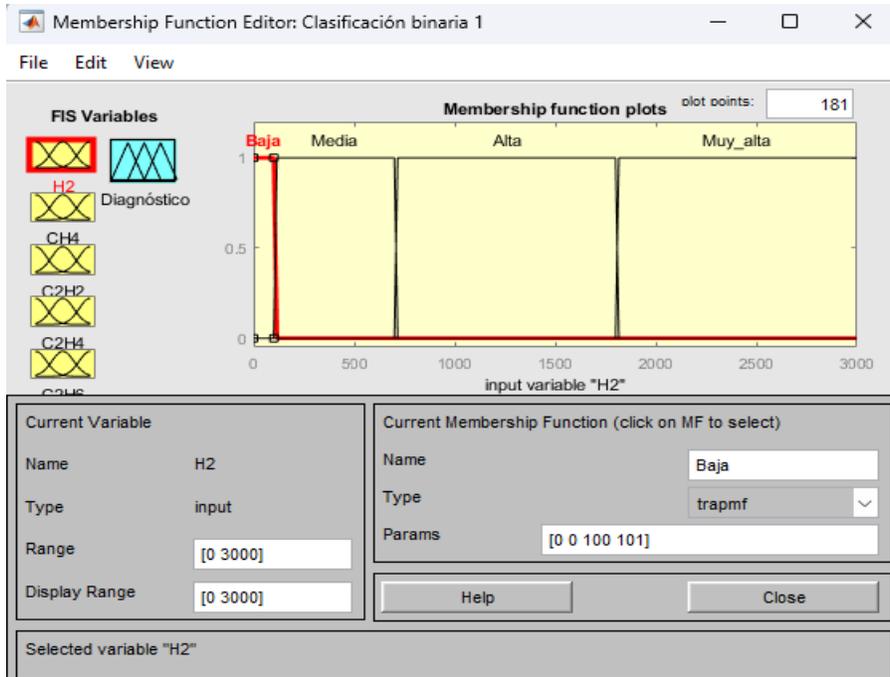


Figura 4.35. Parámetros de la variable de entrada H₂

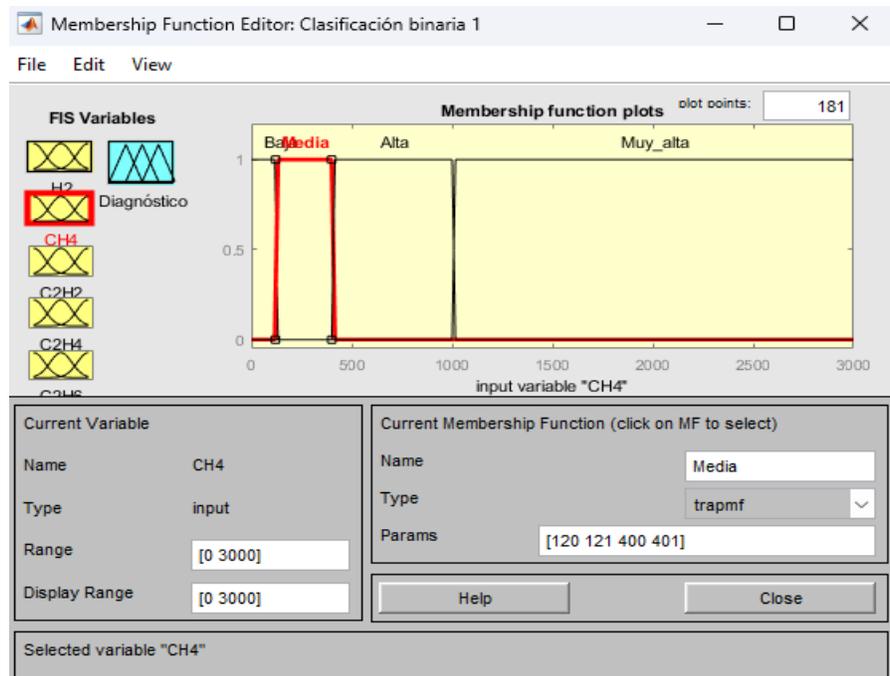


Figura 4.36. Parámetros de la variable de entrada CH₄

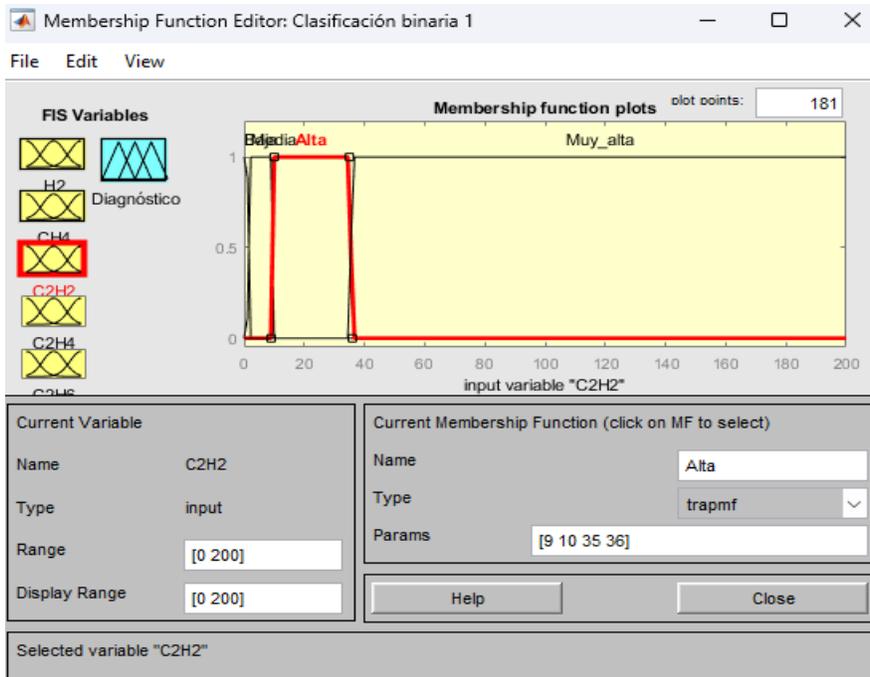


Figura 4.37. Parámetros de la variable de entrada C_2H_2

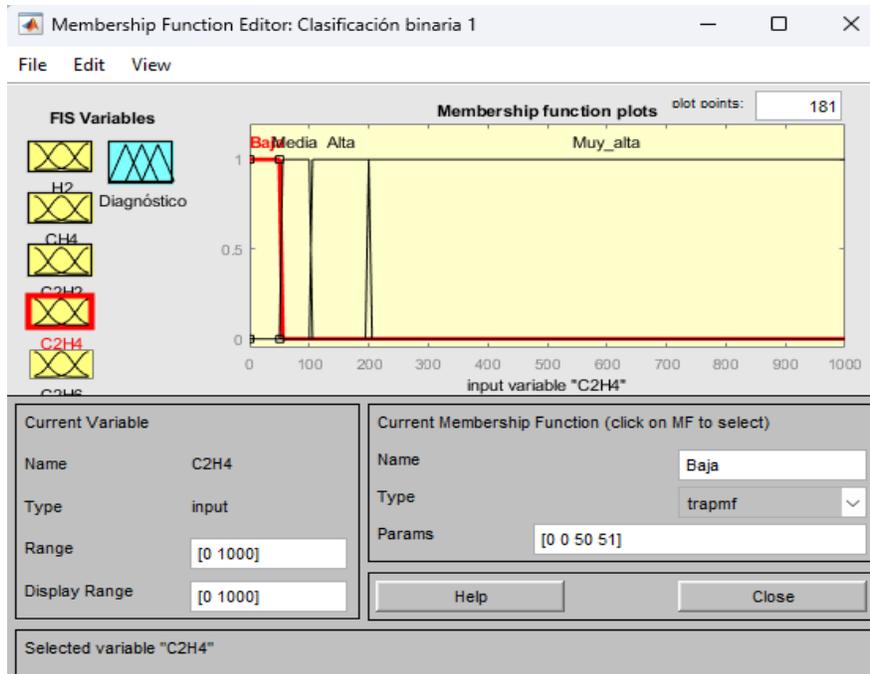


Figura 4.38. Parámetros de la variable de entrada C_2H_4

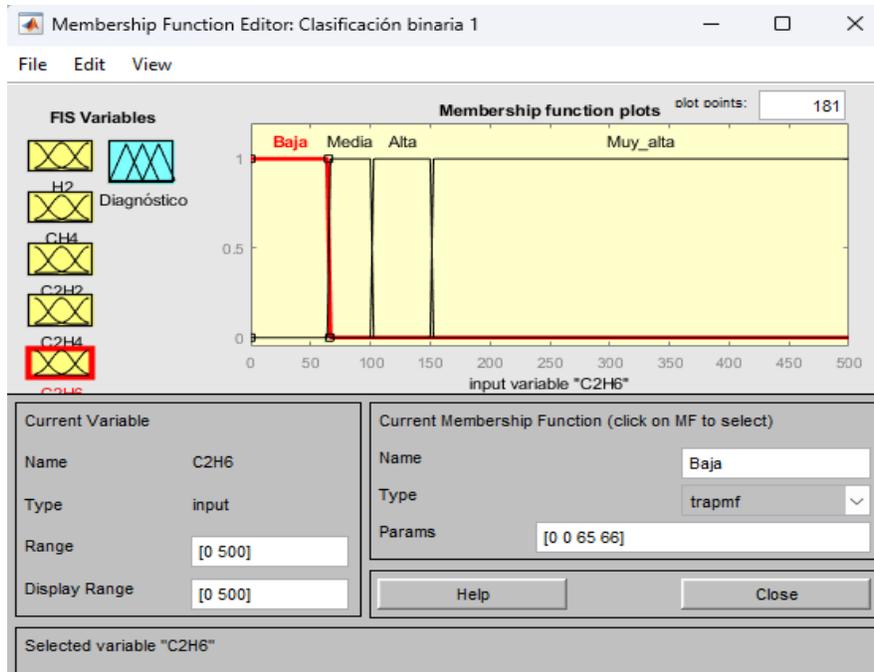


Figura 4.39. Parámetros de la variable de entrada C_2H_6

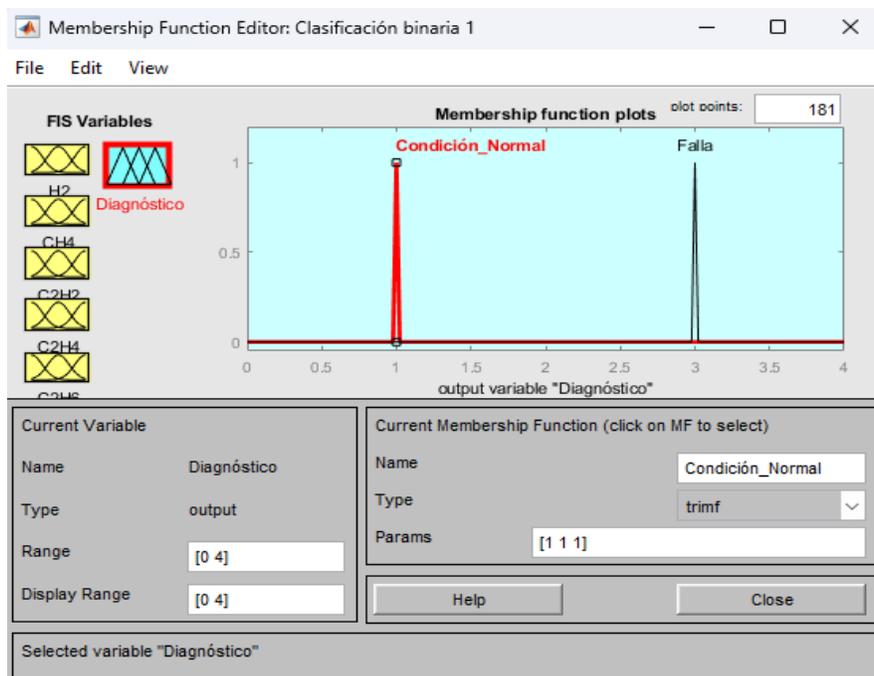


Figura 4.40. Parámetros de la variable de salida Diagnóstico

Una vez establecidas las reglas de inferencia se procede a alimentar el modelo en Matlab. La Figura 4.41 muestra el editor de reglas que a su vez permite ver el total y cómo están conformadas. Además, en la Figura 4.42 se muestra cómo se visualizan finalmente las reglas de inferencia en conjunto con las funciones de membresía, donde además se lleva a cabo la predicción de los casos introduciendo las variables de entrada.

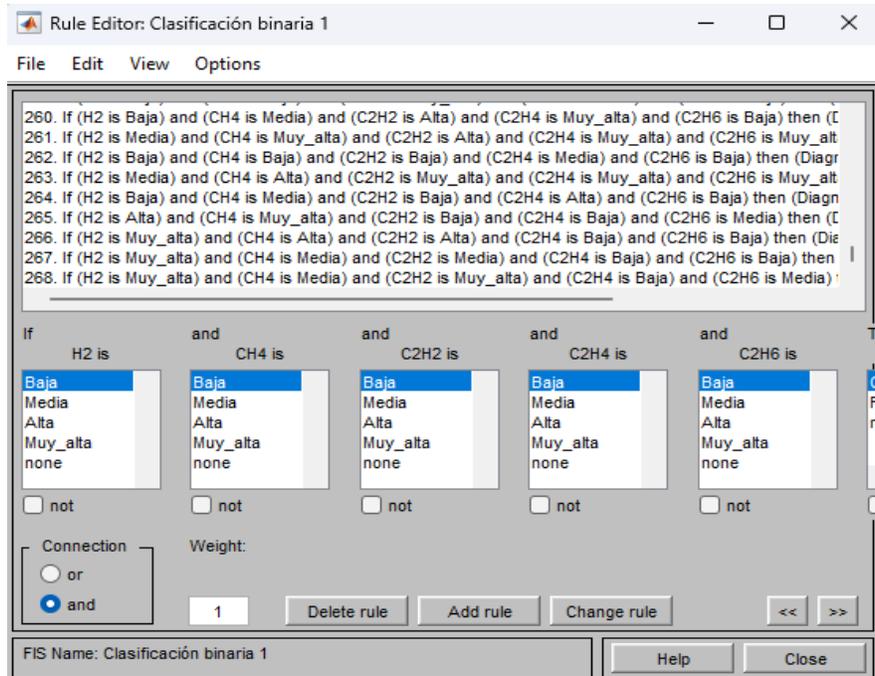


Figura 4.41. Editor de reglas de inferencia en Matlab

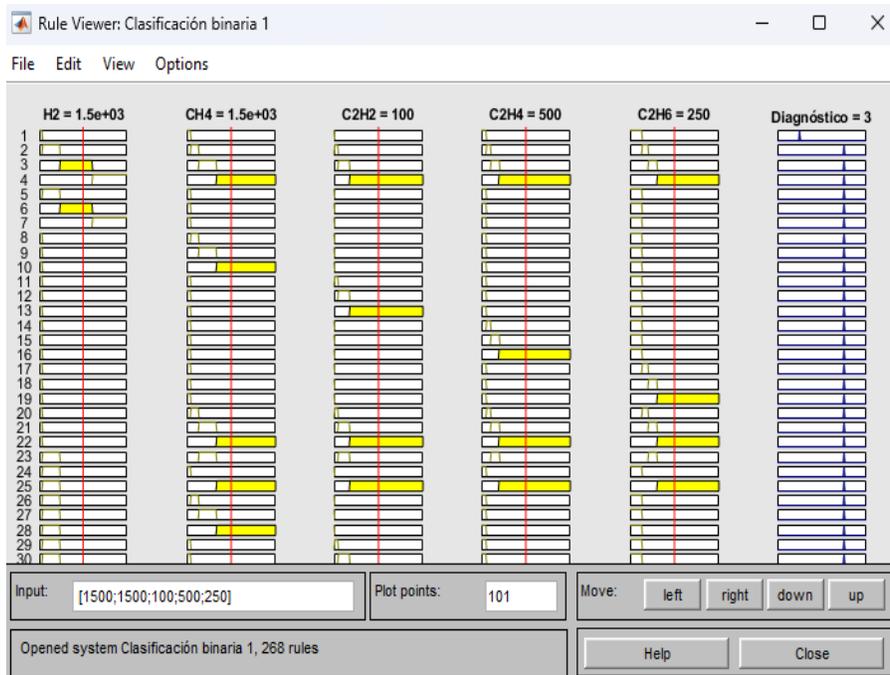


Figura 4.42. Visualizador de reglas de inferencia en Matlab

4.7.2. Modelo de lógica difusa de clasificación multiclase

En el modelo de clasificación binaria tiene como objetivo determinar si los transformadores se encuentran en una condición normal, o si estos presentan algún tipo de falla. Si el modelo determina una condición normal el problema se termina, sin embargo, si el modelo determina lo contrario e indica que se trata de una falla, se procede a probar con el segundo modelo.

El primer modelo toma en cuenta para su modelación los estándares de concentraciones en ppp de gases establecidos por la IEEE. El segundo modelo de clasificación de fallas se basa en el método de Rogers, visto anteriormente en el marco teórico. El método de Rogers consta de tres razones o bien, tres relaciones las cuales toma en cuenta los cinco gases obtenidos en la prueba de gases disueltos en aceite. También, el método de Rogers cuenta con una interpretación de fallas mediante un rango de relaciones, estos rangos son los que se plantean en este modelo.

4.7.2.1 Establecimiento de los parámetros de las variables de entrada

Siguiendo con el siguiente paso para el desarrollo del modelo de LD, de los cinco gases disueltos se obtienen tres relaciones. Esto, aunado a los rangos de relaciones establecidos por Rogers, se establecen cuatro conjuntos difusos de orden 1-4 respectivamente como: baja, media, alta y muy alta.

