



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
MAESTRÍA EN OPTIMIZACIÓN Y CÓMPUTO APLICADO

**Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones
al Comportamiento Normal Usando Modelos Probabilistas**

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN
EN OPTIMIZACIÓN Y CÓMPUTO APLICADO

PRESENTA:

URIEL ALEJANDRO GARCÍA LÓPEZ

DIRECTOR DE TESIS:

DRA. LORENA DÍAZ GONZÁLEZ

CO-ASESOR DE TESIS:

DR. PABLO HÉCTOR IBARGÜENGOYTIA GONZÁLEZ

CUERNAVACA, MORELOS.

AGOSTO 2019.

A mi hijo Mateo Gabriel García Evangelista, quien me enseñó un claro ejemplo de luchar día a día por alcanzar un sueño, con esfuerzo, interés y constancia.

Quien me enseñó que un verdadero guerrero es quien a pesar de una derrota se levanta una y otra vez para salir vencedor.

Agradecimientos

En especial al Dr. Pablo Héctor Ibargüengoytia González por su confianza y apoyo incondicional para impulsarme en el mundo de la investigación. Por ser mi mentor durante mi formación profesional y académica. Por los conocimientos, apoyo y amistad otorgados cuando más lo necesité.

A mi directora de tesis la Dra. Lorena Díaz González por el apoyo, paciencia, comentarios y conocimientos brindados.

Al Dr. Jorge Hermsillo Valadez por su apoyo y conocimiento brindados.

Al CEMIE-Eólico del Fondo CONACYT-SENER- Sustentabilidad y al proyecto 14629/P12 del INEEL. Por darme la oportunidad de formar parte del grupo de investigadores.

Al Instituto Nacional de electricidad y Energías Limpias por el apoyo brindado durante mi proceso de formación profesional.

Al Ing. Fortino Mejía Neri del INEEL por su apoyo con el programa de simulación Focus6, de igual manera al Maestro Miguel Ángel Alvarado Morales.

A la Gerencia de Energías Renovables del INEEL por el apoyo brindado durante el proceso de experimentación.

A la Gerencia de Control Electrónica y Comunicaciones por el apoyo incondicional durante mi formación profesional y académica. En forma personal al MC. Julio Cesar Montero Cervantes por darme la oportunidad de crecer profesionalmente.

A la universidad Autónoma del Estado de Morelos por darme la oportunidad de llevar a cabo mi trabajo de investigación.

A mi Esposa Dra. América Libertad Bermúdez Salgado e hija Lya Montserrat Yépez Bermúdez por su amor, apoyo y paciencia incondicional proporcionada.

Índice general

Dedicatoria	I
Agradecimientos	II
Índice general	III
Índice de figuras	VI
Índice de tablas	VIII
Resumen	1
Capítulo 1	3
Introducción	3
1.1 Antecedentes	3
1.2 Motivación	4
1.3 Planteamiento de problema	6
1.4 Objetivo general	9
1.5 Objetivos específicos	9
1.6 Metodología propuesta	10
1.7 Organización de la tesis	11
Capítulo 2	12
Aerogeneradores	12
2.1 Funcionamiento del aerogenerador	12
2.2 Componentes externos	14
2.3 Componentes internos	16
Capítulo 3	18
Sistema de diagnóstico de aerogeneradores (DxAG)	18
3.1 Descripción general del sistema DxAG	18
3.2 Redes bayesianas	20

3.3	Metodología para crear modelos de comportamiento	22
Capítulo 4	26
Experimentos con simulador	26
4.1	Descripción de la Maquina Eólica Mexicana (MEM)	26
4.2	Descripción del simulador Focus6	29
4.3	Catálogo de fallas de la MEM:	30
4.3.1	Sistema de control de <i>Yaw</i>	31
4.3.2	Sistema de control de <i>pitch</i>	32
4.3.3	Cortocircuito	33
4.3.4	Sensor de dirección y velocidad de viento	33
4.3.5	Descripción del catálogo de fallas	34
4.4	Descripción del modelo de comportamiento de la MEM	35
4.4.1	Configuración del modelo de comportamiento con producción normal	36
4.4.2	Modelo de comportamiento con el sistema DxAG	38
4.5	Simulación de fallas en la MEM.....	39
4.6	Patrón de comportamiento	41
4.7	Procedimiento de extracción datos de fallas y no falla	45
Capítulo 5	48
Reconocimiento de falla raíz	48
5.1	Reconocimiento de patrones	48
5.1.1	Procedimiento de reconocimiento de patrones.....	49
5.2	Aprendizaje Automático.....	51
5.3	Principales tipos de aprendizaje automático	52
5.3.1	Aprendizaje supervisado	52

5.3.2	Aprendizaje no supervisado	55
5.3.3	Aprendizaje por refuerzo	56
5.4	Algoritmos de clasificación	57
5.4.1	<i>K-Nearest-Neighbor (K-NN)</i>	57
5.4.2	<i>Support Vector Machines</i>	59
5.4.3	<i>Naive Bayes (NB)</i>	61
5.4.4	Árboles de decisión	63
5.4.5	<i>Random Forest</i>	70
5.5	Evaluación de modelo	72
5.6	Validación cruzada	77
5.7	Experimentación.....	78
5.7.1	Formación de objetos	78
5.7.2	Reducción de dimensionalidad	81
5.7.3	Modelo de clasificación	83
Capítulo 6	102
Conclusiones y trabajo futuro	102
Bibliografía	104
Anexo	108

Índice de figuras

Figura 1: Suposición de la falla “fuga de aceite en rodamientos”	6
Figura 2: Generación de potencia respecto a la velocidad de viento del aerogenerador KW300.....	7
Figura 3: Patrón de comportamiento anormal del aerogenerador KW300	8
Figura 4: Componentes externos del aerogenerador	14
Figura 5: Sensor de velocidad y dirección de viento	15
Figura 6: Componentes externos del aerogenerador	16
Figura 7: Configuración general del DxAG.....	19
Figura 8: Representación de la Red Bayesiana	21
Figura 9: Diámetro, altura y giro del aerogenerador.....	27
Figura 10: Curva de potencia de potencia diseñada para la MEM	28
Figura 11: Estructura del funcionamiento de un aerogenerador (Wind Turbine, 2019).....	31
Figura 12: Sistema de control de pitch.....	32
Figura 13: Velocidad promedio de 12 m/s con 3 semillas diferentes.....	38
Figura 14: Modelo de comportamiento de la MEM en operación normal	39
Figura 15: Simulación de cortocircuito en la MEM	40
Figura 16: Velocidad de viento en diferentes variaciones	41
Figura 17: Serie de temporal de la variable <i>RWindDirectionHub_6</i>	42
Figura 18: Patrón de comportamiento de la F1	43
Figura 19: Patrones de comportamiento dado un catálogo de fallas	44
Figura 20: Procedimiento de extracción de datos	47
Figura 21: Procedimiento de reconocimiento de patrones	50
Figura 22: Tipos de aprendizaje	52
Figura 23: Procedimiento de aprendizaje supervisado.....	53
Figura 24: Problema de clasificación binaria y multiclase	54
Figura 25: Problema de regresión lineal	55
Figura 26: Algoritmo KNN.....	58
Figura 27: Distancia Euclidiana entre dos puntos	59

Figura 28: Algoritmo SVM.....	60
Figura 29: Árbol de decisión identificando como nodo raíz a la variable “Comunicación”	67
Figura 30: Árbol de decisión resultante.....	69
Figura 31: Idea general de <i>Bagging</i>	71
Figura 32: Validación cruzada con $k=5$	78
Figura 33: Vector de características cuantitativas.....	79
Figura 34: Conjunto de Patrones de Falla.....	80
Figura 35: Distribución de Fallas.....	81
Figura 36: Ejemplo de Selección de características.....	82
Figura 37: Características importantes	83
Figura 38: Comparación de algoritmos	86
Figura 39: Algoritmo de clasificación <i>Random Forest</i>	87
Figura 40: Matriz de confusión sin validación cruzada y sin reducción de características.....	89
Figura 41: Matriz de confusión sin validación cruzada y con reducción de características.....	91
Figura 42: Curvas ROC sin validación cruzada y con reducción de características.....	92
Figura 43: Visualización del conjunto de patrones de falla con PCA.....	94
Figura 44: Matriz de confusión. 2 PCA sin validación cruzada y sin reducción de características.....	95
Figura 45: Varianza acumulativa. El eje x, representa número de componentes y el eje y, la varianza a acumulativa.....	96
Figura 46: Matriz de confusión. 50 PCA sin validación cruzada y sin reducción de características.....	98
Figura 47: Matriz de confusión con validación cruzada y reducción de características.....	100
Figura 48: Matriz de confusión del experimento 5.....	101

Índice de tablas

Tabla 1: Catálogo de fallas insertada al aerogenerador	34
Tabla 2: Nomenclatura de un caso de carga	36
Tabla 3: Características de operación de las simulaciones para modelos de comportamiento	37
Tabla 4: Formato de la variable: velocidad de viento	46
Tabla 5: Atributos y clases para generar un árbol d decisión	65
Tabla 6: Conteo total de la clase “si” y la clase “no”	65
Tabla 7: Cálculo de la entropía de la clase Decisión dado Programación y su ganancia de información	66
Tabla 8: Cálculo de la entropía de la clase Decisión dado Comunicación y su ganancia de información	66
Tabla 9: Cálculo de la entropía de la clase Decisión dado Presentación y su ganancia de información	66
Tabla 10: Conteo total del atributo aburrido para la clase “si” y para la clase “no”	67
Tabla 11: Medidas de rendimiento	85
Tabla 12: Reporte de clasificación sin validación cruzada y sin reducción de características	88
Tabla 13: Reporte de clasificación sin validación cruzada y con reducción de características	90
Tabla 14: Reporte de clasificación. 2 PCA sin validación cruzada y sin reducción de características	95
Tabla 15: Reporte de clasificación. 50 PCA sin validación cruzada y sin reducción de características	97
Tabla 16: Reporte de clasificación con validación cruzada y reducción de características	99

Resumen

La energía eólica es una de las fuentes de energías renovables más importantes e inagotables. Esta energía se convierte en energía eléctrica a través de aerogeneradores eficientes y de gran capacidad. Las fallas en los aerogeneradores se deben a varios factores como las condiciones de viento, desgaste por envejecimiento de componentes, deficiencias de construcción, diseños no adecuados al lugar de operación, entre otros. Por lo tanto, un punto importante a considerar en los aerogeneradores es la implementación de sistemas de detección de fallas que logre mantener la confiabilidad, eficiencia, durabilidad y minimizar costos de mantenimiento o reparación.

EL Centro Mexicano de Innovación en Energía Eólica (CEMIE-Eólico) diseñó un sistema de diagnóstico de aerogeneradores basado en modelos de comportamiento de la turbina utilizando las señales del sistema de control supervisorio y adquisición de datos (SCADA por sus siglas en inglés). El sistema proporciona un patrón de variables que presentan comportamiento anormal en presencia de una falla. Los patrones se forman con la detección del comportamiento anormal de las variables durante una ventana de tiempo en que se manifiesta la falla. En este trabajo se presenta la aplicación de técnicas de aprendizaje automático para la identificación de fallas raíz en aerogeneradores después del sistema de diagnóstico. Los datos de entrenamiento y prueba se obtuvieron a partir de la simulación de seis diferentes fallas en el aerogenerador de la Máquina Eólica Mexicana (MEM) diseñada en el Instituto Nacional de Electricidad y Energías Limpias (INEEL). Se aplicó el sistema de diagnóstico, se generaron los perfiles de comportamiento anormal y se realizaron experimentos para la clasificación multiclase de patrones de fallas usando el algoritmo de *Random Forest*. Finalmente, se evaluó el desempeño del modelo de clasificación de múltiples fallas generado por el algoritmo usando métricas como la exactitud y precisión logrando como resultado un 91% de desempeño en la clasificación de patrones para identificar la falla raíz.

Palabras clave: Diagnóstico de aerogeneradores, *Random Forest*, aprendizaje automático, patrones de falla.

Capítulo 1

Introducción

1.1 Antecedentes

El trabajo de investigación que se presenta en esta tesis de posgrado forma parte del proyecto P12 del Centro Mexicano de Innovación en Energía Eólica del Fondo Sectorial SENER-Sustentabilidad y del CONACYT con título “Desarrollo de tecnología basada en inteligencia artificial y mecatrónica para integrar un parque de generación eólica a una red inteligente” y con la participación del Instituto Nacional de Electricidad y Energías Limpias y el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).

Para el INEEL, el proyecto consiste en el diseño, construcción y pruebas de un sistema de diagnóstico innovador que considera el comportamiento normal de la turbina eólica o aerogenerador y que detecta desviaciones a ese comportamiento normal utilizando técnicas de Inteligencia Artificial (IA). Para el funcionamiento de este sistema, se adquieren datos históricos del SCADA de la turbina eólica en operación normal y se genera un modelo del comportamiento normal. El modelo se construye utilizando técnicas de inteligencia artificial, específicamente redes Bayesianas (Pearl 1988). Estas redes son modelos gráficos probabilistas que representan las relaciones probabilistas entre las variables en algún dominio de aplicación. Dicha relación probabilista se utiliza en este proyecto para la representación del comportamiento de la turbina eólica en conjunto con la Teoría de Validación de Información (Ibargüengoytia, Vadera y Sucar 2006) para la detección de desviaciones al comportamiento.

Usando estos elementos, se construyó un sistema computacional para el diagnóstico de fallas en turbinas eólicas (DxAG). El sistema funciona en línea y

despliega patrones de comportamiento, los cuales representan la probabilidad de existencia de fallas en la turbina.

1.2 Motivación

Existe una necesidad urgente de realizar una transición hacia el uso de recursos renovables para la generación de potencia eléctrica a nivel mundial. Todos los países en mayor o menor grado hacen esfuerzos para lograr esa transición y México no es la excepción. Según la Asociación Mexicana de Energía Eólica (AMDEE), en el 2018 existían 2,447 aerogeneradores (AG) formando parte de 54 parques eólicos y con una generación total de 4,935 Mega Watts, resultando en un 5.08% de la producción nacional. Las turbinas son principalmente de las marcas Gamesa, Vestas, GE, Acciona, Clipper y Alstom. Adicionalmente, se cuenta con la turbina eólica (marca Komai) del INEEL instalada en el Centro Regional de Tecnología Eólica (CERTE). Los parques están en la actualidad en los estados de Baja California, Nuevo León, Tamaulipas, San Luis Potosí, Jalisco, Puebla, Oaxaca, Chiapas, Sonora, Coahuila, Durango, Zacatecas, Querétaro, Veracruz y Yucatán (AMDEE 2016)

Dada la creciente importancia de la energía eólica en la generación total nacional y dado que un AG tiene un tiempo de vida promedio entre 20 y 25 años, la necesidad de contar con sistemas de diagnóstico de aerogeneradores (AGs) resulta evidente. Lograr la identificación de fallas incipientes podrá lograr acciones de mantenimiento y reparación que permitan elevar los índices de confiabilidad y disponibilidad de los parques eólicos. La aplicación de este trabajo de investigación de identificación de fallas de turbinas eólicas al detectar desviaciones al comportamiento normal se enfoca en la energía eólica.