Este modelo al igual que el primero se dividen en cuatro funciones de membresía de tipo trapezoidal, ya que la MF trapezoidal permite establecer un valor de pertenencia constante igual a 1 en un rango definido y no solo en un valor puntual. De este modo, los parámetros para la modelación del modelo de lógica difusa de clasificación multiclase se presentan en la Tabla 4.16.

Tabla 4.16. Parámetros de modelación de LD multiclase de las variables de entrada.

Variable de entrada	Variables lingüísticas	Función de membresía	Conjunto difuso
CH ₄ /H ₂	Baja	Trapezoidal	[0, 0, 0.09, 0.1]
	Media		[0.09, 0.1, 0.95, 1]
	Alta		[0.95, 1, 2.95, 3]
	Muy alta		[2.95, 3, 10, 10]
C ₂ H ₂ /C ₂ H ₄	Baja	Trapezoidal	[0, 0, 0.09, 0.1]
	Media		[0.09, 0.1, 0.95, 1]
	Alta		[0.95, 1, 2.95, 3]
	Muy alta		[2.95, 3, 10, 10]
C ₂ H ₄ /C ₂ H ₆	Baja	Trapezoidal	[0, 0, 0.09, 0.1]
	Media		[0.09, 0.1, 0.95, 1]
	Alta		[0.95, 1, 2.95, 3]
	Muy alta		[2.95, 3, 10, 10]

4.7.2.2 Establecimiento de los parámetros de la variable de salida

En un principio se contaba con diez clases, las cuales consistía en condición normal y nueve tipos de fallas. El primer modelo descarta por completo la condición normal, por lo que, para este modelo se modelan las nueve fallas restantes.

El establecimiento de la variable de salida también cubre el rango de 0 a 1 y se divide en nueve funciones de membresía de tipo triangular, ya que la MF triangular permite establecer un valor de pertenencia en un valor puntual. De este modo, los parámetros para la modelación de la variable de salida se presentan en la Tabla 4.17.

Tabla 4.17. Parámetros de modelación de LD multiclase de la variable de salida.

Variable de salida	Variables lingüísticas	Función de membresía	Intervalo
Diagnóstico	PD	Triangular	[1, 1, 1]
	D1		[2, 2, 2]
	D2		[3, 3, 3]
	T1 O		[4, 4, 4]
	T1 C		[5, 5, 5]
	T2 C		[6, 6, 6]
	T3 C		[7, 7, 7]
	T3 H		[8, 8, 8]
	S		[9, 9, 9]

4.7.2.3 Reglas de inferencia

Para este primer modelo se obtienen 51 reglas de inferencia, las que posteriormente se alimentan al modelo en Matlab. Algunas de las reglas se presentan en la Tabla 4.18, y el resto se encuentran en la sección de *Anexo 2*.

Tabla 4.18. Reglas de inferencia para el modelo de LD de clasificación multiclase.

CH ₄ /H ₂	C ₂ H ₂ /C ₂ H ₄	C ₂ H ₄ /C ₂ H ₆		Diagnóstico
Baja	Media	Baja	1	PD
Baja	Alta	Media	1	PD
Baja	Media	Media	1	PD
Baja	Alta	Baja	1	PD
Media	Media	Alta	9	S
Media	Media	Muy alta	2	D1
Media	Alta	Alta	2	D1
Media	Alta	Muy alta	3	D2
Media	Muy alta	Alta	2	D1
Media	Muy alta	Muy alta	2	D1
Media	Baja	Alta	9	S
Alta	Baja	Baja	9	S
Muy alta	Baja	Media	4	T1 O
Alta	Baja	Media	4	T1 O
Muy alta	Baja	Baja	4	T1 O
Alta	Baja	Alta	6	T2 C
Muy alta	Baja	Alta	6	T2 C
Alta	Baja	Muy alta	8	T3 H
Muy alta	Baja	Muy alta	8	T3 H
Baja	Media	Alta	1	PD
Baja	Baja	Media	1	PD
Media	Media	Media	9	S
Media	Media	Baja	1	PD
Media	Alta	Baja	1	PD
Baja	Baja	Baja	1	PD

4.7.2.4 Modelación en Matlab

Una vez establecidas las variables lingüísticas y funciones de membresía se modelan en Matlab. Para ello, se introduce la palabra *Fuzzy* en la ventana de comandos, donde posteriormente se abrirá una nueva ventana de diseño con un modelo tipo mamdani de manera predeterminada.

Una vez ya establecidas las variables de entrada y salida con sus respectivas variables lingüísticas y conjuntos difusos se procede a alimentar el modelo. En las Figuras 4.43-4.46 se muestra cómo se establece cada variable de entrada y salida con sus respectivos parámetros de modelación.

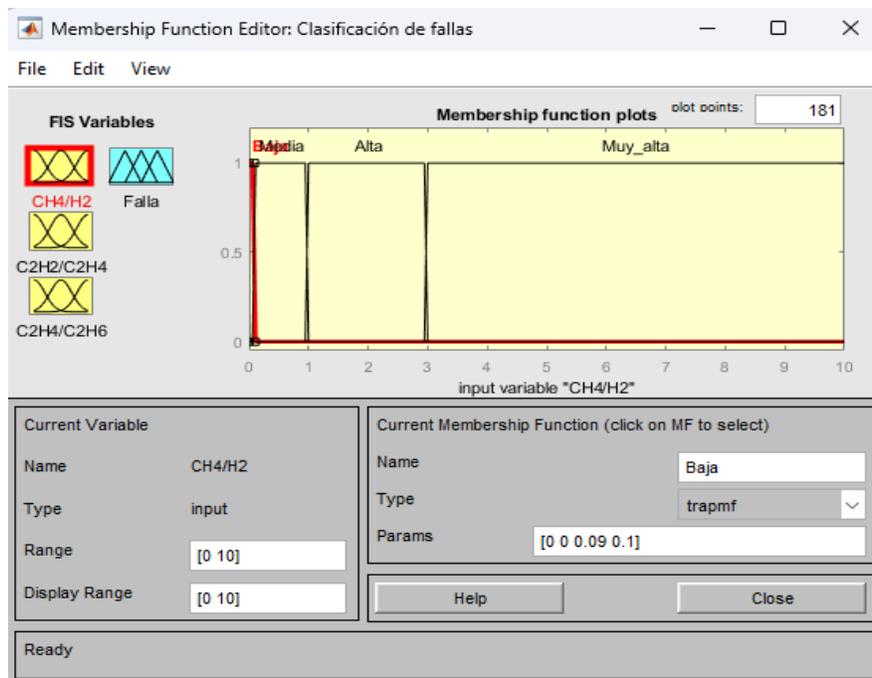


Figura 4.43. Parámetros de la variable de entrada CH_4/H_2

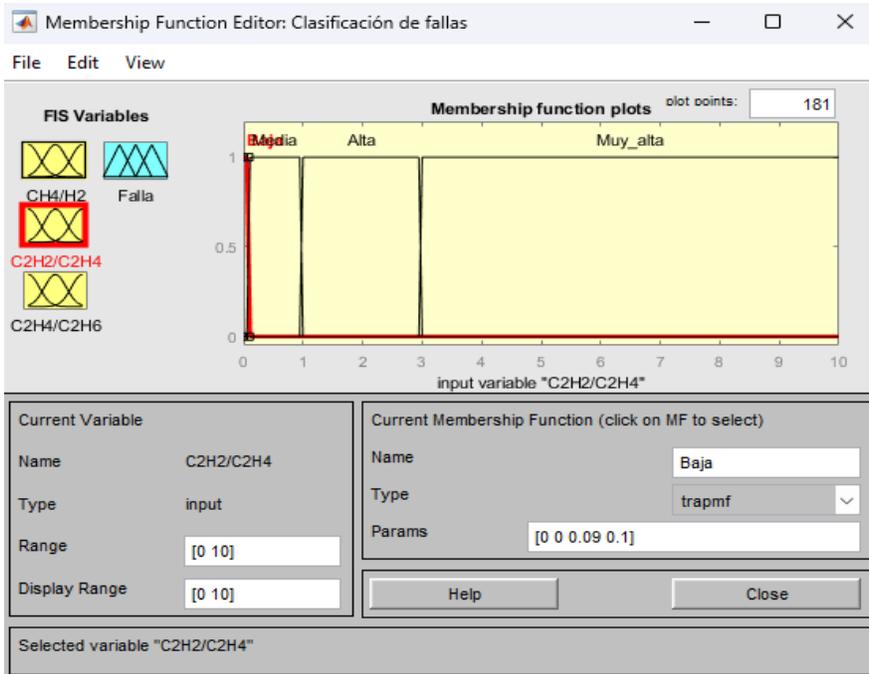


Figura 4.44. Parámetros de la variable de entrada C_2H_2/ C_2H_4

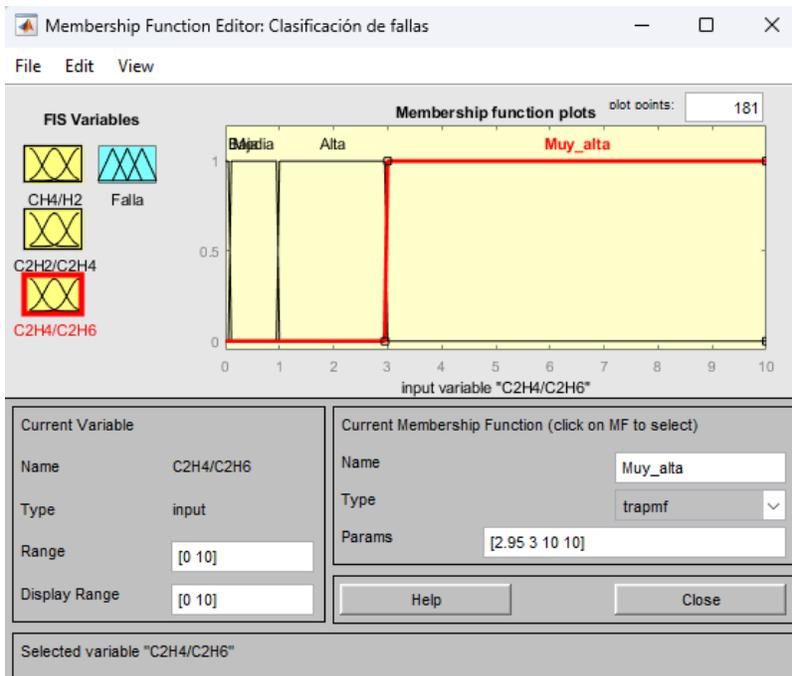


Figura 4.45. Parámetros de la variable de entrada C_2H_4/ C_2H_6

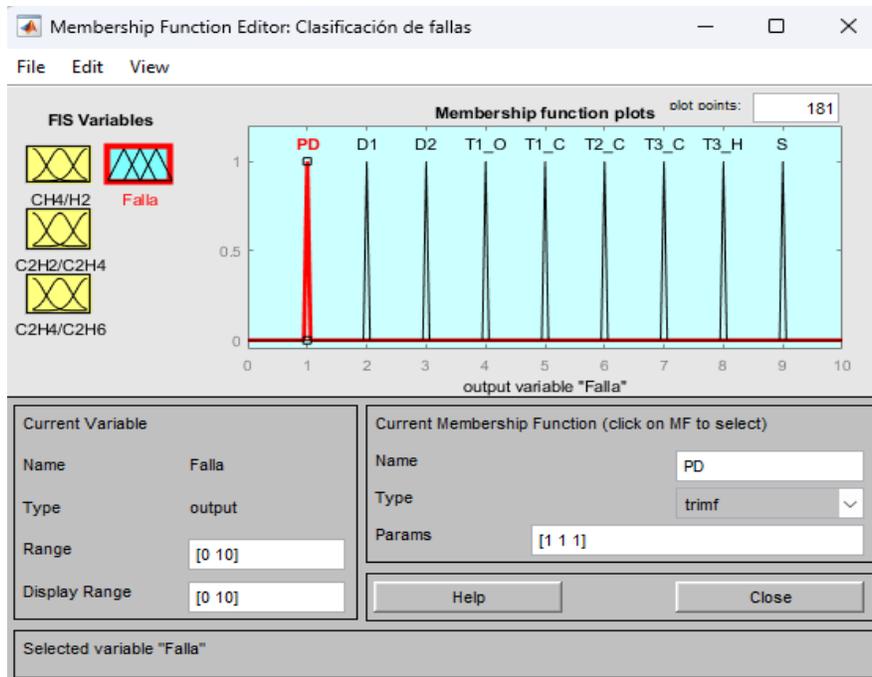


Figura 4.46. Parámetros de la variable de salida Falla

Una vez establecidas las reglas de inferencia se procede a alimentar el modelo en Matlab. La Figura 4.47 muestra el editor de reglas que a su vez permite ver el total y cómo están conformadas. Además, en la Figura 4.48 se muestra cómo se visualizan finalmente las reglas de inferencia en conjunto con las funciones de membresía, donde también se lleva a cabo la predicción de los casos introduciendo las variables de entrada.

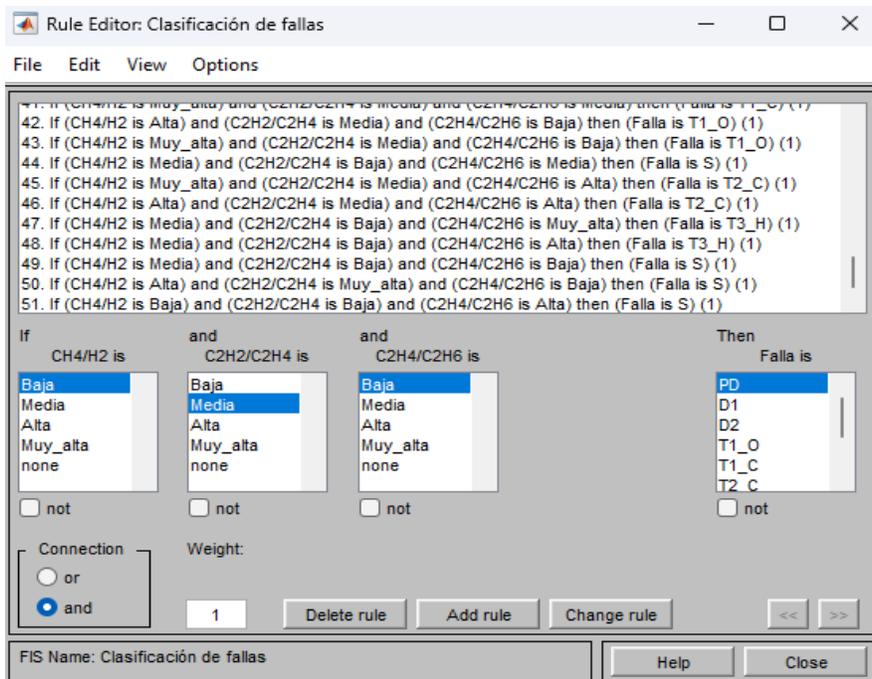


Figura 4.47. Editor de reglas de inferencia en Matlab

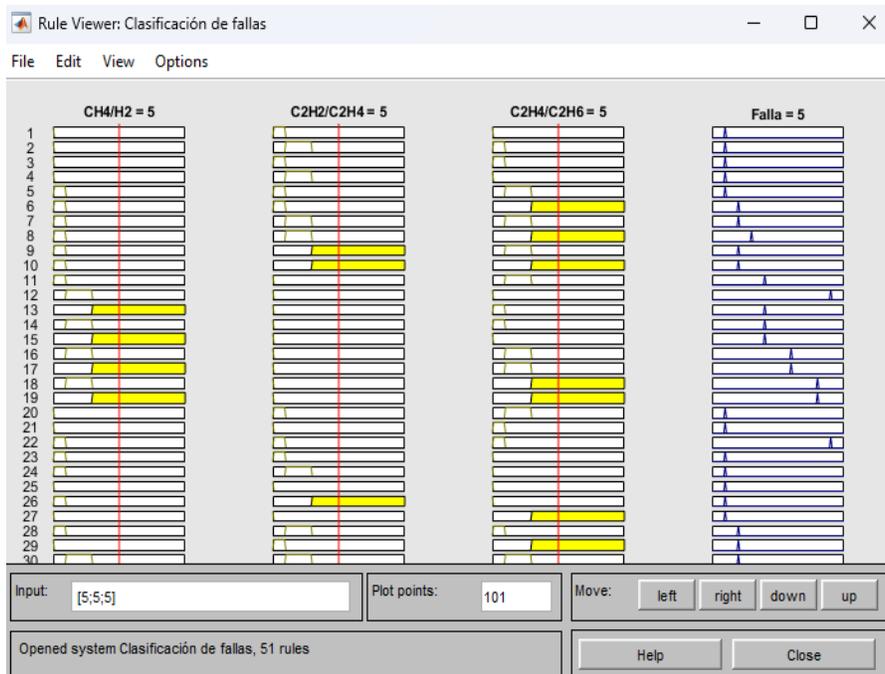


Figura 4.48. Visualizador de reglas de inferencia en Matlab

4.7.3. Modelo de lógica difusa de clasificación multiclase (Versión 2)

El modelo anterior de LD de clasificación multiclase, no obtuvo resultados de predicción tan favorables como se esperaba (*ver capítulo de resultados*), ya que existe mucha confusión entre clases, en específico entre la clase D1 y D2 que se trata de una descarga de baja energía y una descarga de alta energía. Por lo tanto, para reducir los errores del primer modelo de clasificación multiclase, se ha propuesto unir estas clases como una sola y dejándola solo como D de descarga de energía. También, se descarta la falla T1 C, ya que esta solo cuenta con 4 casos para validar el modelo, además de que todos los casos se confunden con las fallas de mayor volumen de datos. Es así, que la variable de salida queda establecida como se muestra en la Tabla 4.19

Tabla 4.19. Clases de la variable de salida del modelo propuesto.