La literatura reporta diferentes formas de atacar el problema de diagnóstico de AGs. La mayoría de los trabajos consultados pertenecen a la comunidad de Monitoreo de la Condición (*Condition monitoring*, CM, en inglés). En el artículo de

(García Márquez, y otros 2012) se documentan los métodos más utilizados en el monitoreo de la condición o diagnóstico. Los cuales se clasifican en: i) análisis de vibraciones, ii) análisis de emisiones acústicas, iii) técnicas de pruebas ultrasónicas, iv) análisis en aceite, v) medición del esfuerzo o tensión en aspas, vi) termografía, vii) métodos de pulso de choque, viii) inspecciones radiográficas, entre otros métodos. Sin embargo, esos métodos tradicionales han requerido siempre personal experto en el área y la formación de modelos muy complejos y difíciles de construir y mantener. Alternativamente, en la actualidad existen métodos computacionales y basados en técnicas de inteligencia artificial para realizar el diagnóstico. Por ejemplo, el trabajo reportado por (Zhou, Yu y Zhang 2015) está basado en ontologías y en análisis crítico de efectos de modos de falla (FEMCA por sus siglas en inglés). Con este análisis, la metodología reportada genera una ontología y crea una base de conocimientos para alimentar un sistema experto. El problema de este método y en general con los sistemas expertos es que no manejan la incertidumbre que es común en esta aplicación. Otro trabajo reportado en (Meik, Ferreira Santos y Achiche 2013) utiliza modelos de comportamiento de la turbina y datos históricos del SCADA. El enfoque utilizado es de sistemas de inferencia neuro-difusa adaptada (ANFIS por sus siglas en inglés) pero no se tiene ninguna distinción de los modos de operación de la turbina eólica.

Este trabajo de investigación está formado por dos etapas. La primera utiliza modelos del comportamiento para poder detectar desviaciones al comportamiento normal. Los modelos de comportamiento se basan en redes Bayesianas que capturan las relaciones probabilistas entre las variables del equipo. La salida de la primera etapa consiste en un conjunto de variables que presentan un comportamiento anormal según los modelos de comportamiento generados por el sistema de diagnóstico de aerogeneradores (DxAG). La segunda etapa toma el conjunto de variables con comportamiento anormal y forma un patrón de falla identificado previamente con base en un catálogo de fallas. El aporte principal de este trabajo de investigación es la utilización de técnicas de aprendizaje de automático y de reconocimiento de patrones para identificar la falla raíz causante del patrón de comportamiento anormal de un conjunto de variables. Por ejemplo, en la

Figura 1 se presenta en forma esquemática la suposición de una fuga de aceite en rodamientos, la cual produciría un comportamiento anormal en variables como el incremento en la temperatura de las chumaceras, disminución de la velocidad angular del rotor e incremento en las señales de vibración, produciendo un patrón de falla.



Figura 1: Suposición de la falla “fuga de aceite en rodamientos”

1.3 Planteamiento de problema

Durante el proceso de modelación del comportamiento de una turbina eólica se analizaron datos históricos reales (ver Figura 2 y Figura 3) de la turbina del CERTE, tanto para entrenar como para validar el modelo. Para entrenar el modelo se emplearon datos de comportamiento normal de la turbina, es decir, sin fallas según el experto.

En la Figura 2, se presenta la velocidad del viento *WindSpeed* (en color rojo) y la potencia generada por el aerogenerador *GridActivePower* (en color azul). En donde se observan 3 momentos (enumerados como 1, 2 y 3) que identifican diferentes comportamientos de la turbina. Al inicio del último momento o

comportamiento, se observa que la generación de potencia presenta una leve caída y más adelante, se aprecia una caída total de la generación. Se puede suponer que en estos momentos sucedió algún evento que forzó al cambio de comportamiento de la turbina.

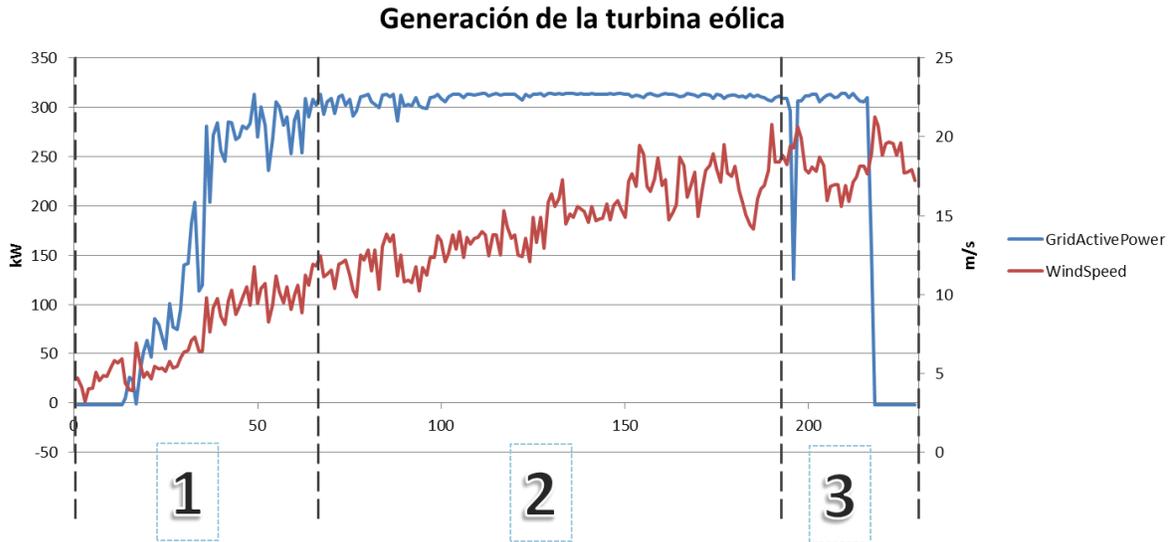


Figura 2: Generación de potencia respecto a la velocidad de viento del aerogenerador KW300

En La Figura 3, se grafican las fallas evidentes de distintas variables involucradas en la operación de la turbina eólica, las cuales presentan desviación al comportamiento normal. En el eje vertical se puede apreciar el porcentaje de falla detectada en esas variables y en el eje horizontal se observan 3 momentos enumerados, donde se observan comportamientos de la turbina. Algunas variables evidentes con comportamiento anormal son las siguientes:

- *RBearingTemp_B* (Temperatura interior de soportes del generador).
- *RBearingTemp_A* (Temperatura de soportes del eje principal).
- *RGridVoltage* (Voltaje en la Red).
- *RNacellePosition* (Posición de la góndola).
- *RControlPanelTemperature* (Temperatura de panel de control de la turbina eólica).

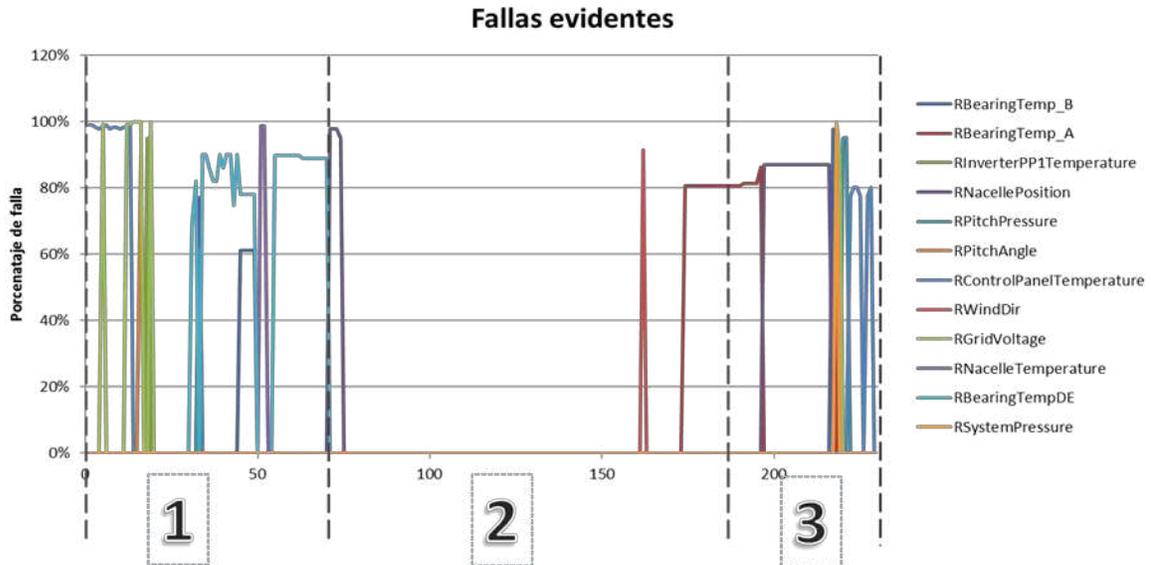


Figura 3: Patrón de comportamiento anormal del aerogenerador KW300

Si imaginamos que sobreponemos las dos figuras anteriores, se contrasta la generación de potencia de la turbina eólica y las variables afectadas por la inestabilidad en la generación. Esto muestra la capacidad del modelo para la detección de comportamiento anormal. Cuando se presenta algún evento inesperado, que puede ser falla en algún componente de la turbina, se genera un patrón de comportamiento anormal en algunas variables del modelo. Ese patrón es el que se pretende mapear a un catálogo de fallas para realizar un diagnóstico completo.

Para determinar el tipo de falla con la cual se está tratando, se deben someter a una etapa de discriminación, donde es preciso contar con un catálogo de fallas para mapear el tipo de anomalía mediante el reconocimiento de patrones usando técnicas de aprendizaje automático.

Por lo tanto, el problema a resolver es la Identificación de fallas del aerogenerador al detectar desviaciones al comportamiento normal. Las desviaciones se muestran en comportamiento anormal en un conjunto de variables, tal y como se

muestra en el primer y tercer momento de la Figura 2 y Figura 3, en donde se representa un patrón de comportamiento.

1.4 Objetivo general

El objetivo general del trabajo de investigación es desarrollar una técnica para la identificación de fallas al detectar desviaciones al comportamiento normal de aerogeneradores, usando datos obtenidos de simulación de seis diferentes fallas en el aerogenerador de la Máquina Eólica Mexicana (MEM) diseñada en el Instituto Nacional de Electricidad y Energías Limpias (INEEL). La detección del comportamiento se hace con modelos probabilistas y la identificación de fallas con reconocimiento de patrones.

1.5 Objetivos específicos

Al perseguir el objetivo general descrito anteriormente, se logran los siguientes objetivos específicos:

- Simular datos de operación normal y fallas de la MEM.
- Generar un procedimiento de reconocimiento de patrones.
- Generar un catálogo de fallas de la MEM.
- Evaluar 5 algoritmos de aprendizaje automático: *K-Nearest-Neighbor*, *Support Vector Machines*, Árboles de decisión, *Naive Bayes* y *Random Forest* (Fernández Delgado, y otros 2014) para la identificación de falla raíz de la MEM.
- Determinar y evaluar el rendimiento del algoritmo con mayor rendimiento para clasificar nuevas fallas raíz.

1.6 Metodología propuesta

La metodología propuesta para alcanzar los objetivos y llegar a los resultados es la siguiente:

- a) Generar el catálogo de fallas con la simulación de la MEM en la herramienta Focus6 y la ayuda de expertos.
- b) Simular las fallas conocidas insertando las fallas al comportamiento normal del aerogenerador.
- c) Seleccionar un conjunto de datos de entrenamiento y otro de validación para generar patrones de comportamiento anormal mediante el sistema DxAG.
- d) Extraer datos representativos a las fallas evidentes generadas para el procesamiento y mapear los resultados de la ejecución del módulo al catálogo de fallas.
- e) Realizar la extracción de características de los patrones de comportamiento para generar conjuntos de patrones de fallas.
- f) Experimentar con diferentes algoritmos de aprendizaje automático para lograr el reconocimiento de los patrones típicos en las fallas consideradas.
- g) Analizar los resultados emitidos por las pruebas efectuadas.
- h) Evaluar el rendimiento del mejor modelo de clasificación de falla raíz.

1.7 Organización de la tesis

Esta tesis está constituida por 6 capítulos. En el presente capítulo se describe la motivación de llevar a cabo esta investigación, así como el problema a resolver, el objetivo general, los objetivos particulares y la metodología a implementar. En el siguiente capítulo se describe el funcionamiento y componentes principales involucrados en las turbinas eólicas. El capítulo 3 describe el sistema de diagnóstico de aerogeneradores construido en el proyecto del CEMIE-Eólico el cual se denomina DxAG. El capítulo 4 describe la simulación de datos en operación normal y en falla generados con la Máquina Eólica Mexicana simulada en la herramienta computacional Focus6. El capítulo 5 describe los experimentos y resultados generados para el reconocimiento de falla raíz de la MEM con base en un procedimiento de reconocimiento de patrones. Finalmente, el capítulo 6 concluye este trabajo de investigación y propone recomendaciones para trabajo futuro.

Capítulo 2

Aerogeneradores

Un aerogenerador o turbina eólica es un dispositivo mecánico que captura la energía cinética del viento y lo convierte en energía eléctrica. El INEEL cuenta con el CERTE, el cual constituye un medio para incrementar el nivel de investigación y desarrollo tecnológico facilitando proyectos demostrativos o experimentales mediante una plataforma de demostración, validación y evaluación de aerogeneradores. En la infraestructura del CERTE se cuenta con una subestación eléctrica y con línea de interconexión para 5 MW de capacidad máxima para integrar aerogeneradores de diferentes capacidades. Cuenta con dos torres anemométricas de 80 y 40 metros de altura (INEEL 2016). Actualmente, el CERTE cuenta con la instalación de tres aerogeneradores. El primero de 300 KW de generación de potencia. El segundo de 2 MW de potencia y el tercero es un aerogenerador experimental de 3 KW (CEMIE-Eólico 2018). Se contempla la instalación de un cuarto aerogenerador, la MEM de 1.2 MW, que actualmente se encuentra en construcción por el INEEL.

En este capítulo se describe brevemente el funcionamiento y componentes de un aerogenerador. Para esta descripción se tomará como ejemplo la turbina eólica de la marca KOMAI de 300 KW (Komaihaltec Inc. 2015) instalada en el CERTE.

2.1 Funcionamiento del aerogenerador

La energía eólica es la energía que se obtiene del viento, es considerada como una fuente de energía renovable limpia que no contamina y ayuda a reemplazar la energía que se produce a través de los combustibles fósiles.

Las turbinas eólicas se emplean para transformar la energía cinética producida por el efecto de las corrientes de aire en energía eléctrica y se clasifican en dos tipos con base en la orientación de las palas. El primero de eje horizontal y el segundo de eje vertical.

- i). Los de eje vertical no cuentan con un mecanismo de orientación y el generador eléctrico puede ser instalado al ras de suelo. Suelen ser de bajo costo y facilita su mantenimiento e instalación. La principal desventaja es su poca producción energética.
- ii). Los de eje horizontal son los más usados en generación de baja mediana y alta potencia instaladas en grandes parques eólicos, dado que este tipo de aerogeneradores alcanzan mayores velocidades de rotación, manteniendo así una mayor eficiencia y producción energética. Además, este tipo de aerogeneradores adoptan el concepto de diseño danés, el cual consta básicamente de un rotor aerodinámico de tres palas o aspas de velocidad variable en conjunto con un sistema de orientación, generador asíncrono, cajas de engranes, torre y un sistema SCADA. El software SCADA es una herramienta para estructurar el sistema, que generalmente, se utiliza para controlar y monitorear procesos y datos medidos.

Para determinar el diseño y fabricación de los aerogeneradores se toman en cuenta diferentes factores como la velocidad, la turbulencia y las velocidades de ráfagas de viento basándose en la norma IEC 61400-1 de la Comisión Electrotécnica Internacional (IEC) (International Standard IEC 61400-1 2007). Así también, esta norma estipula 4 categorías de aerogeneradores que pueden instalarse en lugares con vientos altos, medios y bajos, basándose en 4 clases de viento diferentes. Siendo las turbinas de clase I las más robustas y estándar para operar en condiciones de vientos altos con un tiempo de vida útil de 20-25 años.