Falla	Nomenclatura	Intervalo
Descargas parciales	PD	[1, 1, 1]
Descargas de energía	D	[2, 2, 2]
Falla térmica de menos de 300°C con sobrecalentamiento	T1 O	[3, 3, 3]
Falla térmica entre 300°C y 700°C con carbonización	T2 C	[4, 4, 4]
Falla térmica de más de 700°C con carbonización	T3 C	[5, 5, 5]
Falla térmica de más de 700°C que solo involucra aceite	T3 H	[6, 6, 6]
Stray gassing	S	[7, 7, 7]

4.7.3.1 Establecimiento de los parámetros de las variables de entrada

Este modelo es muy parecido al anterior, se basa de igual manera en el método de Rogers y al igual que el primero se dividen en cuatro funciones de membresía de tipo trapezoidal, ya que la MF trapezoidal permite establecer un valor de pertenencia constante igual a 1 en un rango definido y no solo en un valor puntual. No obstante, los valores de los conjuntos difusos se ajustan ligeramente para predicciones más precisas. De este modo, los parámetros para la modelación se presentan en la Tabla 4.20.

Tabla 4.20. Parámetros de modelación de LD multiclase de las variables de entrada.

Variable de entrada	Variabes lingüísticas	Función de membresía	Conjunto difuso
CH ₄ /H ₂	Baja	Trapezoidal	[0, 0 ,0.099, 0.1]
	Media		[0.099, 0.1, 0.99, 1]
	Alta		[0.99, 1, 2.99, 3]
	Muy alta		[2.99, 3, 10, 10]
C ₂ H ₂ /C ₂ H ₄	Baja	Trapezoidal	[0, 0 ,0.099, 0.1]
	Media		[0.099, 0.1, 0.99, 1]
	Alta		[0.99, 1, 2.99, 3]
	Muy alta		[2.99, 3, 10, 10]
C ₂ H ₄ /C ₂ H ₆	Baja	Trapezoidal	[0, 0 ,0.099, 0.1]
	Media		[0.099, 0.1, 0.99, 1]
	Alta		[0.99, 1, 2.99, 3]
	Muy alta		[2.99, 3, 10, 10]

4.7.3.2 Establecimiento de los parámetros de la variable de salida

El establecimiento de la variable de salida también cubre el rango de 0 a 1 y se divide en 7 funciones de membresía de tipo triangular, ya que la MF triangular permite establecer un valor de pertenencia en un valor puntual. De este modo, los parámetros para la modelación de la variable de salida se presentan en la Tabla 4.21.

Tabla 4.21. Parámetros de modelación de LD multiclase de la variable de salida.

Variable de salida	Variables lingüísticas	Función de membresía	Intervalo
Diagnóstico	PD	Triangular	[1, 1, 1]
	D		[2, 2, 2]
	T1 O		[3, 3, 3]
	T2 C		[4, 4, 4]
	T3 C		[5, 5, 5]
	T3 H		[6, 6, 6]
	S		[7, 7, 7]

4.7.3.3 Reglas de inferencia

Para este modelo se obtienen 49 reglas de inferencia, las que posteriormente se alimentan al modelo en Matlab. Algunas de las reglas se presentan en la Tabla 4.22, y el resto se encuentran en la sección de *Anexos 3*.

Tabla 4.22. Reglas de inferencia para el modelo de LD de clasificación multiclase.

CH ₄ /H ₂	C ₂ H ₂ /C ₂ H ₄	C ₂ H ₄ /C ₂ H ₆		Diagnóstico
Baja	Baja	Baja	1	PD
Media	Muy alta	Baja	1	PD
Baja	Baja	Muy alta	1	PD
Media	Alta	Media	2	D
Baja	Media	Muy alta	2	D
Alta	Alta	Alta	2	D
Baja	Alta	Alta	2	D
Media	Muy alta	Media	2	D
Baja	Alta	Baja	2	D
Baja	Muy alta	Muy alta	2	D
Alta	Media	Muy alta	2	D
Alta	Alta	Muy alta	2	D
Alta	Media	Media	3	T1 O
Alta	Alta	Media	3	T1 O
Muy alta	Muy alta	Media	3	T1 O
Muy alta	Media	Media	3	T1 O
Alta	Media	Baja	3	T1 O
Muy alta	Media	Baja	3	T1 O
Media	Baja	Media	7	S
Muy alta	Media	Alta	4	T2 C
Alta	Media	Alta	4	T2 C
Media	Baja	Muy alta	6	T3 H
Media	Baja	Alta	6	T3 H
Media	Baja	Baja	7	S
Alta	Muy alta	Baja	7	S
Baja	Baja	Alta	7	S

4.7.3.4 Modelación en Matlab

Una vez establecidas las variables lingüísticas y funciones de membresía se modelan en Matlab. Para ello, se introduce la palabra *Fuzzy* en la ventana de comandos, donde posteriormente se abrirá una nueva ventana de diseño con un modelo tipo mamdani de manera predeterminada.

Una vez ya establecidas las variables de entrada y salida con sus respectivas variables lingüísticas y conjuntos difusos se procede a alimentar el modelo. En las Figuras 4.49-4.52 se muestra cómo se establece cada variable de entrada y salida con sus respectivos parámetros de modelación.

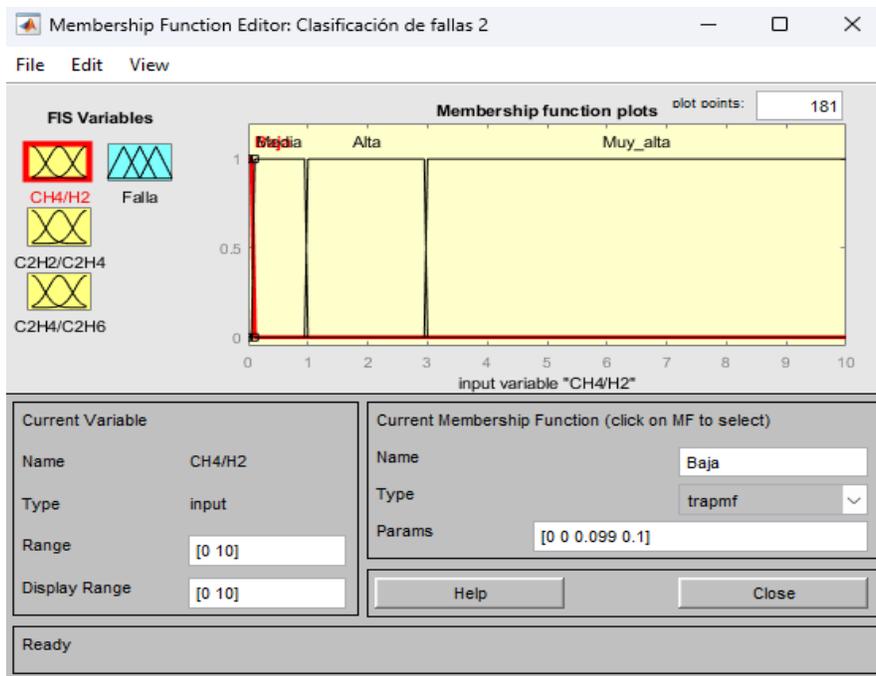


Figura 4.49. Parámetros de la variable de entrada CH_4/H_2

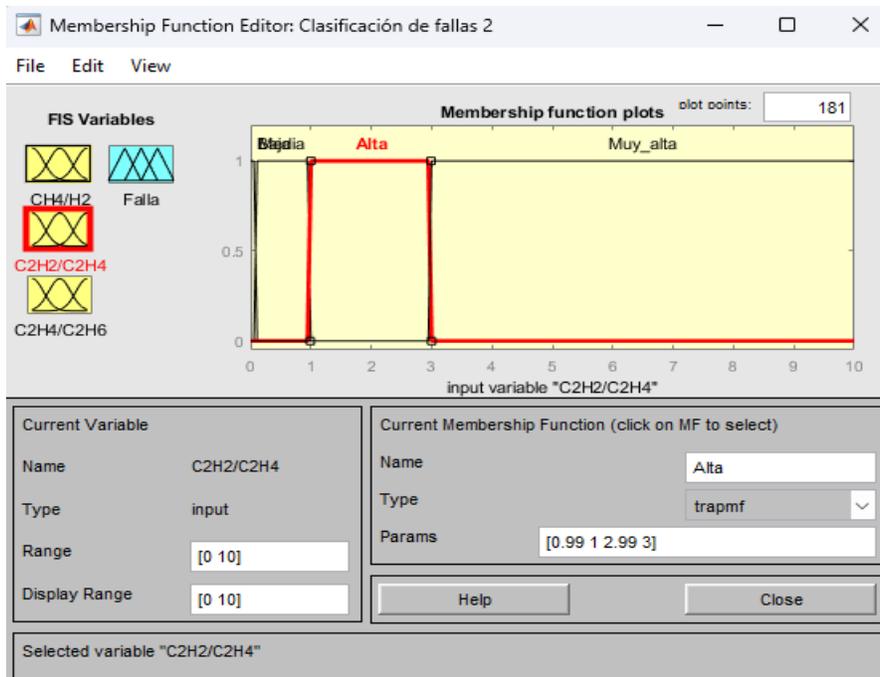


Figura 4.50. Parámetros de la variable de entrada C_2H_2/ C_2H_4

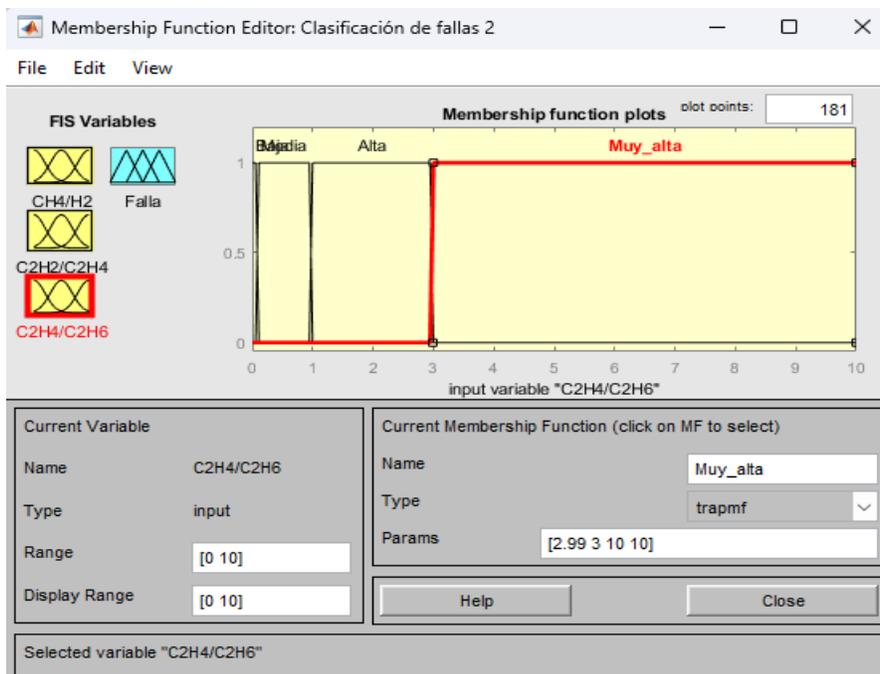


Figura 4.51. Parámetros de la variable de entrada C_2H_4/ C_2H_6

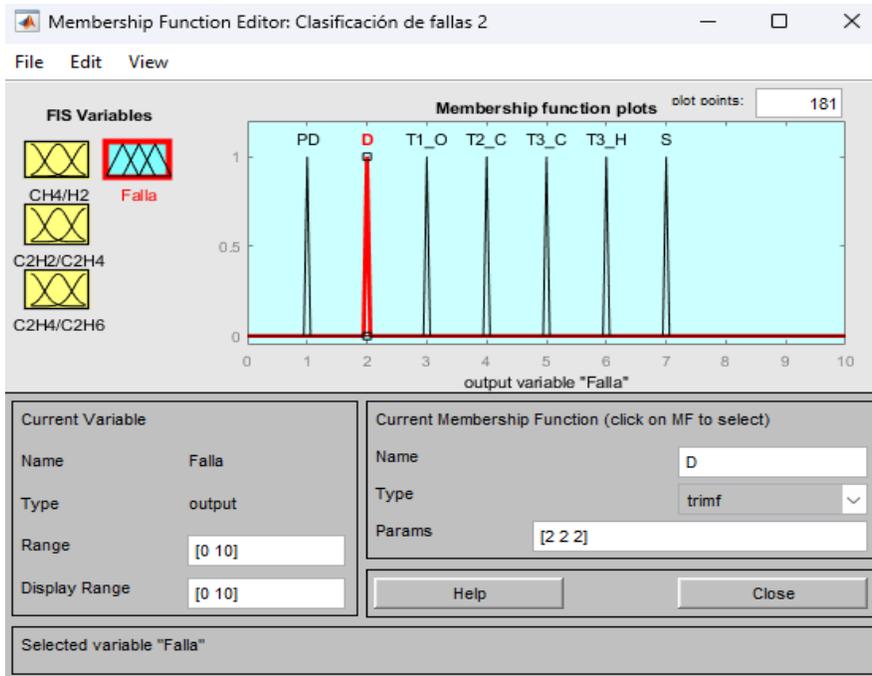


Figura 4.52. Parámetros de la variable de salida Falla

Una vez establecidas las reglas de inferencia se procede a alimentar el modelo en Matlab. La Figura 4.53 muestra el editor de reglas que a su vez permite ver el total y cómo están conformadas. Además, en la Figura 4.54 se muestra cómo se visualizan finalmente las reglas de inferencia en conjunto con las funciones de membresía, donde también se lleva a cabo la predicción de los casos introduciendo las variables de entrada.

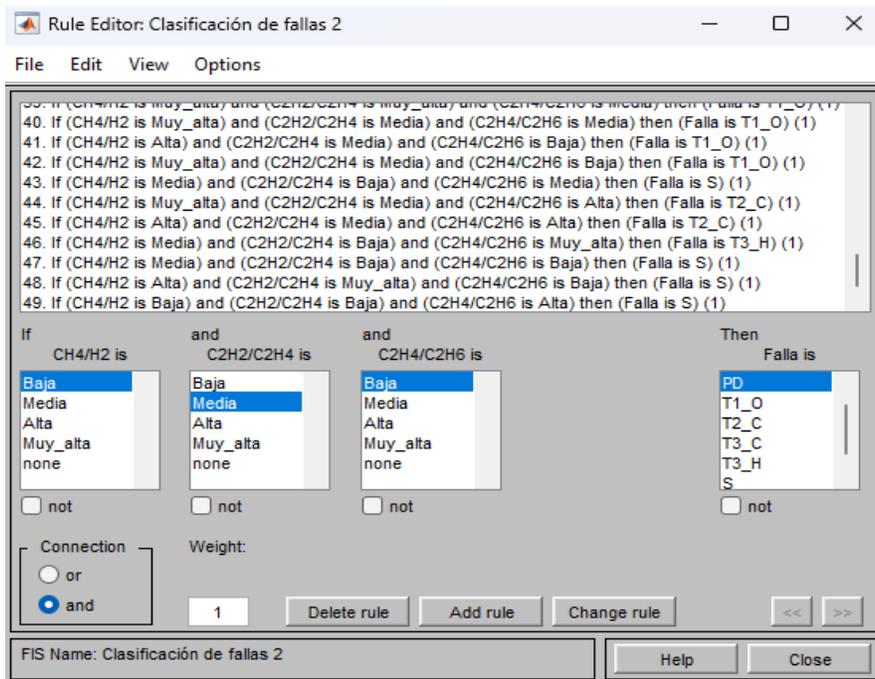


Figura 4.53. Editor de reglas de inferencia en Matlab

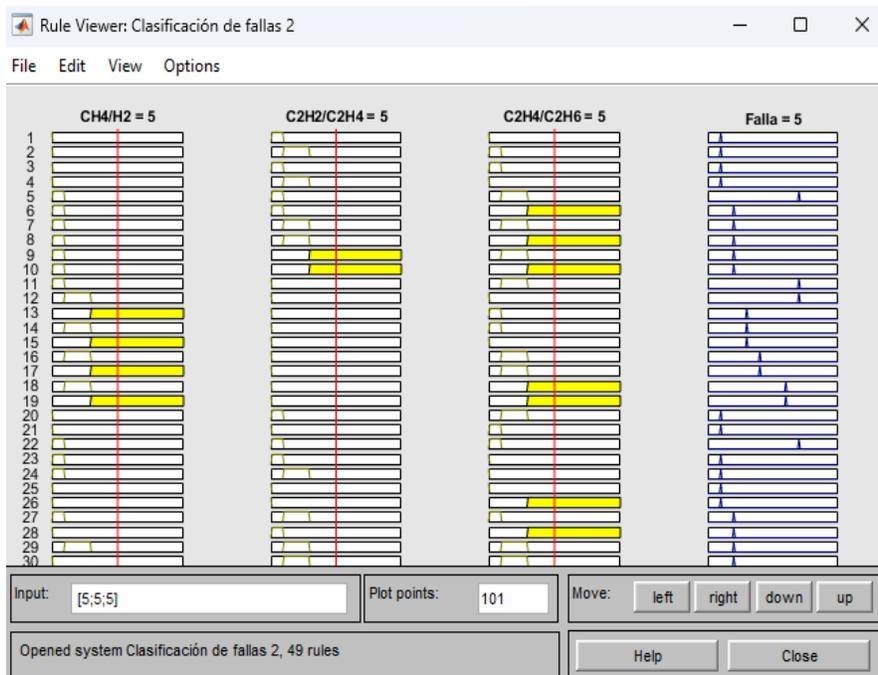


Figura 4.54. Visualizador de reglas de inferencia en Matlab

4.8. Conclusión

El desarrollo de modelos inteligentes para el diagnóstico de fallas en transformadores de potencia, mediante técnicas de inteligencia artificial, se presenta como una vía prometedora para mejorar la eficiencia y precisión en la identificación de problemas en estos equipos críticos. Aunque la investigación en este campo aún se encuentra en etapas incipientes en México, la metodología propuesta aborda de manera estructurada y sistemática la implementación de modelos de machine learning y lógica difusa.