Para controlar y llevar a cabo el buen funcionamiento de los aerogeneradores se cuenta con un sistema de control, el cual es el cerebro del aerogenerador encargado de la seguridad y autonomía de éste, así como su disponibilidad, confiabilidad, eficiencia y durabilidad.

A continuación, se describen los componentes externos e internos del aerogenerador de la marca KOMAI de 300 kW de potencia nominal con un diámetro de rotor de 33 m, altura de 41.5 m, velocidad de inicio de 3 m/s y velocidad de paro de 25 m/s.

2.2 Componentes externos

La Figura 4 señala los componentes externos del aerogenerador, los cuales constan de góndola, mástil meteorológico, palas y torre.

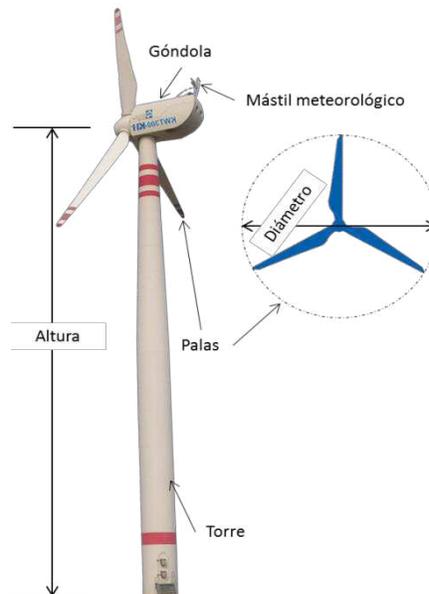


Figura 4: Componentes externos del aerogenerador

A continuación, se describe brevemente cada uno de los componentes:

Palas: Las palas están hechas de plástico reforzado con fibra de vidrio. En cada punta de las palas, se coloca un receptor para protección contra rayos. Las palas son una de las partes más importantes porque son los encargados de recoger la energía del viento. Esta energía es transmitida al buje, del buje a un sistema de transmisión mecánica y de ahí al generador para transformarla en energía eléctrica.

Torre: La torre tiene una estructura que consiste en tubos de acero de sección circular y de elevación cónica, en caso de rayos, funciona como un conductor para liberar la corriente eléctrica.

Mástil meteorológico: Está equipado con anemómetros y sensor de dirección de viento en una posición alejada del techo de la góndola.

- Anemómetros: Sensor para medir la velocidad del viento. El anemómetro de cazoletas tiene un eje vertical y tres cazoletas que capturan el viento, tal y como se observa en la Figura 5 de lado izquierdo
- Sensor de dirección de viento o *Wind Vane*: Es una veleta para medir la dirección de viento, tal y como se observa en la Figura 5 de lado derecho.



Figura 5: Sensor de velocidad y dirección de viento

Góndola: Es el chasis principal del aerogenerador, la cual se encuentra unida a la torre por medio de un sistema de orientación en la parte superior de la torre.

2.3 Componentes internos

La Figura 6 señala los elementos eléctricos y mecánicos necesarios para generar energía eléctrica. Algunos de los componentes internos del aerogenerador incluyen el rotor, generador eléctrico, caja multiplicadora, sistema de enfriamiento y sistema de orientación

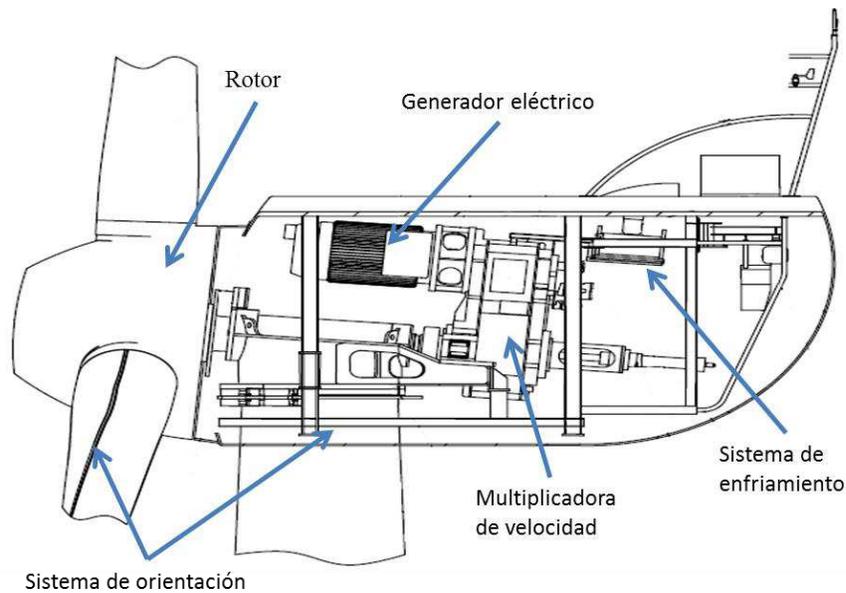


Figura 6: Componentes externos del aerogenerador

A continuación, se describe brevemente cada uno de los componentes:

Rotor: El rotor consta de tres palas, buje del rotor y eje del rotor. Está equipado con un sistema de *pitch* (inclinación) que ajusta el ángulo de la pala de acuerdo con la velocidad del viento y controla la rotación del rotor. El sistema de *pitch* consta de palas, soporte o rodamiento de palas, cubo y cilindro de *pitch*. La adquisición de energía es mayor cuando el ángulo de inclinación está configurado para tener el área de recepción de viento más grande. Funciona como un freno para

detener la rotación, cuando el ángulo está configurado para tener el área de recepción de viento más pequeña.

Caja de engranes o multiplicadora de velocidades: La caja de engranes aumenta la rotación del rotor con engranajes con el fin de mantener la velocidad de salida en cercano al ideal para el funcionamiento del generador.

Generador eléctrico: El Generador de inducción síncrono convierte la energía mecánica en energía eléctrica. El eje giratorio del generador está conectado con el eje lateral de alta velocidad de la caja de engranajes a través del acoplamiento de alta velocidad.

Sistema de orientación: Es un sistema para mantener al aerogenerador perpendicular al viento, se utiliza el sistema de *yaw* para tener la góndola en dirección del viento, y el sistema de *pitch* para cambiar el ángulo de inclinación de las palas para controlar la rotación o la salida de una turbina eólica.

Sistema de enfriamiento: Es un ventilador que está equipado para regular la temperatura de la caja de engranes y la temperatura dentro de la góndola.

Capítulo 3

Sistema de diagnóstico de aerogeneradores (DxAG)

El monitoreo y el diagnóstico temprano de fallas en aerogeneradores es esencial, ya que colabora en mejorar su seguridad, fiabilidad, rendimiento y la productividad de parques eólicos. Podemos definir el diagnóstico como el análisis de información que se realiza para determinar situaciones fuera de lo habitual que pueden estar ocurriendo en el funcionamiento de los aerogeneradores. Para esta tarea se requiere modelar el funcionamiento de la turbina según los estímulos que recibe del exterior. Por ejemplo y principalmente, según la velocidad y dirección del viento en la localidad. Entonces, el modelo deberá indicar los cambios que se registran en las partes de la turbina cuando se presenten cambios en el entorno. En otras palabras, algunas variables tendrán valores diferentes cuando otras variables cambien de valor.

En este capítulo se describe el funcionamiento del sistema DxAG para detectar desviaciones al comportamiento normal de aerogeneradores.

3.1 Descripción general del sistema DxAG

El sistema de diagnóstico de aerogeneradores fue desarrollado dentro del CEMIE-Eólico. La idea es crear modelos de comportamiento del AG para identificar desviaciones al comportamiento normal aprendido (Ibargüengoytia, García, y otros 2016). Los modelos de comportamiento se representan con una red Bayesiana donde se codifican las relaciones probabilistas entre las señales del SCADA. La detección de desviaciones al comportamiento normal se hace utilizando la teoría de validación de información (Ibargüengoytia, Vadera y Sucar 2006).