La obtención de datos, el procesamiento y la exploración son pasos esenciales para comprender la naturaleza de la información disponible. La evaluación de diversos algoritmos de machine learning, considerando la desbalanceada naturaleza de la base de datos, resalta la importancia de seleccionar y optimizar modelos para obtener resultados precisos y equitativos.

La propuesta de inclusión de un modelo de lógica difusa, con enfoques binario y multiclase, agrega versatilidad al sistema, permitiendo no solo detectar la presencia de fallas sino también clasificarlas según su tipo. Esta diversificación de enfoques proporciona herramientas más completas para la toma de decisiones, contribuyendo a la eficacia en el mantenimiento y operación de los transformadores.

A pesar de que la implementación de estas tecnologías puede ser un desafío, especialmente en entornos con limitada experiencia en el tema, los beneficios potenciales en términos de precisión, rapidez y capacidad predictiva son sustanciales. En conclusión, la aplicación de inteligencia artificial en el diagnóstico de fallas de transformadores de potencia tiene un gran potencial para transformar la gestión de estos equipos críticos en el contexto mexicano y, posiblemente, en otros lugares del mundo.

CAPÍTULO V. RESULTADOS

Resumen

En este capítulo se tiene como objetivo mostrar los resultados obtenidos de los modelos planteados para el diagnóstico de fallas de transformadores de potencia. También se discuten las áreas de oportunidad en la cuales los modelos pueden ser mejorados. Los resultados de este trabajo de investigación deben convencer y demostrar a los interesados que estos modelos cumplen sus expectativas, y así poder utilizarlos en la práctica habitual como herramientas de mejora en el monitoreo y diagnóstico de fallas y ser de apoyo a los expertos.

5.1. Resultados de los modelos de algoritmos de ML

Se probó un total de diez algoritmos de machine learning, para problemas de clasificación y que, a su vez también son para base de datos desbalanceadas, ya que como se mostró en el desarrollo de la metodología, la base de datos se encontraba muy desequilibrada con respecto a sus clases.

Los algoritmos probados fueron: Support Vector Classification (SVC), Naive Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Decision Tree, Random Forest Classifier, Gradient Boosting Classifier, K-Nearest Neighbors, Neural Network y XGboost.

Por otra parte, se probaron diferentes métricas, que permiten evaluar cuán efectivo es un modelo en este caso de clasificación. Por ejemplo, las métricas como la precisión (accuracy), precisión balanceada (balanced accuracy), el recall, la F1-score y el área bajo la curva ROC y PR se utilizan para evaluar la calidad de las predicciones: la precisión (accuracy) mide la proporción de predicciones correctas sobre el total de predicciones; la precisión balanceada es similar a la anterior, pero se ajusta para conjuntos de datos desbalanceados, considerando la proporción de verdaderos positivos para cada clase; el recall mide la proporción de verdaderos positivos con respecto a todos los casos positivos reales; la precisión mide la proporción de verdaderos positivos con respecto a todas las instancias clasificadas como positivas; f1 score combina precisión y recall y es útil cuando hay desequilibrio entre clases; el área bajo la curva ROC mide la capacidad del modelo para distinguir entre clases. A mayor área, mejor es el rendimiento del modelo; el área bajo la curva PR mide la precisión del modelo en la clasificación de ejemplos positivos, especialmente útil en conjuntos de datos desbalanceados; la matriz de confusión proporciona una visión útil detallada de los resultados de la clasificación, mostrando los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. También se probaron otras métricas para medir y evaluar el rendimiento de los algoritmos tales como el Coeficiente de correlación de Matthews; para evaluar el rendimiento de un modelo de clasificación binaria, especialmente con conjuntos desbalanceados y toma en cuenta los elementos de la matriz de confusión (verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos) para proporcionar una medida equilibrada del rendimiento del modelo sin verse afectado por la distribución de clases, Kappa de Cohen; en problemas de clasificación desbalanceada puede ser preferible sobre la precisión simple, ya que toma en cuenta la posibilidad de que la concordancia ocurra por azar, y por último el

Coeficiente de Jaccard; mide la proporción de la intersección de los conjuntos de elementos predichos y reales con respecto a su unión. Es especialmente útil en elementos predichos y reales con respecto a su unión en problemas de clasificación desbalanceada, un valor más alto de este coeficiente indica una mayor similitud entre las predicciones y las clases reales.

Para el entrenamiento y prueba de los algoritmos se utilizaron 656 casos de la base de datos. Mientras que 85 casos fueron utilizados para validar el mejor modelo obtenido.

En la Tabla 5.1 se muestran los resultados obtenidos de las métricas para cada algoritmo probado. Como se puede observar, el algoritmo K-Nearest Neighbors (K-NN) fue el algoritmo que obtuvo la mayor exactitud en una de las métricas más importantes que es el Balanced Accuracy o exactitud equilibrada ya que esta métrica evalúa los conjuntos de datos desequilibrados y busca proporcionar una medida más equitativa del rendimiento del modelo al considerar el equilibrio entre las clases, en lugar de simplemente maximizar la precisión total del modelo que es la presentada en la columna de Accuracy score, donde el algoritmo XGBoost fue el de mayor rendimiento.

Tabla 5.1. Resultados obtenidos por las diferentes métricas aplicadas a los algoritmos.

Algoritmo	Accuracy score	Balanced Accuracy	C.C. Matthews	Kappa de Cohen	C. de Jaccard
SVC	0.71	0.51	0.65	0.65	0.56
Naive Bayes	0.58	0.56	0.52	0.51	0.41
Linear Discriminant Analysis	0.61	0.55	0.57	0.54	0.44
Quadratic Discriminant Analysis	0.71	0.65	0.67	0.66	0.56
Decision Tree	0.66	0.48	0.61	0.60	0.50
Random Forest Classifier	0.78	0.53	0.73	0.72	0.64
Gradient Boosting Classifier	0.82	0.64	0.79	0.78	0.70
K-Nearest Neighbors	0.75	0.67	0.70	0.70	0.61
Neural Network	0.68	0.43	0.61	0.59	0.52
XGBoost	0.84	0.66	0.81	0.81	0.74

En la Figura 5.1 se presenta el gráfico de las curvas ROC de cada uno de los algoritmos probados, donde se obtuvo el macro promedio para cada algoritmo, esta métrica indica que tan bueno es el modelo en todas las clases por igual, sin importar el número de instancias, es decir, sugiere un

buen rendimiento en todas las clases, independientemente de su tamaño. Se puede observar que los algoritmos XGBoost y KNN son los que obtuvieron el macro promedio más alto siendo este de 0.82 para ambos. Por otra parte, en la Figura 5.2 se muestra el grafico PR para micro promedios, el cual indica el rendimiento general del modelo dándole más pesos a las clases más grandes (en términos de numero de instancias), es decir, un alto micro promedio indica un buen equilibrio general en la capacidad del modelo para clasificar correctamente todas las instancias, independientemente de la clase. Se puede observar que los algoritmos XGBoost, Gradient Boosting y KNN obtuvieron los micro promedios más altos.

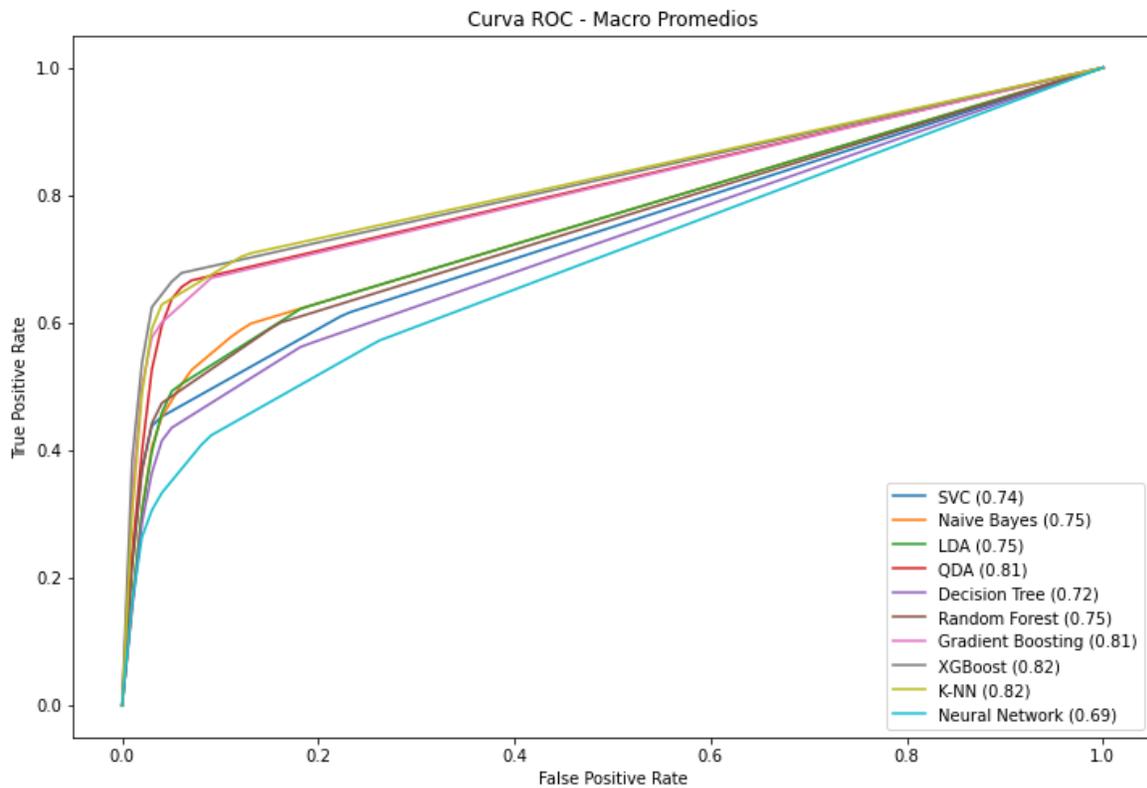


Figura 5.1. Curva ROC- Macro promedios de cada algoritmo probado.

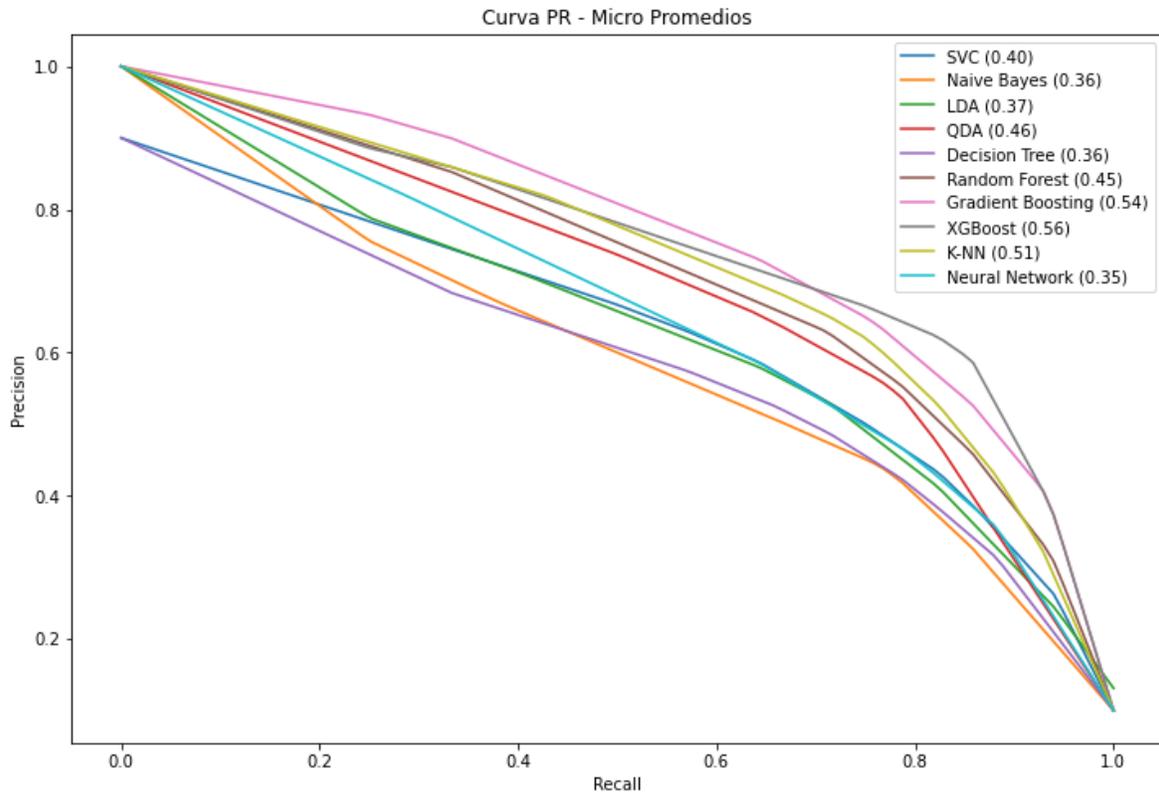


Figura 5.2. Curva PR- Micro promedios de cada algoritmo probado.

En comparación, si ambos valores (micro y macro promedio) son altos, eso indica que el modelo está funcionando bien en general y también tiene un buen rendimiento en clases más pequeñas, sin embargo, existe una diferencia significativa entre ambos valores, lo cual indica que el modelo se desempeña significativamente mejor en clases más grandes. De este modo, los modelos enfrentan desafíos al no poder clasificar de mejor manera clases con menos instancias, lo cual afecta considerablemente la precisión, ya que al tratarse de una BD desbalanceada, por ende, habrá clases con menos instancias y que sin embargo siguen siendo clases únicas e importantes de clasificación.

Una vez obtenidas las métricas de todos los algoritmos, se tomaron a los dos mejores y se llevó a cabo la optimización de los dos mejores algoritmos (Tabla 5.2), siendo estos K-Nearest Neighbors y XGboost. Donde una vez más el algoritmo de K-NN fue el que obtuvo el mejor rendimiento siendo este del 0.7, a pesar de que fue el algoritmo con el mejor rendimiento, este valor sigue siendo no tan adecuado considerando el tamaño de la base de datos y el número de variables involucradas.

Tabla 5.2. Resultados obtenidos por las diferentes métricas aplicadas a los algoritmos optimizados.

Algoritmo	Accuracy score	Balanced Accuracy	C.C. Matthews	Kappa de Cohen	C. de Jaccard
K-Nearest Neighbors	0.77	0.70	0.72	0.71	0.62
XGBoost	0.84	0.66	0.81	0.81	0.73

En la Figura 5.3 se presenta el gráfico de las curvas ROC de los algoritmos optimizados (KNN y XGBoost), donde se obtuvo el macro promedio para cada uno. Se puede observar que el algoritmo KNN obtuvo el macro promedio más alto siendo este de 0.84 contra 0.82 obtenido por XGBoost. Por otra parte, en la Figura 5.4 se muestra el gráfico PR para micro promedios, donde se observa que de igual forma, el algoritmo KNN obtuvo un mejor valor sobre el algoritmo XGBoost.

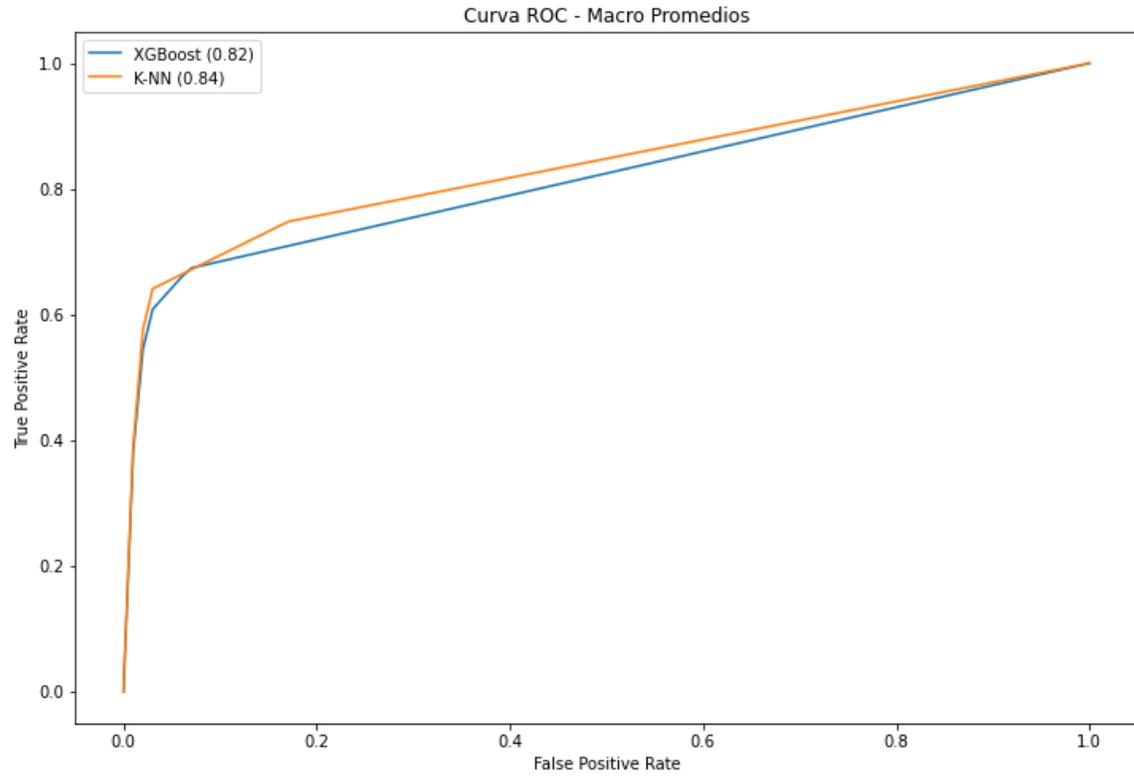


Figura 5.3. Curva PR- Macro promedios de KNN y XGboost optimizados.