Una vez aprendidos los modelos de comportamiento de una turbina eólica (García, y otros 2016) con datos históricos del SCADA o de los adquiridos con la simulación de la MEM, se tiene el DxAG en condiciones de operación. Para su funcionamiento, se conecta al SCADA con el DxAG para adquirir la información actualizada de las variables, cada cierto tiempo de muestreo, por ejemplo, cada 5 minutos. El DxAG estima el valor de cada variable infiriéndola en el modelo probabilista de comportamiento y lo compara con la lectura del valor real de la variable en turno. Si el valor real coincide con el valor inferido en una distribución de probabilidad, entonces se considera comportamiento normal. En caso de que no correspondan los valores, se identifica un comportamiento anormal en las condiciones de operación correspondientes. Cuando se realiza este procedimiento de inferencia-comparación de todas las variables, se genera un patrón de comportamiento anormal como se muestra a la derecha de la Figura 7.

El patrón de comportamiento anormal resultante se puede ver como un video de una ventana de tiempo donde se muestran las variables (una con cada color) que presentaron comportamiento anormal. Sin embargo, esos patrones solo indican que hubo una situación anormal, pero no se puede identificar la falla raíz. El capítulo 5 explica en detalle el uso de la técnica de aprendizaje automático para el reconocimiento de patrones de fallas raíces.

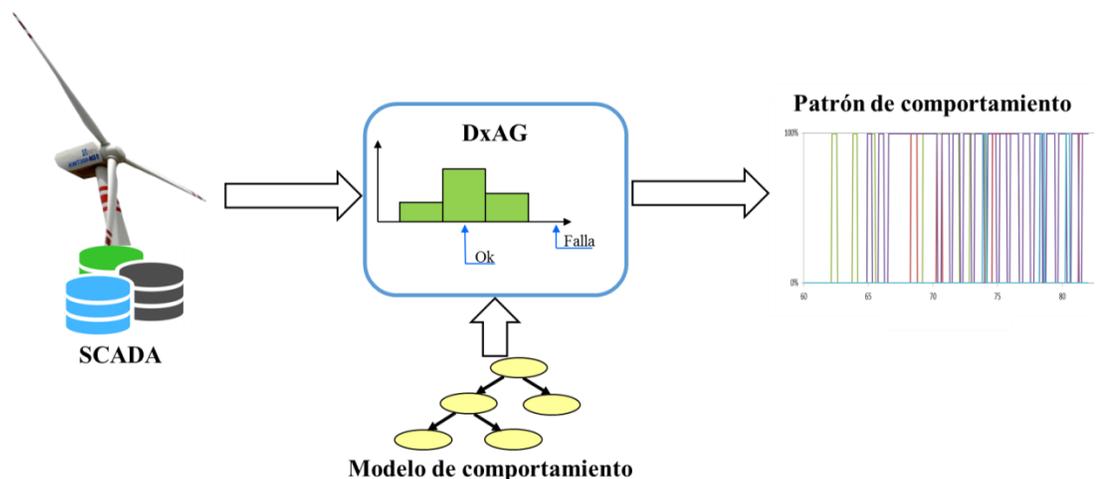


Figura 7: Configuración general del DxAG

La Figura 7 describe la configuración general del sistema de diagnóstico, la cual consta de un conjunto de datos de entrada y del modelo de comportamiento para calcular las desviaciones al comportamiento normal y así generar el patrón de comportamiento.

3.2 Redes bayesianas

Los modelos probabilistas son modelos que describen la relación probabilista que existe entre las diferentes variables del aerogenerador. Estos modelos se obtuvieron utilizando redes Bayesianas (RB).

Las RB son grafos acíclicos dirigidos que codifican relaciones probabilistas. Representan la dependencia y la independencia de variables aleatorias utilizadas en inteligencia artificial. En una RB como de describe en la Figura 8 de lado izquierdo, cada nodo representa una variable del dominio de aplicación y cada arco representa dependencia probabilista. La variable al final del arco (el hijo) es dependiente de la variable al origen del arco (el padre). Esta relación en algunos casos puede representar causalidad. Es decir, la variable al origen de un arco causa los cambios o efectos en la variable destino. Las RB se desarrollan a partir del teorema de Bayes (Bayes y Mr. Price 1763) que permite estimar la probabilidad de la hipótesis (H) dada cierta evidencia (E), basándose en la probabilidad de la evidencia dada la hipótesis (1), tal y como se observa en la Figura 8 de lado derecho.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (1)$$

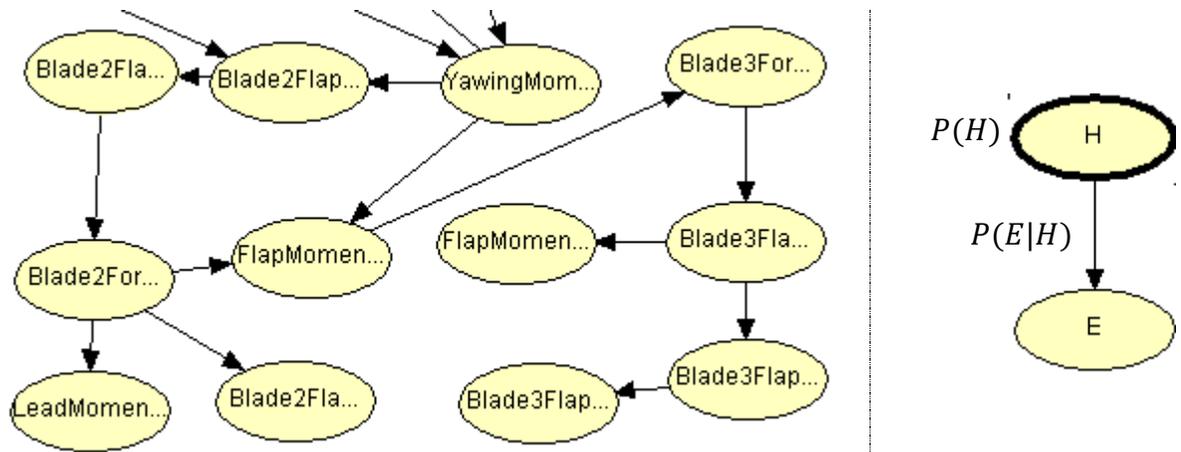


Figura 8: Representación de la Red Bayesiana

Como método de representación del conocimiento, se requiere que el experto o los datos, proporcionen dos tipos de conocimiento:

- i) Cualitativo: referente a la estructura de la red.
- ii) Cuantitativo: referente a los parámetros. Es decir, el vector de probabilidades a-priori de los nodos raíz $P(H)$, y las matrices de probabilidad condicional $P(E|H)$.

El formalismo de RB incluye algoritmos tanto para el cálculo de las probabilidades posteriores de algunas variables dada cierta evidencia. Es decir, si se cuenta con cierta evidencia (se adquiere el valor de un nodo), se calcula la probabilidad posterior de los demás nodos. Este formalismo incluye algoritmos para el aprendizaje automático de la estructura y los parámetros necesarios en una RB.

El aprendizaje de modelos probabilistas se puede hacer de alguna de las cuatro formas siguientes:

- i) Automáticamente con alguno de los algoritmos disponibles, tanto la estructura como los parámetros.
- ii) Estructura sugerida por los expertos en el dominio de aplicación y parámetros basándose en datos.

- iii) Estructura y parámetros sugerida por expertos.
- iv) Combinación entre métodos automáticos y manuales por experto.

Lo importante en todos ellos, es el contar con información histórica de valores de las variables relacionadas. En el anexo de esta tesis se muestra el conjunto de variables seleccionadas que participan en el proceso de aprendizaje.