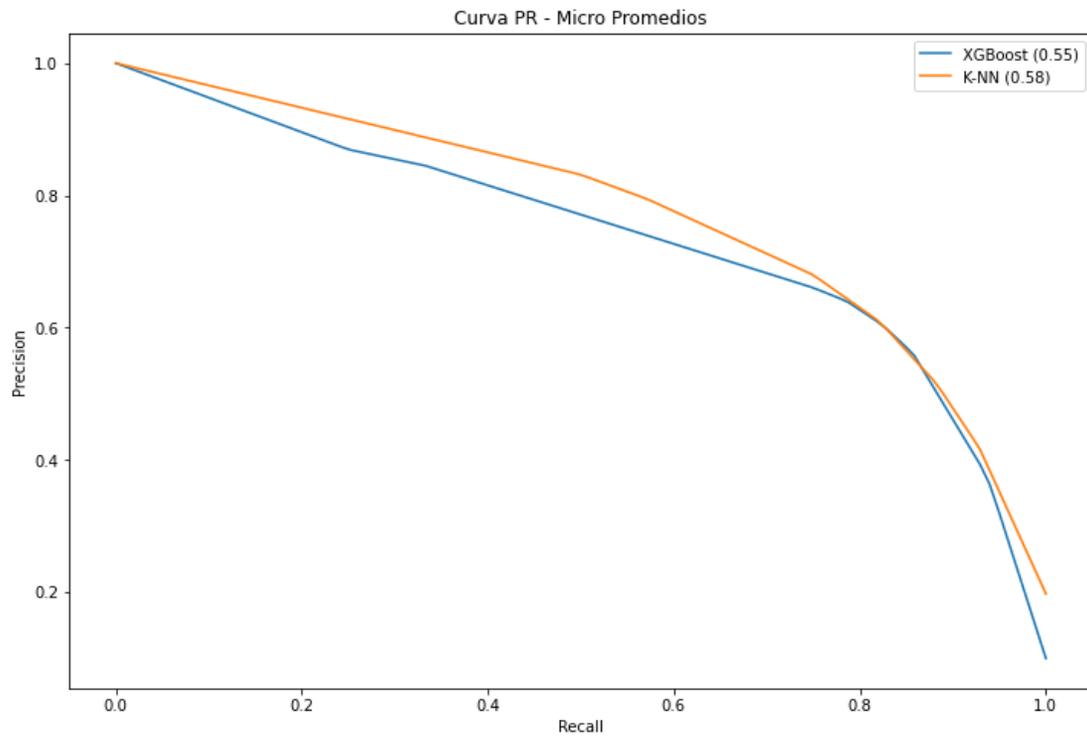


Figura 5.4. Curva PR- Micro promedios de KNN y XGboost optimizados.

Por otra parte, en la Figura 5.5 se presenta un gráfico donde se visualiza la importancia de las características de la base de datos por parte del algoritmo de XGBoost. Este algoritmo proporciona una indicación de cuánto cada característica contribuye a las decisiones tomadas por el modelo durante el proceso de entrenamiento. El gráfico muestra en su eje “y” las características y sus importancias en este caso relativa en el eje “x”. La importancia se da en términos relativos y se calcula en relación con las demás características del conjunto de datos. Un valor de importancia más alto tiene un impacto más significativo en las decisiones del modelo, que para este caso es el acetileno (C_2H_2), seguida del etileno (C_2H_4), etano (C_2H_6), hidrógeno (H_2) y con el de menor importancia se tiene al metano (CH_4). Sin embargo, la importancia de las características más bajas no significa necesariamente que esas características no son importantes ya que podrían desempeñar un papel crucial en combinación con otras características o en contextos específicos para la clasificación.

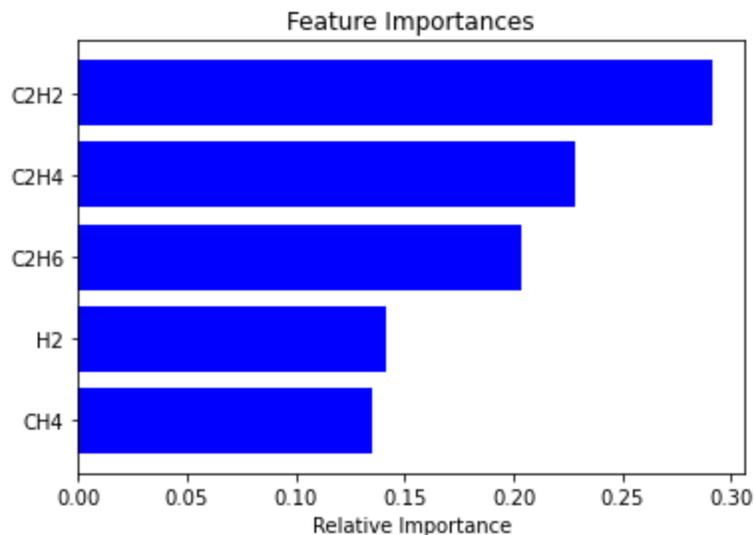


Figura 5.5. Gráfico de importancia de características de XGBoost

Otro paso crítico en la interpretación de un modelo como el de XGBoost es la visualización de árboles individuales para comprender cómo el modelo toma decisiones en cada uno de ellos.

El árbol de decisión divide las muestras en función de condiciones, asignando valores de destino. Este proceso se repite a medida que desciende por el árbol hasta llegar a las hojas.

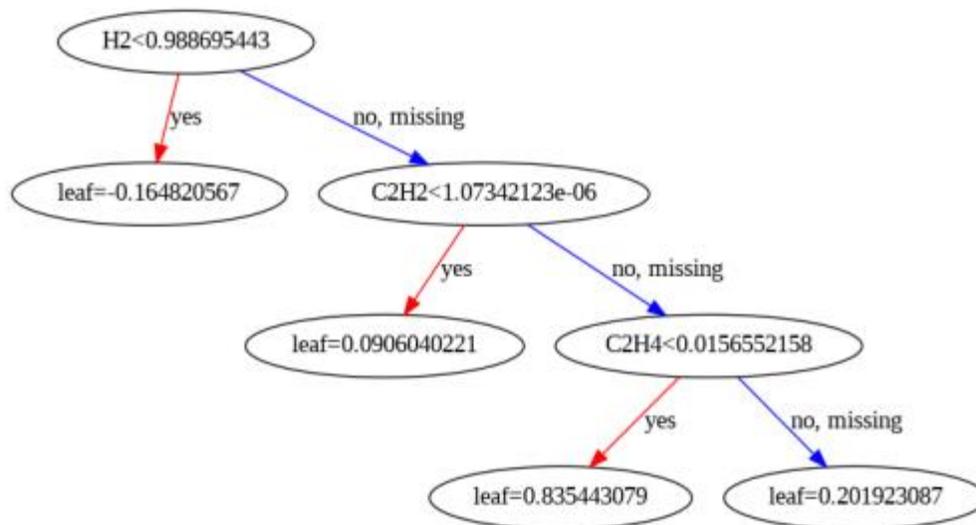


Figura 5.6. Primer árbol del modelo optimizado de XGBoost

En la Figura 5.6 se presenta el primer árbol de decisión del modelo obtenido por el algoritmo XGBoost optimizado. En donde, en el nodo raíz, todas las muestras del conjunto de entrenamiento se dividen inicialmente en función de la condición impuesta por $H_2 < 0.9886$ (Hidrógeno). Donde cierto porcentaje de las muestras que cumplieron con dicha condición del nodo raíz se dirigen hacia la hoja izquierda que representa una salida o etiqueta. En cambio, el porcentaje de muestras que no cumplieron con la condición del nodo raíz se dirigen hacia el nodo derecho que representa la condición impuesta por $C_2H_2 < 1.0734 \times 10^{-06}$ (acetileno). Las muestras que cumplen con esa condición se dirigen hacia la hoja izquierda que representa otra etiqueta o salida, mientras que las muestras que no cumplen se dirigen a un tercer nodo con la condición impuesta $C_2H_4 < 0.0156$ (etileno). Este último nodo dirige a las muestras que cumplan o no a diferentes hojas que establecen diferentes condiciones y que son el comienzo de un nuevo árbol.

Es importante mencionar que un árbol individual de XGBoost no proporciona directamente la visualización de las etiquetas de clase para un problema de clasificación, dado que los árboles están diseñados para trabajar en conjunto y mejorar la predicción combinada del conjunto. Es decir, las predicciones de clases son resultado de las decisiones de múltiples árboles.

La optimización de los dos mejores algoritmos mostró resultados muy similares, sin embargo, el algoritmo KNN mostró ser mejor en cuanto a la clasificación en términos de equidad de clases, lo cual es un punto importante en esta clasificación, ya que todas las clases por muy desbalanceadas que se encuentren son importantes debido a que cada una representa una falla en particular de los transformadores de potencia. Es así, que se toma al algoritmo KNN para su validación y evaluación de métricas con los casos restantes que no fueron incluidos en el entrenamiento de los algoritmos.

5.1.1. Evaluación del modelo del algoritmo optimizado de K-Nearest Neighbor

Después de haber elegido al mejor modelo y haberlo optimizado, se requiere probar con datos reales. El modelo se probó con los 85 casos que no fueron incluidos en el entrenamiento y prueba de los algoritmos.

Se llevó a cabo la predicción de los ochenta y cinco casos y se evalúalo su rendimiento contra las etiquetas reales, donde se obtuvo una exactitud resultante de 0.70 que representa la proporción de predicciones correctas frente al total. Mientras, que la métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.62. Las métricas como la precisión, recall y f1 score para evaluar el rendimiento del modelo se muestran en la Tabla 5.3

Tabla 5.3. Reporte de clasificación de las métricas precisión, recall y f1 score del modelo KNN.

Reporte de clasificación										
Clase	0	1	2	3	4	5	6	7	8	9
Precision	0.47	0.83	0.67	0.6	0.91	0	1	0.6	0.92	0.6
Recall	0.73	0.83	0.67	0.75	0.83	0	0.6	0.5	0.8	0.55
F1 score	0.57	0.83	0.67	0.67	0.87	0	0.75	0.55	0.86	0.57

En la Tabla 5.3 se presenta el rendimiento que obtuvo el modelo para cada una de las clases/etiquetas. Donde se puede observar que la clase 0 correspondiente a una condición normal es la que tiene el peor rendimiento de predicción. La clase 1,4 y 8 correspondientes a descargas parciales (PD), falla térmica menor a 300°C con sobrecalentamiento y falla térmica mayor a 700°C que solo involucra aceite, obtuvieron más del 80% de precisión en su clasificación. Sin embargo,

el modelo no fue capaz de predecir correctamente la clase 5, correspondiente a una falla térmica menor a 300°C con carbonización.

5.1.2. Matriz de confusión

En la matriz de confusión (Figura 5.7) se puede observar que el modelo predijo correctamente 59 casos de 85: donde la clase 5, el único caso lo clasificó como clase 1. En la matriz de confusión normalizada (5.8) se observa lo mismo que en la Tabla 5.3, donde las clases 1, 4 y 8 obtuvieron una precisión mayor al 80%, mientras que el 100% de la clase 5 fue clasificada como clase 1.

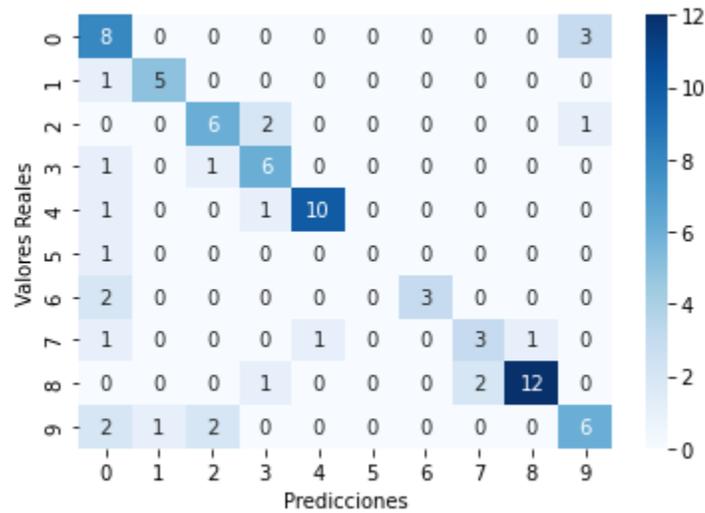


Figura 5.7. Matriz de confusión obtenida de los valores de predicción versus las etiquetas reales.

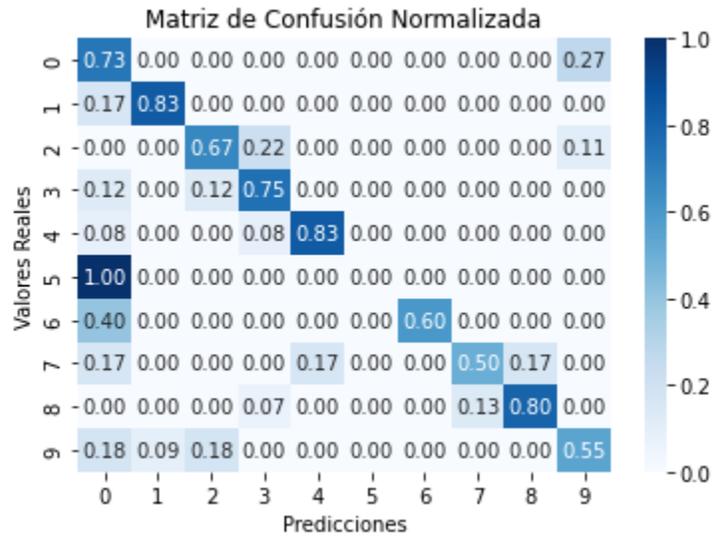


Figura 5.8. Matriz de confusión normalizada de los valores de predicción versus las etiquetas reales.

5.1.3. Curva ROC

En la Figura 5.9 se presenta el gráfico de la Curva ROC de la evaluación del modelo de las predicciones versus las etiquetas reales para cada clase. Donde, el eje X de la curva ROC representa la tasa de los falsos positivos y el eje Y representa la tasa de verdaderos positivos.

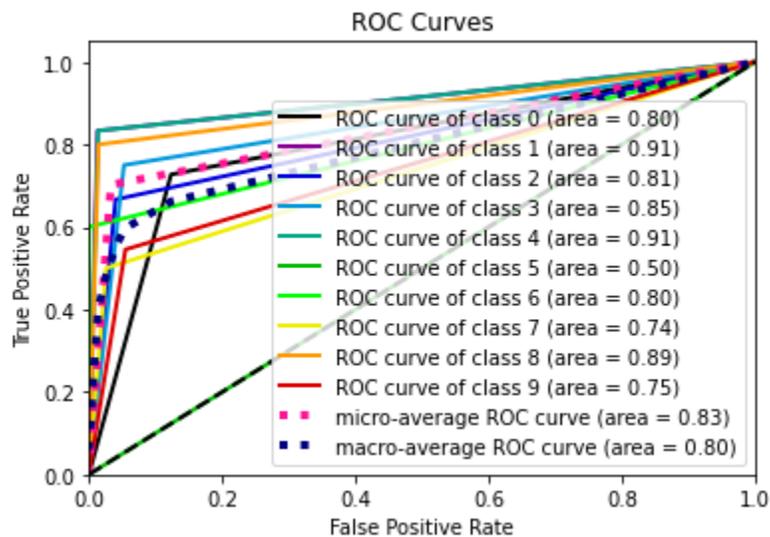


Figura 5.9. Curva ROC de los valores de predicción versus las etiquetas reales.

De igual manera, se observa que las clases con la mayor área bajo la curva son la 1, 4 y 8 mientras que la clase 5, se encuentra justo en la diagonal principal la cual indica que la clasificación para esta clase se comporta como un modelo aleatorio. Por otra parte, el micro average trata a todas las instancias por igual, es decir como si todas las instancias fueran parte de una única clase, obteniendo un rendimiento del 83%. Mientras que el macro average calcula un promedio no ponderado de las métricas obtenidas por cada clase por separado, es decir, calcular la métrica para cada clase y toma el promedio.

5.1.4. Curva PR

En la Figura 5.10 se presenta el gráfico de la Curva PR de la evaluación del modelo de las predicciones versus las etiquetas reales. Donde, el eje X representa la tasa de verdaderos positivos, que es equivalente al recall, y el eje Y que representa la precisión.

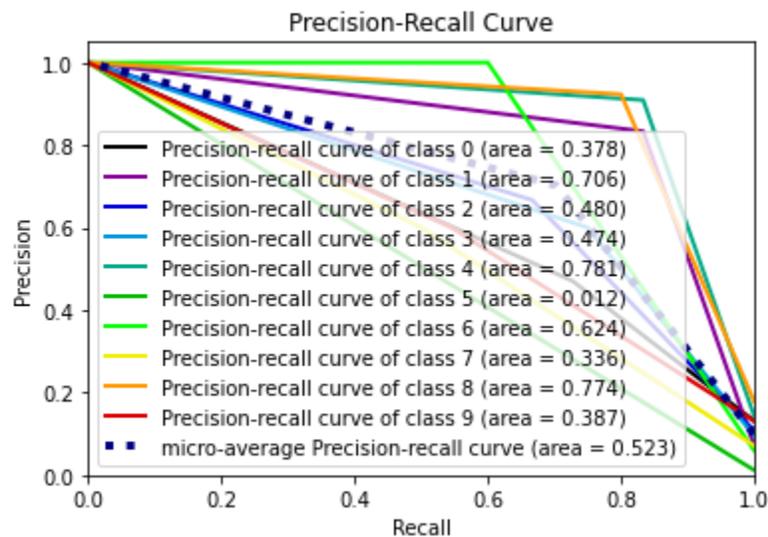


Figura 5.10. Curva PR de los valores de predicción versus las etiquetas reales.

En la curva PR se observa que las clases con la mayor área bajo la curva son la 1, 4 y 8 mientras que la clase 5, se encuentra justo en la diagonal principal la cual indica que la clasificación para esta clase se comporta como un modelo aleatorio. Por otra parte, el micro average trata a todas las instancias por igual, es decir como si todas las instancias fueran parte de una única clase, obteniendo un rendimiento del 83%.

5.1.5. Ganancia acumulativa

La ganancia acumulativa es una métrica que se utiliza para medir la efectividad del modelo en la identificación de instancias positivas. Este gráfico se crea calculando la proporción acumulativa de verdaderos positivos (instancias positivas correctamente clasificadas) en relación con el total de instancias positivas, a medida que se varía el umbral de clasificación. Asimismo, en la Figura 5.10 se puede interpretar cómo la proporción acumulativa de verdaderos positivos cambia a medida que se consideran más instancias, por ejemplo, del 60% de instancias totales, alrededor del 20% de estas han sido clasificadas correctamente. Por lo tanto, una curva que sube rápidamente indica un modelo eficaz en la identificación de instancias positivas, de lo contrario, el modelo tiene problemas para clasificar correctamente dichas instancias y que por consecuencia afectan a su rendimiento de clasificación.

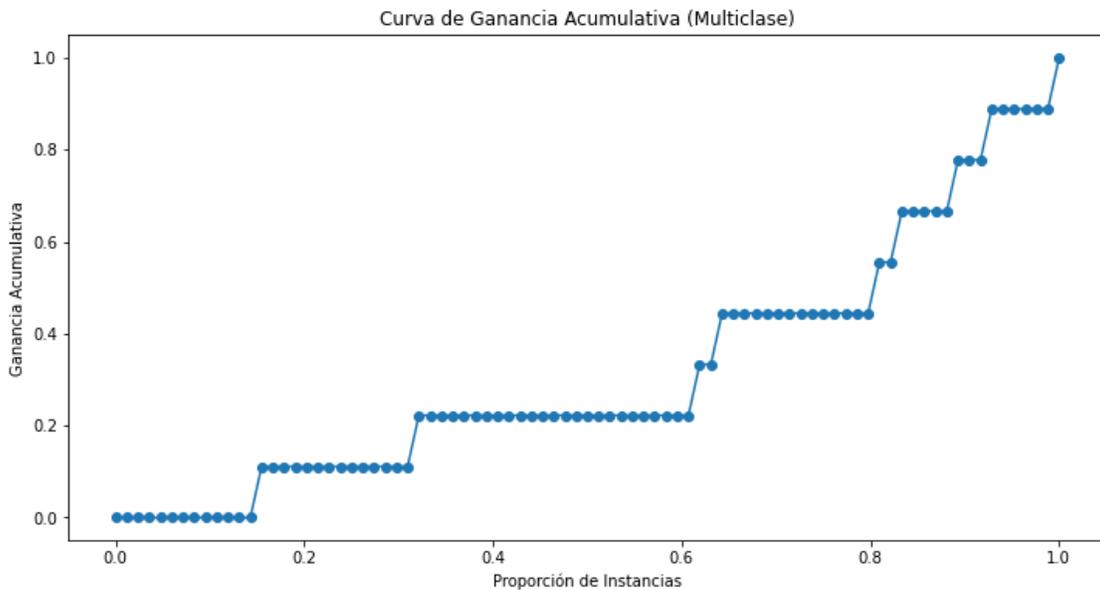


Figura 5.10. Curva de ganancia acumulativa de la validación del algoritmo KNN

5.2. Resultados de los modelos de lógica difusa

Es importante mencionar que la diferencia en el comportamiento del modelo de lógica difusa en comparación con los modelos de machine learning (ML) tradicionales puede atribuirse a las diferencias fundamentales en la forma en que estos enfoques abordan y modelan la información. En el caso de los modelos de LD: suelen ser altamente interpretables, es decir, que las reglas y la lógica subyacente utilizadas para tomar decisiones son comprensibles y explicables para los humanos. Cada regla en el sistema tiene un significado intuitivo y puede describirse fácilmente. Asimismo, la LD es efectiva para modelar situaciones en las que la información disponible es vaga o incierta (manejo de incertidumbre), introduciendo grados de pertenencia y permitiendo representar la incertidumbre de manera más realista que los modelos de ML tradicionales. También, permite una explicación clara de cómo se llega a una conclusión, ya que cada regla y su contribución a la salida final son transparentes, proporcionando una comprensión detallada de cómo se realiza la inferencia. Por otra parte, muchos modelos de ML, especialmente los basados en redes neuronales profundas, tienden a ser complejos y actúan como cajas negras. Esto significa que la relación entre las entradas y las salidas puede ser difícil de entender y explicar. Además de que los modelos de ML a menudo requieren la optimización de una gran cantidad de parámetros lo que al final dificulta la interpretación. Esto conlleva a que los modelos complejos de ML puedan tener un rendimiento impresionante, pero a menudo carecen de explicabilidad.

Es así como, en el contexto actual de la inteligencia artificial, la explicabilidad se ha vuelto crucial. Los modelos que pueden explicar sus decisiones son preferibles en aplicaciones críticas donde se requiere confianza y comprensión del proceso de toma de decisiones. Explainable AI busca un equilibrio entre el rendimiento y explicabilidad, es decir, sus enfoques buscan encontrar un equilibrio entre el rendimiento del modelo y la capacidad de explicar sus decisiones. Métodos como la lógica difusa, árboles de decisión y modelos lineales pueden ofrecer este equilibrio. Esto es importante para fortalecer la confianza del usuario, más aún en entornos donde esta confianza del usuario es esencial (como la atención médica, finanzas o sistemas de control), la capacidad de entender y justificar las decisiones del modelo es crucial.

En pocas palabras, mientras los modelos de lógica difusa se destacan por su interpretabilidad y manejo de la incertidumbre, los modelos de ML a menudo ofrecen un rendimiento impresionante, pero a expensas de la explicabilidad.

Como se mostró en el desarrollo de la metodología, se desarrollaron dos modelos de LD, el primero con una salida binaria, el cual clasifica en condición normal y falla. Mientras, que el segundo modelo va de la mano del primero, es decir, si el primer modelo indica que se trata de una falla, el segundo modelo te predice de que falla se trata. Sin embargo, para predecir en el primer modelo, se utilizan las concentraciones en ppm de los gases obtenidos directamente de las pruebas DGA (gases disueltos en aceite), mientras que para la clasificación de fallas en el segundo modelo se requiere de 3 relaciones, las cuales se obtienen con la división de la ppm de los gases, tal como se mostró en la metodología.

No obstante, también se probaron métricas para medir y evaluar el rendimiento de los modelos tales como el Coeficiente de correlación de Matthews, Kappa de Cohen y Coeficiente de Jaccard.

El primer modelo de LD se probó con la base de datos completa, es decir, se probaron 741 casos. Para el segundo modelo, se descartan los casos con salida igual a condición normal, así como también los casos donde no se obtienen valores en las relaciones, quedando un total de 467 casos para probar el modelo.

5.2.1. Evaluación del modelo de lógica difusa con salida binaria

Una vez probado el modelo, se cuenta con los datos de las etiquetas reales, así como también con datos de predicción. Estos datos se evalúan con las métricas más comunes utilizadas en los algoritmos, como lo son el accuracy score, balanced accuracy y la matriz de confusión.

De este modo, se llevó a cabo la predicción de los 741 casos y se evaluó su rendimiento contra las etiquetas reales, donde se obtuvo una exactitud resultante de 0.96 que representa la proporción de predicciones correctas frente al total. Mientras, que la métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.97.

5.2.1.1 Matriz de confusión

En la matriz de confusión (Figura 5.11) se puede observar que el modelo predijo correctamente 714 casos de 741: donde la clase 1 representa la condición normal y clasificó todos los casos de manera correcta, mientras que la clase 3 representa la falla, y se puede observar que 3 casos los clasificó como clase 1 y 24 no pudo clasificarlos correctamente. Esto último quiere decir, que el modelo no encontró alguna regla de inferencia por la cual pudiera clasificar esa instancia y por consiguiente lo clasifica como el valor medio del rango establecido, en este caso siendo 2 el valor medio del rango 0-4.

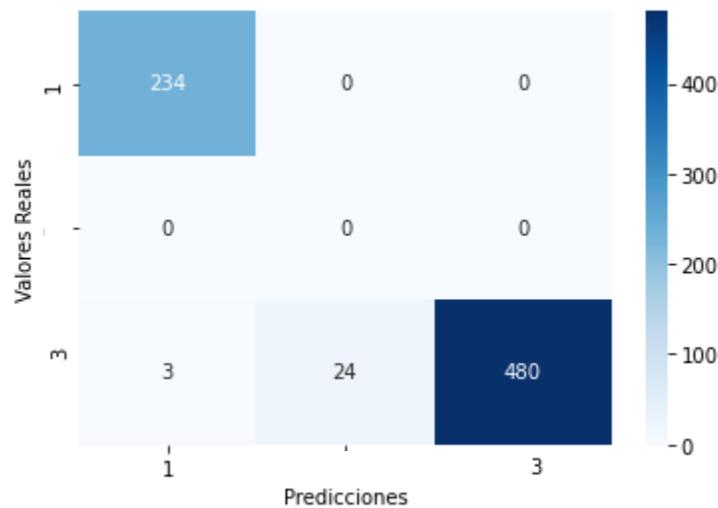


Figura 5.11. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD binario.

5.2.2. Evaluación del modelo de lógica difusa con salida multiclase

Siguiendo la misma lógica, se llevó a cabo la predicción de los 467 casos y se evaluó su rendimiento contra las etiquetas reales, donde se obtuvo una exactitud resultante de 0.65 que representa la proporción de predicciones correctas frente al total. Mientras, que la métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.63.

5.2.2.1 Matriz de confusión

En la matriz de confusión (Figura 5.12) se puede observar que el modelo predijo correctamente 308 casos de 467: donde la clase 4 y 8 son las de mejor rendimiento. La clase 2 es clasificada 15 veces como clase 3, mientras la clase 3 es clasificada 25 veces como clase 2. Por otra parte, la clase 7 es clasificada erróneamente en todas las instancias, siendo clasificadas en su mayoría como clase 8. Estos errores se deben a que existen reglas de inferencia que coinciden con más de una etiqueta.

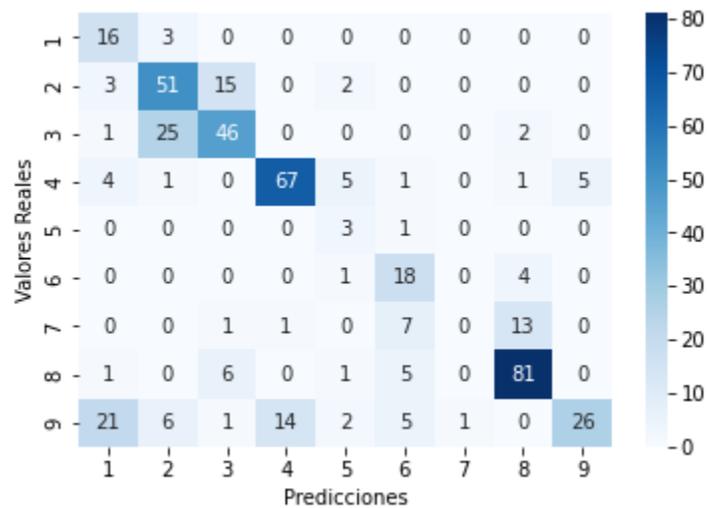


Figura 5.12. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD multifalla.

5.2.3. Evaluación del modelo de lógica difusa con salida multiclase (Versión 2)

Las evaluaciones del modelo anterior no obtuvieron los mejores valores de predicción. Es por ello que, este modelo unifica las fallas D1 y D2 por D, además elimina las 4 instancias correspondientes a la falla de tipo T1 C, ya que no se considera significativa la cantidad de etiquetas presentes en esta base de datos. De tal modo que, siguiendo la misma lógica, se llevó a cabo la predicción de los 463 casos y se evaluó su rendimiento contra las etiquetas reales, donde se obtuvo una exactitud resultante de 0.80 que representa la proporción de predicciones correctas frente al total. Mientras, que la métrica Balanced Accuracy tiene en cuenta la distribución de las clases, es decir, proporciona una medida más equitativa del rendimiento del modelo, el cual obtuvo 0.67.

5.2.3.1 Matriz de confusión

En la matriz de confusión (Figura 5.13) se puede observar que el modelo predijo correctamente 371 casos de 463: donde la clase 2, 3 y 6 son las de mejor rendimiento. La clase 7 es clasificada 12 veces como clase 3, 9 como clase 1 y 8 como clase 2. Mientras la clase 3 es clasificada 10 veces como clase 7. Por otra parte, la clase 5 es clasificada erróneamente en todas las instancias, siendo clasificadas en su mayoría como clase 8. Estos errores se deben a que existen reglas de inferencia que coinciden con más de una etiqueta.

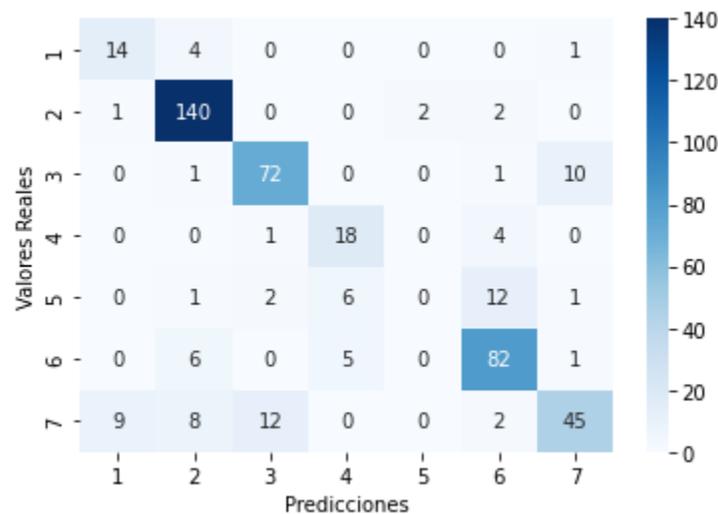


Figura 5.13. Matriz de confusión de los valores de predicción versus las etiquetas reales del modelo de LD multifalla 2.

En la Tabla 5.4 se muestran los resultados obtenidos de las métricas para cada modelo de LD evaluado. Como se puede observar, el modelo binario obtuvo las mejores evaluaciones por parte de las métricas probadas, sobrepasando el 90% exactitud incluso en la exactitud balanceada, lo cual indica la facilidad para el modelo de clasificar con alta precisión entre una condición normal y una falla presente en los transformadores de potencia. Los valores obtenidos para el modelo multifalla presentaron muchas limitaciones como se vio anteriormente, su baja precisión de clasificación se vio reflejada en las matrices de confusión donde el modelo presentó problemas al momento de clasificar ciertas clases. La versión mejorada del modelo multifalla siendo la versión 2, mostró una mejoría en su precisión general y en la mayoría de las métricas. Sin embargo, a pesar de los ajustes realizados en su modelación la exactitud balanceada es un poco limitada.

Tabla 5.4. Resultados obtenidos por las diferentes métricas aplicadas a los modelos de LD.

Modelo de LD	Accuracy score	Balanced Accuracy	C.C. Matthews	Kappa de Cohen	C. de Jaccard
Binario	0.96	0.97	0.92	0.92	0.92
Multifalla	0.65	0.63	0.60	0.60	0.49
Multifalla version 2	0.8	0.68	0.74	0.74	0.66

5.2. Conclusión

El estudio abordó el diagnóstico de transformadores de potencia utilizando tanto modelos de machine learning como lógica difusa. Los resultados indicaron que el algoritmo K-Nearest Neighbors (K-NN) destacó en la clasificación de clases desbalanceadas, con un enfoque en la precisión equilibrada. Sin embargo, se observaron limitaciones en la identificación de ciertos tipos de fallas, especialmente en las relacionadas con carbonización.

No obstante, a pesar de los algoritmos utilizados, la naturaleza de los datos ha limitado la obtención de una exactitud superior al 70%. Esto plantea la necesidad de explorar algoritmos más complejos robustos, adaptados a la base de datos, como es el caso de algoritmos ensamblados y de mejorar

continuamente dicha base, por ejemplo, mediante el aumento de casos de fallas para lograr un equilibrio en la proporción.

Por otro lado, los modelos de lógica difusa mostraron una alta precisión en la clasificación binaria entre condición normal y falla (Tabla 5.5). Sin embargo, al expandir el modelo para predecir el tipo específico de falla, surgieron desafíos, como la falta de reglas de inferencia y clasificaciones incorrectas.

Tabla 5.5. Resultados obtenidos por las diferentes métricas aplicadas a los modelos de LD.

Modelo	Accuracy score	Balanced Accuracy	C.C. Matthews	Kappa de Cohen	C. de Jaccard
SVC	0.71	0.51	0.65	0.65	0.56
Naive Bayes	0.58	0.56	0.52	0.51	0.41
Linear Discriminant Analysis	0.61	0.55	0.57	0.54	0.44
Quadratic Discriminant Analysis	0.71	0.65	0.67	0.66	0.56
Decision Tree	0.66	0.48	0.61	0.60	0.50
Random Forest Classifier	0.78	0.53	0.73	0.72	0.64
Gradient Boosting Classifier	0.82	0.64	0.79	0.78	0.70
K-Nearest Neighbors	0.75	0.67	0.70	0.70	0.61
Neural Network	0.68	0.43	0.61	0.59	0.52
XGBoost	0.84	0.66	0.81	0.81	0.74
K-Nearest Neighbors Optimizado	0.77	0.70	0.72	0.71	0.62
XGBoost Optimizado	0.84	0.66	0.81	0.81	0.73
Binario	0.96	0.97	0.92	0.92	0.92
Multifalla	0.65	0.63	0.60	0.60	0.49
Multifalla version 2	0.8	0.68	0.74	0.74	0.66

El modelo de lógica difusa de clasificación binaria destacó al obtener resultados sobresalientes, con una precisión general y balanceada superior al 95%, quiere decir, que es muy probable que el modelo clasifique correctamente el estado normal de un transformador o bien si este está sufriendo algún tipo de falla. Por otra parte, el modelo de clasificación multiclase no obtuvo la mejor evaluación en cuanto a las métricas, siendo la precisión general 0.65 frente a 0.63 de la balanceada. Esto es debido a que el modelo confunde muchas etiquetas, es decir, clasifica ciertas fallas con otras a las que no corresponden las instancias de predicción. También, no fue capaz de clasificar correctamente una de las fallas. Todo esto es debido a que ciertas reglas de inferencia coinciden

con varias clases y por lo tanto el modelo toma las de mayor peso. No obstante, también es debido a que cuando los datos de entrada se encuentran cerca o dentro de los límites que diferencian a una clase de la otra los vuelve muy complicados predecirlos con este tipo de clasificación.