Los datos se extraen de la simulación de la MEM o del sistema SCADA si se tratara de la turbina eólica KOMAI instalado en el CERTE. Se consideró que estos datos pueden ser suficientes para identificar el mejor algoritmo de aprendizaje para construir el modelo del comportamiento del AG. Con este conjunto de datos se aplica técnicas de aprendizaje automático. Existen varios algoritmos desarrollados para aprender. Algunos permiten la participación del experto humano. Para el caso de la generación de los modelos de comportamiento se utilizó el algoritmo de aprendizaje *Greedy-Search and Score* (Spirtes, Glymour and Sheines 2000), es un algoritmo que propone inicialmente una estructura de red y entra en un ciclo de modificaciones de la estructura para evaluar la que más representa a los datos. El algoritmo continúa buscando modificaciones a la estructura hasta que se logre una que ya no pueda ser mejorada con cambios. Además, incluye la restricción del número máximo de padres que puede tener cualquier nodo. Esto es importante para impedir la explosión combinatoria en el tamaño de memoria que necesitaría un modelo con alta interconexión entre nodos.

3.3 Metodología para crear modelos de comportamiento

Los modelos de comportamiento (García, y otros 2016) se utilizan para describir el comportamiento normal del aerogenerador para determinar si existen posibles fallas en su funcionamiento, para crear los modelos se basa en la historia de los datos de operación normal de los aerogeneradores.

Para la creación de modelos de comportamiento aplicados a dominios industriales, es decir, dominios donde es difícil crear modelos a través de expertos humanos, exclusivamente se realiza con base en la siguiente metodología:

La metodología se puede resumir en los siguientes 7 pasos:

- a) Adquisición de datos e identificación de variables: Generalmente, los dominios industriales cuentan con una gran cantidad de variables medidas en los sistemas de control. Se requiere identificar las variables que participan en el proceso a modelarse. Para el caso de estudio propuesto en esta tesis, se seleccionaron 111 variables simuladas de la MEM. El proceso de adquisición de información para crear conjuntos de entrenamiento para el modelo se describe en el capítulo 4. La relación y descripción de las variables se encuentran en el anexo de esta tesis.
- b) Limpieza de datos: La información recibida en los archivos históricos son mediciones realizadas por sensores físicos y luego transmitidos cientos de metros hacia el SCADA y al tablero de control. En todo este proceso, parte de la información puede corromperse por varias razones. A veces el instrumento deja de funcionar, el proceso se detiene por alguna razón o existe ruido eléctrico en los instrumentos. Esto produce datos erróneos. Sin embargo, existen técnicas para detectar inconsistencias y limpiar los archivos de lecturas erróneas. Para el caso de estudio propuesto en esta tesis por tratarse de datos simulados, la intervención de limpieza de los datos fue mínima. Esto se encuentra descrito en el “Procedimiento de extracción datos de fallas y no falla” del capitulo 4.
- c) Definición de variables calculadas que puedan proponer información: Una vez visualizada la información de todas las variables y después de limpiar de información corrupta, los expertos pueden sugerir analizar otras variables calculadas con las variables medidas.

- d) Normalización y discretización de datos: Con el conjunto de datos limpios y la generación de las variables calculadas, se procede a la normalización y a la discretización. La normalización se refiere a la identificación de los valores mínimo y máximo de cada variable para dividir a todos los datos entre el valor máximo. Esto produce que toda la información estará entre 0 y 1. Con esto se puede comparar el comportamiento que tienen las variables entre sí. La discretización se refiere a la división del rango de valores de cada variable en un número definido de intervalos. Esto permite la utilización de modelos probabilistas de variables discretas.

- e) Selección del algoritmo de aprendizaje apropiado: Con una matriz de datos limpios, normalizados y discretizados, se procede a la aplicación de algoritmos de aprendizaje automáticos. Existen varios tipos de algoritmos dependiendo de las necesidades de expresividad de las relaciones probabilísticas y dependiendo de la participación que se permite a un experto humano en la construcción del modelo.

- f) Consulta con expertos humanos para guiar el proceso de aprendizaje: Cuando un algoritmo permite participación humana, se debe consultar a los expertos.

- g) Aprendizaje de modelos iniciales: Con los datos y los algoritmos listos, se procede al aprendizaje de la estructura de la red. La estructura resultante indicará si algunas variables no tienen ninguna relación probabilista con las demás. También se evaluarán si algunas relaciones representadas en la red tienen sentido o pueden eliminarse a criterio de los expertos en la aplicación.

- h) Selección de variables: Se pueden identificar variables que no tienen relación probabilística con las demás. En este paso se eliminan para lograr un archivo histórico de menos variables que nos permiten analizar con más detalle a las que sí importan.

Finalmente, una vez contruidos los modelos de comportamiento, el DxAG se encarga de generar los patrones de comportamiento de las fallas simuladas. En el siguiente capítulo se describe el uso del simulador Focus6 para generar los datos simulados de operación normal y con falla de la MEM.

Capítulo 4

Experimentos con simulador

Simular fallas al insertarlas al funcionamiento normal de la Máquina Eólica Mexicana consiste en generar conjuntos de variables con comportamientos anormales. En este capítulo se describe la configuración del simulador para generar conjunto de datos de comportamiento normal y con fallas del aerogenerador, así como la descripción de los parámetros de configuración para generar un modelo de comportamiento normal. También se describe un catálogo de fallas para facilitar la generación e identificación de patrones de fallas y con base en éstos, orientar en la selección de técnicas de reconociendo de patrones.

4.1 Descripción de la Máquina Eólica Mexicana (MEM)

La Máquina Eólica Mexicana se desarrolla actualmente en el INEEL con el propósito de aprovechar las oportunidades que ofrece el mercado eólico internacional. El concepto de diseño utilizado en la Máquina Eólica Mexicana es el concepto danés: un rotor aerodinámico de eje horizontal de tres aspas, velocidad variable, con control de potencia por cambio de ángulo de paso (*pitch*), independiente en cada aspa y con respaldo de baterías, provisto de una unidad de control y supervisión (SCADA). Un sistema de orientación activo basado en motorreductores, disco de fricción y mordazas hidráulicas activas y pasivas, un sistema de paro basado en un freno principal de tipo aerodinámico (posición bandera de las aspas) y un secundario mediante disco de freno y mordazas hidráulicas a la salida de la caja de engranes, provisto además de un sistema manual para inmovilizar el rotor durante el mantenimiento de éste, todo ello montado sobre un chasis unitario y una torre tubular de acero (González, y otros 2013). La MEM es un

proyecto con múltiples etapas dentro de las cuales se encuentra la etapa del diseño, especificaciones de componentes, construcción e instalación de un prototipo en las instalaciones del CERTE. De acuerdo con la etapa de diseño la MEM cuenta con las siguientes características principales de operación:

- Potencia nominal: 1.2 MW.
- Diámetro del rotor: 60 m.
- Altura de instalación: 60 m
- Velocidad nominal: 13 m/s (± 2 m/s)
- Velocidad de inicio: 4 m/s.
- Velocidad de paro: 25 m/s.

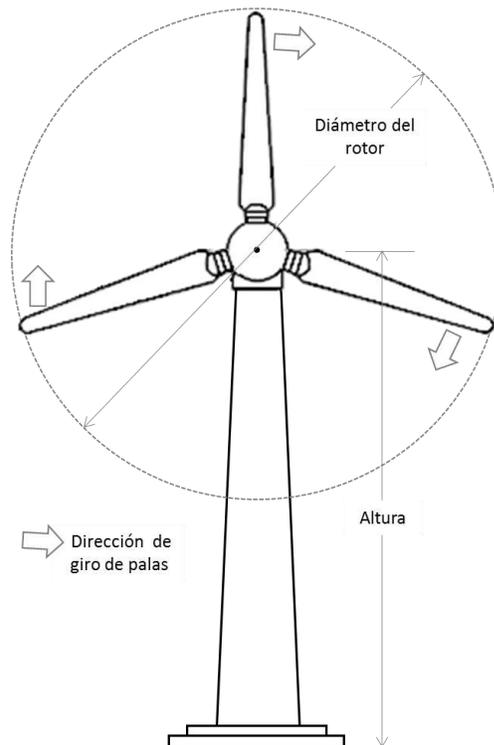


Figura 9: Diámetro, altura y giro del aerogenerador.