Debido a esto, se planteó un arreglo diferente de la variable de salida con tal de obtener mejores resultados de predicción, donde se unió la falla D1 y D2 para establecer a D. Estas fallas por si solas brindaron más de 40 errores en la clasificación. Además, se optó por no incluir a la falla T1 C debido a que solo se contaba con 4 instancias para predecir, aunado a que las reglas de inferencia pertenecientes a esa etiqueta coincidían con la de otras etiquetas que tenían mayor peso. La evaluación de este modelo obtuvo una precisión general del 80% y balanceada del 68%, quiere decir, que en general, el modelo clasifica correctamente alrededor del 80% de los datos 6 de las 7 fallas establecidas de un transformador.

Dentro del estado del arte, los métodos más populares y eficaces para el diagnóstico de fallas, son el método de Duval, Rogers entre otros. El método de Duval es el más preciso, con una eficiencia superior al 90%, sin embargo, este método considera varios aspectos, como para obtener una de las 6 fallas de las regiones del triángulo se deben obtener concentraciones en ppm de 3 de los 5 gases ya antes mencionados. Posteriormente Duval consideró otras fallas que están relacionados entre sí pero que requieren de otro gas, por lo que implementó otros 2 triángulos. Más adelante, más fallas se fueron presentado por lo que los triángulos no eran suficientes para diagnosticar todas las fallas, por lo que creó el Pentágono de Duval y a su vez un segundo pentágono. Triángulos y Pentágonos en conjunto es posible diagnosticar más de 10 fallas con alta precisión.

Los métodos de Duval no son métodos lineales, lo cual lo convierte en un desafío incluso para la IA el poder encontrar patrones de clasificación. La base de datos de este estudio fue probada en los algoritmos creados por el Dr. Duval, los cuales coinciden perfectamente en las variables de salida. Es decir, las fallas fueron obtenidas mediante el método de Duval, mientras que los modelos de ML crean patrones de acuerdo con la información que se le brinda y como se vio en las matrices de confusión, los modelos tuvieron cierto grado de confusión con una clase respecto a la otra, es decir, que los valores de entrada clasificaban que pertenecían a otra clase y viceversa.

Por otra parte, los modelos de LD fueron creados por medio de la experiencia obtenida por los intervalos proporcionados por organizaciones internacionales como la IEEE y por las relaciones brindadas por el método de Rogers. Este último, bibliográficamente brinda un 70% de predicción

correcta en sus relaciones. Sin embargo, Duval, no brinda intervalos, codificaciones o arreglos que se tomen como base de experiencia para el planteamiento de reglas de inferencia para la modelación de LD. Siendo así, que los resultados obtenidos en este estudio, son considerablemente acordes y aceptables en la comparativa de que se basaron en la experiencia obtenida por métodos menos precisos como el método de Rogers pero que tales modelos brindan un arreglo que permite reducir los rangos de valores de una gama extensa de posibilidades por parte de los gases disueltos en aceite.

En conclusión, el diagnóstico de fallas de esta base de datos se ajusta muy bien al método de Duval (triángulos y pentágonos), sin embargo, se vuelve muy complejo encontrar patrones o bien intervalos que permitan crear reglas de inferencia y modelar en LD, así como también crear patrones de clasificación más precisos para el caso de modelación en ML. Los algoritmos de ML se ajustaron lo más que pudieron para crear patrones con el menor error posible de clasificación, pero la naturaleza de la base de datos es lo que vuelve complicado la clasificación. En cuanto a los modelos de LD se volvía muy complicado crear reglas de inferencia solo por medio de las concentraciones en ppm de los 5 gases, ya que el universo de posibilidades con bastante incertidumbre se hacía presente, por lo que se recurrió a la experticia obtenida por el método de Rogers el cual brindaba un conjunto reducido de relaciones el cual disminuía considerablemente los conjuntos difusos que fueron establecidos. Si bien, los resultados mostraron que sigue existiendo cierto grado de confusión entre clases, esto es debido de igual forma a la naturaleza de los datos. Los resultados obtenidos, demuestran que el método de Duval es un método con gran robustez que, si bien fue de los primeros en establecerse, este se ha ido actualizando con el paso de los años creando complementos como lo son más triángulos y pentágonas que en conjunto brindan más opciones y una mayor precisión en el diagnóstico de fallas de transformadores de potencia, pero este desarrollado se ha logrado con años de estudios y pruebas. Incluso con la inteligencia artificial y los estudios previos realizados con esta, ha demostrado buenos resultados, pero en conjunto con otras técnicas o bien con más parámetros considerados dentro de sus modelaciones, es decir, se consideran parámetros como lo son la longevidad del transformador, análisis de otros gases o técnicas como lo son análisis de furanos etc., lo que lo vuelve un estudio con más recursos para ser analizados y obtener resultados más favorables.

La combinación de enfoques de machine learning y lógica difusa proporcionó una comprensión más completa del estado de los transformadores de potencia. A pesar de los resultados prometedores, se destaca la necesidad de mejoras continuas y optimización, especialmente para abordar las dificultades en la identificación de ciertos tipos de fallas. Este estudio subraya la importancia de la investigación continua para perfeccionar las técnicas de diagnóstico y garantizar una evaluación precisa y confiable de la salud de los transformadores de potencia.

Conclusiones

Actualmente, el empleo de técnicas de IA en diversos ámbitos del desarrollo de proyectos ha experimentado un notable aumento debido a los buenos resultados que se han obtenido. El diagnóstico de fallas se presenta como un tema idóneo para la aplicación de estas técnicas, como evidencian estudios previos que han logrado modelos inteligentes integrales con una buena precisión en la predicción de fallos en transformadores de potencia.

El desarrollo de un modelo inteligente para el diagnóstico de fallas en transformadores de potencia, apoyado en técnicas de inteligencia artificial, ha demostrado ser bueno en la identificación de fallas en estos equipos. Este enfoque confirma la hipótesis inicial de que la aplicación de IA puede proporcionar un soporte decisivo para la gestión de transformadores, reduciendo los riesgos y daños asociados a fallas no detectadas.

El éxito de los modelos de diagnóstico de fallas en transformadores de potencia basados en inteligencia artificial depende en gran medida de la calidad y cantidad de datos disponibles. La recopilación exhaustiva de datos históricos y actuales de los transformadores, así como la constante actualización de esta base de datos, son fundamentales para mejorar la precisión y confiabilidad de los modelos.

Por otra parte, los modelos de aprendizaje automático (ML) han demostrado ser herramientas poderosas en el diagnóstico de fallas en transformadores de potencia. Desde algoritmos como K-Nearest Neighbors (K-NN) hasta redes neuronales artificiales, estos modelos son capaces de identificar patrones complejos en los datos y predecir con precisión el estado de los

transformadores. Sin embargo, es crucial seleccionar y optimizar adecuadamente estos modelos para adaptarse a las características específicas de los datos y mejorar su desempeño.

La lógica difusa (LD) emerge como una técnica valiosa en el diagnóstico de fallas en transformadores de potencia, especialmente en la clasificación y evaluación de la salud de estos equipos. Los modelos basados en LD ofrecen una interpretación más transparente y comprensible en los resultados, permitiendo una toma de decisiones más informada por parte de los expertos en el campo. La inclusión de modelos de LD junto con modelos de ML puede mejorar la robustez y versatilidad del sistema de diagnóstico.

La comparación entre los modelos tradicionales de diagnóstico de fallas y los modelos basados en inteligencia artificial destaca la superioridad de estos últimos en términos de precisión y eficiencia. Si bien los métodos tradicionales como el análisis de gases disueltos en aceite han sido útiles, los modelos de IA aprovechan al máximo la gran cantidad de datos disponibles y son capaces de identificar patrones sutiles que pueden pasar desapercibidos para los enfoques convencionales. La integración de modelos de IA en la gestión de activos eléctricos promete mejorar significativamente la confiabilidad y la eficiencia de los sistemas eléctricos.

Recomendaciones

Los modelos desarrollados cumplen con lo planteado desde los inicios, predecir la condición en la que se encuentra los transformadores de potencia. Sin embargo, es importante reconocer las limitaciones de esta investigación con el propósito de mejorar los modelos de diagnósticos de fallas de transformadores de potencia para investigaciones futuras. Basado en las conclusiones y resultados obtenidos, se proponen las siguientes recomendaciones para trabajos futuros:

1. Mejora de la base de datos

- Ampliar y mejorar la base de datos mediante la recopilación de más datos históricos y actuales de transformadores de potencia.
- Incluir datos de fallas con menor incidencia o datos relacionados con fallas que no se hayan considerado en el estudio actual, basándose en el conocimiento experto del dominio.

2. Reducción del número de clases:
 - Considerar la posibilidad de reducir el número de clases (tipos de fallas) para simplificar el problema de clasificación y mejorar la precisión de los modelos.
 - Integrar o agrupar clases de fallas con características similares o menos frecuentes, manteniendo un enfoque en las fallas más relevantes y críticas para el funcionamiento del transformador.
3. Exploración de algoritmos de ML:
 - Investigar y comparar el desempeño de algoritmos de aprendizaje automático adicionales en la clasificación binaria de fallas.
 - Explorar la viabilidad y eficacia de algoritmos ensamblados para mejorar la precisión y robustez de modelos de clasificación binaria
4. Implementación de clasificación de fallas después de la detección:
 - Diseñar y evaluar un enfoque que combine la detección de la presencia de falla (clasificación binaria) con la clasificación de la falla específica.
 - Utilizar algoritmos especializados para la clasificación de fallas una vez que se haya detectado la presencia de una anomalía en el transformador, lo que permitiría una respuesta más rápida y precisa a las condiciones de operación anormales.
5. Consulta a expertos para modelación de LD
 - La intervención de expertos y discutir si los parámetros podrían establecerse de diferente manera, ya sea en los conjuntos difusos, tipo de función de membresía etc.

Referencias bibliográficas

- Álvarez, O., Ramirez, F., & Fernández, S. (2011). Sistema para diagnóstico de fallas en transformadores a través de lógica difusa.
- Argumedo, M. J. (2019). Despliegue en Hardware de un Sistema de Diagnóstico Asistido por Computadora basado en Deep Learning para la clasificación y detección de células de cáncer cervicouterino en un examen de Papanicolau.
- Azcárraga Ramos, C., Liñan García, R., Nava Guzmán, J., & Ramírez Niño, J. (2014). *Procedimientos de evaluación de la condición de transformadores de potencia y subestaciones aisladas en gas.*
- Beunza Buini, J. J., Puertas Sanz, E., & Condés Moreno, E. (2019). Manual práctico de Inteligencia Artificial en entornos sanitarios . *ELSEVIER.*
- Bobadilla , J. (2020). *Machine Learning y Deep Learning usando Python, Scikit y Keras.*
- Buitrago Ariza, J. (2023). *Comparación de Métodos de Aprendizaje Automático para la Clasificación de transacciones fraudulentas.* Universidad de los Andes.
- Cadena, J., Cadena, J., & Pérez Londoño, S. (2008). Aplicación de redes neuronales probabilísticas en la detección de fallas incipientes en transformadores.
- Calixto, E. (2016). *PREDICTIVA 21*. Obtenido de <https://predictiva21.com/inteligencia-artificial-aprendizaje-automatico/>
- Cerón, A. F., Echeverry, D. F., Aponte, G., & Romero, A. A. (2015). Índice de Salud para Transformadores de Potencia Inmersos en Aceite Mineral con Voltajes entre 69kV y 23vK usando Lógica Difusa.
- Cerón, A., Orduña, I., Aponte, G., & Romero, A. A. (2015). Panorama de la Gestión de Activos para Transformadores de Potencia. *SCIELO.*
- CFE. (2019). *Programa de Ampliación y Modernización de las Redes Generales de Distribución 2019-2033.*

- Cheim, L. (2018). Machine Learning Tools in Support of Transformer Diagnostics.
- Chilaka, R., & Ashwani Kumar, C. (2017). Expert System for Health Index Assessment of Power Transformers. *Electrical Engineering and Informatics*, 16.
- Contreras, L., Fuentes, H., & Rodríguez, J. (2020). Predicción del rendimiento académico como indicador de éxito/fracaso de los estudiantes de ingeniería, mediante aprendizaje automático. *SCIELO*.
- Delgado, M., Paz, F., & Tupia, M. (2020). Sistemas de Lógica Difusa para la Evaluación de Usabilidad de Sitios Web de Gobierno Electrónico_ Una Revisión Sistemática. *risti*, 143.
- Ehsani, R., & Drablos, F. (2020). Robust Distance Measures for KNN Classification of Cancer Data.
- Energy D. (Office of Electric Delivery and Energy Reliability). (2014). Large Power Transformers and the U.S Electric Grid. United States.
- Fernández Blanco, J., Hernández, F., & Corrales Barrios, L. (2021). Método de lógica difusa para el diagnóstico de fallos incipientes en un transformador de 40 MVA.
- IBM. (2019). Obtenido de <https://www.ibm.com/mx-es/topics/random-forest>
- IEEE C57.152. (2013). Institute of Electrical and Electronics Engineers. *Guides for Diagnostic Field Testing of Fluid-Filled Power Transformers REGulators and Reactors*. New York, United States.
- Lévano Rodríguez, D., & Cerdán León, F. (2022). Discriminación de masas mamográficas mediante K-Nearest Neighbor y atributos BIRADS. *Revista Científica de Sistemas e Informática*.
- López Tagle, A., Equihua Tapia, H., Linán García, R., Pascacio de los Santos, A., & Nuñez Domínguez, A. (2000). Monitoreo en línea para la detección y prevención de fallas en transformadores de potencia.
- Martínez, E. (2010). *Diagnóstico y mantenimiento a transformadores en campo*. Bogotá, Colombia.

- Mileta, Z., & Zlatan, S. (2017). Analysis of artificial intelligence expert systems for power transformer condition monitoring and diagnostics. *ELSEVIER*.
- Mohanty, N. (2013). Handbook of Statistics.
- Parraga Acosta, J. C. (2019). El índice de salud y su influencia en la gestión del ciclo de vida de los transformadores del sistema eléctrico de la Mina Cobriza. Huancayo, Perú.
- Pereira Toledo, A., López Cabrera, J., & Quintero Domínguez, L. (2017). Estudio experimental para la comparación del desempeño de Naive Bayes con otros clasificadores bayesianos. *SCIELO*.
- Pérez B., R., Matos Alfonso, E., & Fernández, S. (2009). Parameter estimation and validation of power transformers top oil temperature model by applying genetic algorithms.
- Ponce Cruz, P. (2010). *INTELIGENCIA ARTIFICIAL CON APLICACIONES A LA INGENIERÍA*. México: Alfaomega.
- Ponce, P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. México: Alfaomega Grupo Editor, S. A. de C. V.
- Quintanilha, I., Elias, V., & B. da Silva, F. (2021). A fault detector/classifier for closed-ring power generators using machine learning. *ELSEVIER*.
- Raschka, S. (2018). *Machine Learning Lecture Notes*. Department of Statistics University of Wisconsin-Madison.
- Rodrigo, J. A. (Octubre de 2020). *cienciadedatos.com*. Obtenido de https://cienciadedatos.net/documentos/py09_gradient_boosting_python
- Rouhianen, L. (2018). INTELIGENCIA ARTIFICIAL 101 cosas que debes saber hoy sobre nuestro futuro. *alienta*.
- SageMaker, A. (2023). *Amazon SageMaker*. Obtenido de https://docs.aws.amazon.com/es_es/sagemaker/latest/dg/sagemaker-dg.pdf#xgboost
- Sarria Arias, J. T., Guerrero Bello, N. A., & Rivas Trujillo, E. (2014). Estado del arte del análisis de gases disueltos en transformadores de potencia. *SCIELO*.

scikitlearn. (2019). *scikit-learn.org*. Obtenido de <https://scikit-learn.org/stable/modules/svm.html#classification>

Sossa Azuela, J. (s.f.). El papel de la inteligencia artificial en la industria 4.0.

Xiaozhou, Y. (9 de 05 de 2020). *Towardsdatascience.com*. Obtenido de <https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b>

Zhigang , S., Guotao, W., Pengfei, L., Hui, W., Min, Z., & Xiaowen, L. (2014). An improved random forest based on the classification accuracy and correlation measurement of decision trees. *ELSEVIER*.

Anexo 1

Reglas de inferencia del modelo de LD de clasificación binaria

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Alta	Muy alta	Alta	Muy alta	Muy alta	3	Falla
Baja	Baja	Media	Media	Muy alta	3	Falla
Baja	Baja	Alta	Alta	Alta	3	Falla
Baja	Baja	Muy alta	Alta	Alta	3	Falla
Baja	Media	Muy alta	Alta	Alta	3	Falla
Baja	Media	Baja	Baja	Muy alta	3	Falla
Media	Alta	Muy alta	Alta	Muy alta	3	Falla
Baja	Media	Alta	Alta	Media	3	Falla
Alta	Alta	Muy alta	Muy alta	Baja	3	Falla
Alta	Media	Muy alta	Muy alta	Baja	3	Falla
Alta	Muy alta	Muy alta	Muy alta	Baja	3	Falla
Baja	Media	Baja	Baja	Muy alta	3	Falla
Baja	Baja	Baja	Baja	Baja	3	Falla
Media	Media	Media	Media	Media	3	Falla
Alta	Alta	Alta	Alta	Alta	3	Falla
Muy alta	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Media	Baja	Baja	Baja	Baja	3	Falla
Alta	Baja	Baja	Baja	Baja	3	Falla
Muy alta	Baja	Baja	Baja	Baja	3	Falla
Baja	Media	Baja	Baja	Baja	3	Falla
Baja	Alta	Baja	Baja	Baja	3	Falla
Baja	Muy alta	Baja	Baja	Baja	3	Falla
Baja	Baja	Media	Baja	Baja	3	Falla
Baja	Baja	Alta	Baja	Baja	3	Falla
Baja	Baja	Muy alta	Baja	Baja	3	Falla
Baja	Baja	Baja	Media	Baja	3	Falla
Baja	Baja	Baja	Alta	Baja	3	Falla
Baja	Baja	Baja	Muy alta	Baja	3	Falla
Baja	Baja	Baja	Baja	Media	3	Falla
Baja	Baja	Baja	Baja	Alta	3	Falla
Baja	Baja	Baja	Baja	Muy alta	3	Falla
Baja	Media	Media	Media	Media	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Baja	Alta	Alta	Alta	Alta	3	Falla
Baja	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Media	Alta	Alta	Alta	Alta	3	Falla
Media	Baja	Baja	Baja	Baja	3	Falla
Media	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Media	Media	Baja	Baja	Baja	3	Falla
Media	Alta	Baja	Baja	Baja	3	Falla
Media	Muy alta	Baja	Baja	Baja	3	Falla
Media	Baja	Media	Baja	Baja	3	Falla
Media	Baja	Alta	Baja	Baja	3	Falla
Media	Baja	Muy alta	Baja	Baja	3	Falla
Media	Media	Media	Baja	Baja	3	Falla
Media	Media	Alta	Baja	Baja	3	Falla
Media	Media	Muy alta	Baja	Baja	3	Falla
Media	Media	Media	Media	Baja	3	Falla
Media	Media	Alta	Media	Baja	3	Falla
Media	Media	Muy alta	Media	Baja	3	Falla
Media	Media	Media	Alta	Baja	3	Falla
Media	Media	Alta	Alta	Baja	3	Falla
Media	Media	Muy alta	Alta	Baja	3	Falla
Media	Media	Media	Muy alta	Baja	3	Falla
Media	Media	Alta	Muy alta	Baja	3	Falla
Media	Media	Muy alta	Muy alta	Baja	3	Falla
Media	Alta	Alta	Alta	Media	3	Falla
Media	Baja	Baja	Baja	Media	3	Falla
Media	Muy alta	Muy alta	Muy alta	Media	3	Falla
Media	Media	Baja	Baja	Media	3	Falla
Media	Alta	Baja	Baja	Media	3	Falla
Media	Muy alta	Baja	Baja	Media	3	Falla
Media	Baja	Media	Baja	Media	3	Falla
Media	Baja	Alta	Baja	Media	3	Falla
Media	Baja	Muy alta	Baja	Media	3	Falla
Media	Media	Media	Baja	Media	3	Falla
Media	Media	Alta	Baja	Media	3	Falla
Media	Media	Muy alta	Baja	Media	3	Falla
Media	Media	Alta	Media	Media	3	Falla
Media	Media	Media	Alta	Media	3	Falla
Media	Media	Alta	Alta	Media	3	Falla
Media	Media	Muy alta	Alta	Media	3	Falla
Media	Media	Media	Muy alta	Media	3	Falla
Media	Media	Alta	Muy alta	Media	3	Falla
Media	Media	Muy alta	Muy alta	Media	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Media	Alta	Alta	Alta	Alta	3	Falla
Media	Baja	Baja	Baja	Alta	3	Falla
Media	Muy alta	Muy alta	Muy alta	Alta	3	Falla
Media	Media	Baja	Baja	Alta	3	Falla
Media	Alta	Baja	Baja	Alta	3	Falla
Media	Muy alta	Baja	Baja	Alta	3	Falla
Media	Baja	Media	Baja	Alta	3	Falla
Media	Baja	Alta	Baja	Alta	3	Falla
Media	Baja	Muy alta	Baja	Alta	3	Falla
Media	Media	Media	Baja	Alta	3	Falla
Media	Media	Alta	Baja	Alta	3	Falla
Media	Media	Muy alta	Baja	Alta	3	Falla
Media	Media	Media	Media	Alta	3	Falla
Media	Media	Alta	Media	Alta	3	Falla
Media	Media	Muy alta	Media	Alta	3	Falla
Media	Media	Media	Alta	Alta	3	Falla
Media	Media	Alta	Alta	Alta	3	Falla
Media	Media	Muy alta	Alta	Alta	3	Falla
Media	Media	Media	Muy alta	Alta	3	Falla
Media	Media	Alta	Muy alta	Alta	3	Falla
Media	Media	Muy alta	Muy alta	Alta	3	Falla
Media	Alta	Alta	Alta	Muy alta	3	Falla
Media	Baja	Baja	Baja	Muy alta	3	Falla
Media	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Media	Media	Baja	Baja	Muy alta	3	Falla
Media	Alta	Baja	Baja	Muy alta	3	Falla
Media	Muy alta	Baja	Baja	Muy alta	3	Falla
Media	Baja	Media	Baja	Muy alta	3	Falla
Media	Baja	Alta	Baja	Muy alta	3	Falla
Media	Baja	Muy alta	Baja	Muy alta	3	Falla
Media	Media	Media	Baja	Muy alta	3	Falla
Media	Media	Alta	Baja	Muy alta	3	Falla
Media	Media	Muy alta	Baja	Muy alta	3	Falla
Media	Media	Media	Media	Muy alta	3	Falla
Media	Media	Alta	Media	Muy alta	3	Falla
Media	Media	Muy alta	Media	Muy alta	3	Falla
Media	Media	Media	Alta	Muy alta	3	Falla
Media	Media	Alta	Alta	Muy alta	3	Falla
Media	Media	Muy alta	Alta	Muy alta	3	Falla
Media	Media	Media	Muy alta	Muy alta	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Media	Media	Alta	Muy alta	Muy alta	3	Falla
Media	Media	Muy alta	Muy alta	Muy alta	3	Falla
Media	Alta	Alta	Alta	Alta	3	Falla
Media	Baja	Baja	Alta	Alta	3	Falla
Media	Muy alta	Muy alta	Muy alta	Alta	3	Falla
Media	Media	Baja	Alta	Alta	3	Falla
Media	Alta	Baja	Alta	Alta	3	Falla
Media	Muy alta	Baja	Alta	Alta	3	Falla
Media	Baja	Media	Alta	Alta	3	Falla
Media	Baja	Alta	Alta	Alta	3	Falla
Media	Baja	Muy alta	Alta	Alta	3	Falla
Media	Media	Media	Alta	Alta	3	Falla
Media	Media	Alta	Alta	Alta	3	Falla
Media	Media	Muy alta	Alta	Alta	3	Falla
Media	Media	Media	Media	Alta	3	Falla
Media	Media	Alta	Media	Alta	3	Falla
Media	Media	Muy alta	Media	Alta	3	Falla
Media	Media	Media	Alta	Alta	3	Falla
Media	Media	Alta	Alta	Alta	3	Falla
Media	Media	Muy alta	Alta	Alta	3	Falla
Media	Media	Media	Muy alta	Alta	3	Falla
Media	Media	Alta	Muy alta	Alta	3	Falla
Media	Media	Muy alta	Muy alta	Alta	3	Falla
Alta	Muy alta	Alta	Alta	Alta	3	Falla
Alta	Alta	Muy alta	Alta	Alta	3	Falla
Alta	Muy alta	Muy alta	Alta	Alta	3	Falla
Alta	Muy alta	Muy alta	Muy alta	Alta	3	Falla
Alta	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Alta	Muy alta	Alta	Alta	Muy alta	3	Falla
Alta	Alta	Muy alta	Alta	Muy alta	3	Falla
Alta	Muy alta	Muy alta	Alta	Muy alta	3	Falla
Muy alta	Muy alta	Alta	Alta	Alta	3	Falla
Muy alta	Alta	Muy alta	Alta	Alta	3	Falla
Muy alta	Muy alta	Muy alta	Alta	Alta	3	Falla
Muy alta	Muy alta	Muy alta	Muy alta	Alta	3	Falla
Muy alta	Muy alta	Muy alta	Muy alta	Muy alta	3	Falla
Muy alta	Muy alta	Alta	Alta	Muy alta	3	Falla
Muy alta	Alta	Muy alta	Alta	Muy alta	3	Falla
Muy alta	Muy alta	Muy alta	Alta	Muy alta	3	Falla
Media	Media	Baja	Media	Muy alta	3	Falla
Baja	Alta	Alta	Muy alta	Muy alta	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Baja	Media	Alta	Muy alta	Alta	3	Falla
Baja	Media	Alta	Muy alta	Muy alta	3	Falla
Baja	Baja	Baja	Muy alta	Media	3	Falla
Baja	Baja	Baja	Muy alta	Alta	3	Falla
Baja	Baja	Baja	Alta	Alta	3	Falla
Media	Media	Baja	Muy alta	Muy alta	3	Falla
Baja	Baja	Alta	Muy alta	Alta	3	Falla
Baja	Baja	Alta	Muy alta	Media	3	Falla
Baja	Media	Baja	Muy alta	Alta	3	Falla
Media	Alta	Baja	Muy alta	Muy alta	3	Falla
Baja	Alta	Baja	Muy alta	Muy alta	3	Falla
Alta	Media	Muy alta	Alta	Muy alta	3	Falla
Baja	Baja	Alta	Alta	Muy alta	3	Falla
Media	Alta	Muy alta	Alta	Alta	3	Falla
Baja	Media	Alta	Baja	Baja	3	Falla
Baja	Baja	Alta	Baja	Media	3	Falla
Media	Baja	Muy alta	Alta	Baja	3	Falla
Alta	Media	Muy alta	Alta	Baja	3	Falla
Muy alta	Media	Muy alta	Alta	Baja	3	Falla
Muy alta	Alta	Muy alta	Muy alta	Muy alta	3	Falla
Alta	Media	Muy alta	Alta	Media	3	Falla
Media	Baja	Media	Media	Muy alta	3	Falla
Baja	Media	Alta	Alta	Muy alta	3	Falla
Baja	Baja	Alta	Media	Alta	3	Falla
Media	Media	Baja	Media	Baja	3	Falla
Alta	Media	Alta	Muy alta	Muy alta	3	Falla
Media	Alta	Media	Muy alta	Muy alta	3	Falla
Media	Baja	Media	Muy alta	Muy alta	3	Falla
Alta	Alta	Baja	Baja	Baja	3	Falla
Media	Alta	Alta	Media	Alta	3	Falla
Media	Alta	Muy alta	Media	Baja	3	Falla
Media	Alta	Muy alta	Media	Media	3	Falla
Baja	Baja	Media	Baja	Muy alta	3	Falla
Baja	Media	Media	Alta	Muy alta	3	Falla
Baja	Media	Alta	Media	Muy alta	3	Falla
Alta	Muy alta	Media	Muy alta	Muy alta	3	Falla
Media	Media	Alta	Baja	Muy alta	3	Falla
Baja	Media	Media	Media	Muy alta	3	Falla
Media	Alta	Baja	Alta	Muy alta	3	Falla
Baja	Baja	Alta	Baja	Muy alta	3	Falla
Baja	Baja	Media	Baja	Media	3	Falla
Baja	Alta	Media	Alta	Muy alta	3	Falla
Baja	Media	Media	Baja	Muy alta	3	Falla
Baja	Alta	Muy alta	Muy alta	Muy alta	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Alta	Alta	Media	Alta	Muy alta	3	Falla
Baja	Media	Alta	Baja	Muy alta	3	Falla
Baja	Media	Baja	Media	Muy alta	3	Falla
Muy alta	Muy alta	Baja	Baja	Baja	3	Falla
Muy alta	Alta	Media	Baja	Baja	3	Falla
Muy alta	Muy alta	Alta	Baja	Muy alta	3	Falla
Muy alta	Alta	Baja	Baja	Baja	3	Falla
Media	Baja	Muy alta	Media	Baja	3	Falla
Muy alta	Alta	Muy alta	Alta	Baja	3	Falla
Alta	Alta	Muy alta	Muy alta	Muy alta	3	Falla
Alta	Baja	Muy alta	Media	Baja	3	Falla
Muy alta	Muy alta	Muy alta	Muy alta	Baja	3	Falla
Muy alta	Muy alta	Muy alta	Muy alta	Media	3	Falla
Muy alta	Alta	Muy alta	Muy alta	Baja	3	Falla
Media	Baja	Muy alta	Alta	Media	3	Falla
Alta	Media	Muy alta	Muy alta	Alta	3	Falla
Media	Alta	Muy alta	Muy alta	Muy alta	3	Falla
Baja	Baja	Alta	Media	Baja	3	Falla
Media	Baja	Baja	Alta	Baja	3	Falla
Alta	Alta	Muy alta	Muy alta	Media	3	Falla
Muy alta	Alta	Muy alta	Muy alta	Muy alta	3	Falla
Baja	Media	Media	Alta	Baja	3	Falla
Baja	Baja	Muy alta	Muy alta	Baja	3	Falla
Baja	Media	Baja	Alta	Baja	3	Falla
Media	Alta	Alta	Muy alta	Media	3	Falla
Media	Media	Baja	Muy alta	Baja	3	Falla
Alta	Alta	Muy alta	Muy alta	Alta	3	Falla
Muy alta	Muy alta	Alta	Alta	Muy alta	3	Falla
Media	Baja	Media	Media	Baja	3	Falla
Alta	Media	Alta	Alta	Baja	3	Falla
Muy alta	Media	Muy alta	Muy alta	Muy alta	3	Falla
Baja	Media	Muy alta	Media	Muy alta	3	Falla
Baja	Alta	Baja	Media	Muy alta	3	Falla
Baja	Muy alta	Media	Muy alta	Muy alta	3	Falla
Baja	Baja	Baja	Media	Muy alta	3	Falla
Alta	Alta	Muy alta	Alta	Media	3	Falla
Baja	Media	Muy alta	Alta	Baja	3	Falla
Baja	Media	Baja	Muy alta	Baja	3	Falla
Baja	Baja	Muy alta	Media	Baja	3	Falla
Baja	Media	Alta	Muy alta	Baja	3	Falla
Media	Muy alta	Alta	Muy alta	Muy alta	3	Falla

H ₂	CH ₄	C ₂ H ₂	C ₂ H ₄	C ₂ H ₆		Diagnóstico
Baja	Baja	Baja	Media	Baja	3	Falla
Media	Alta	Muy alta	Muy alta	Muy alta	3	Falla
Baja	Media	Baja	Alta	Baja	3	Falla
Alta	Muy alta	Baja	Baja	Media	3	Falla
Muy alta	Alta	Alta	Baja	Baja	3	Falla
Muy alta	Media	Media	Baja	Baja	3	Falla
Muy alta	Media	Muy alta	Baja	Media	3	Falla

Anexo 2

Reglas de inferencia del modelo de LD de clasificación de fallas

	C ₂ H ₂ /C ₂ H ₄	C ₂ H ₄ /C ₂ H ₆		Diagnóstico
Media	Muy alta	Baja	1	PD
Baja	Baja	Muy alta	1	PD
Media	Alta	Media	2	D
Baja	Media	Muy alta	2	D
Alta	Alta	Alta	2	D
Baja	Alta	Alta	2	D
Media	Muy alta	Media	2	D
Media	Muy alta	Baja		D1
Baja	Alta	Baja	2	D
Baja	Muy alta	Muy alta	2	D
Alta	Media	Muy alta	2	D
Alta	Alta	Muy alta	2	D
Alta	Media	Media	3	T1 O
Alta	Alta	Media	3	T1 O
Muy alta	Muy alta	Media	3	T1 O
Muy alta	Media	Media	3	T1 O
Alta	Media	Baja	3	T1 O
Muy alta	Media	Baja	3	T1 O
Media	Baja	Media	7	S
Muy alta	Media	Alta	4	T2 C
Alta	Media	Alta	4	T2 C
Media	Baja	Muy alta	6	T3 H
Media	Baja	Alta	6	T3 H
Media	Baja	Baja	7	S
Alta	Muy alta	Baja	7	S
Baja	Baja	Alta	7	S

Anexo 3

Reglas de inferencia del modelo de LD de clasificación de fallas (Versión 2)

$4/H_2$	C_2H_2/C_2H_4	C_2H_4/C_2H_6	Diagnóstico	
Baja	Baja	Muy alta	1	PD
Media	Alta	Media	2	D1
Baja	Media	Muy alta	2	D1
Alta	Alta	Alta	2	D1
Baja	Alta	Alta	2	D1
Media	Muy alta	Media	2	D1
Media	Muy alta	Baja		D1
Baja	Alta	Muy alta	2	D1
Baja	Muy alta	Muy alta	2	D1
Alta	Media	Muy alta	2	D2
Alta	Alta	Muy alta	2	D2
Alta	Media	Media	3	T1 O
Alta	Alta	Media	3	T1 O
Muy alta	Muy alta	Media	3	T1 O
Muy alta	Media	Media	3	T1 C
Alta	Media	Baja	3	T1 O
Muy alta	Media	Baja	3	T1 O
Media	Baja	Media	7	S
Muy alta	Media	Alta	4	T2 C
Alta	Media	Alta	4	T2 C
Media	Baja	Muy alta	6	T3 H
Media	Baja	Alta	6	T3 H
Media	Baja	Baja	7	S
Alta	Muy alta	Baja	7	S
Baja	Baja	Alta	7	S