La Figura 9 describe el sentido de giro de las palas ocasionado por las corrientes de viento, la altura de instalación de aerogenerador y el diámetro del rotor

Las características que definen a un modelo de aerogenerador son dadas por su curva de potencia, en donde se exhibe la relación entre la velocidad del viento y la potencia. La curva de potencia de la MEM (ver Figura 10) indica la potencia eléctrica disponible en el aerogenerador a diferentes velocidades del viento. Los aerogeneradores de velocidad variable cuentan con tres regiones principales de operación (Johnson, y otros 2006):

- 1) Región 1: Consiste en el arranque normal del aerogenerador cambiando el *pitch* a posición de trabajo en relación a la velocidad de inicio del aerogenerador.
- 2) Región 2: Consiste en la transición entre la curva de potencia de la región 2 y la curva de potencia constante de la región 3, donde el objetivo es maximizar la cantidad de energía posible de viento dentro del rango de velocidades de 4-13 m/s.
- 3) Región 3: Es la región donde se encuentran velocidades de viento altas para generar potencia nominal, el objetivo es la regulación de velocidad del generador, por medidas de seguridad se limita la energía eólica capturada a una velocidad de 25 m/s.

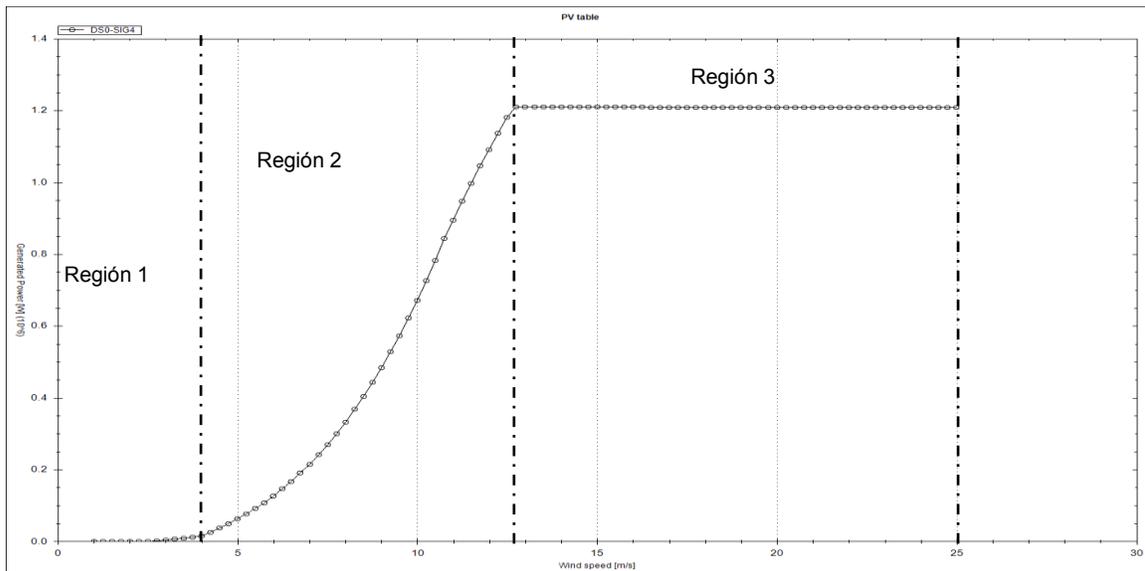


Figura 10: Curva de potencia de potencia diseñada para la MEM

La Figura 10 describe la curva de potencia de la MEM. En el eje x, se muestra la velocidad de viento en metros por segundo (m/s) y el eje y, la potencia generada en Mega Watts (MW). La velocidad de inicio y nominal de la MEM se describe en la segunda región donde el aerogenerador alcanza una producción de potencia nominal de 1.2 MW y la de velocidad de paro se encuentra en la tercera región donde con vientos muy fuertes (≥ 25 m/s) las palas se colocan en posición de bandera para que el aerogenerador frene y así evitar tensiones que dañen a la máquina.

4.2 Descripción del simulador Focus6

Un simulador puede crear modelos abstractos de aerogeneradores el cual permite reproducir el comportamiento de turbinas eólicas en ciertas condiciones de operación ayudando a minimizar riesgos relacionados con la exposición de expertos a situaciones peligrosas y al aerogenerador mismo, además de que en un aerogenerador real difícilmente se puede insertar fallas al funcionamiento de éste.

La idea de usar simulaciones radica en la facilidad de operación y manipulación en la generación de fallas, ayuda a reducir el tiempo y el costo asociados con el entrenamiento de modelos sobre el comportamiento normal de aerogeneradores, además proporciona herramientas para contar con la información de los componentes de los aerogeneradores y su comportamiento en diferentes condiciones de operación en tiempo real.

Focus6 (WMC 2010) es una herramienta computacional de simulación modular integrada para diseñar aerogeneradores y componentes de los mismos como palas de rotor, pre-diseño de rotor y soporte estructurales. Esta herramienta de simulación proporciona una interfaz de usuario consistente que integra una serie de herramientas para simular el funcionamiento normal o con falla del aerogenerador mediante múltiples herramientas de visualización y análisis de resultados.

Actualmente, la MEM cuenta con un modelo de diseño con cálculos Aeroelásticos que simula el funcionamiento del aerogenerador.

La norma IEC 61400-1 considera la utilización de un Modelo Dinámico Estructural (MDE) (WMC 2010) para predecir las cargas de diseño. Este modelo se utiliza para determinar cargas en un rango de velocidades considerando diferentes condiciones de viento según la norma. Los Casos de Carga de Diseño (DLC, por sus siglas en inglés) son combinaciones de condiciones para las diferentes situaciones de diseño del aerogenerador como lo son la producción eléctrica con y sin falla, paro normal, arranque, entre otras. Las cargas de diseño se generan considerando las condiciones de turbulencia de viento. Para esto se utiliza un modelo normal de turbulencia (NTM por su siglas en ingles) (International Standard IEC 61400-1 2007).

Para generar el modelo dinámico estructural del aerogenerador se usa el Programa de Simulación y Análisis de Turbinas Eólicas de Eje Horizontal (PHATAS, por sus siglas en ingles) (Lindenburg 2003). Este programa computacional permite determinar las curvas características del aerogenerador, así como el comportamiento dinámico, aerodinámico, estructural y las cargas (fuerzas y momentos) correspondientes a un aerogenerador de eje horizontal (Lindenburg 2003).

EL MDE es utilizado para determinar las cargas en un rango de velocidades de viento, considerando las condiciones de turbulencia y otras condiciones particulares del viento y de diseño para analizar la resistencia del aerogenerador.

4.3 Catálogo de fallas de la MEM:

En esta sección se describe el catálogo de fallas y los componentes del aerogenerador involucrados en un funcionamiento correcto y con falla de éste. En las fallas seleccionadas para la clasificación de patrones de fallas se involucran diferentes componentes, como lo es sistema de control de *yaw*, el sistema de control

de *pitch*, cortocircuito en el generador, sensor de dirección y velocidad de viento. A continuación, se describe el funcionamiento de cada uno de estos:

4.3.1 Sistema de control de *Yaw*

Es un sistema de control de orientación localizado entre la góndola y la torre del aerogenerador, para que la góndola siga la dirección del viento. Durante la operación normal, el sistema controla la góndola para que esté en la posición óptima en dirección perpendicular al viento para aumentar la capacidad de salida. El *yaw* (ver Figura 11) se controla siguiendo la góndola a la dirección del viento con el motor de *yaw* para mantener el ángulo de la dirección del viento a 0° .

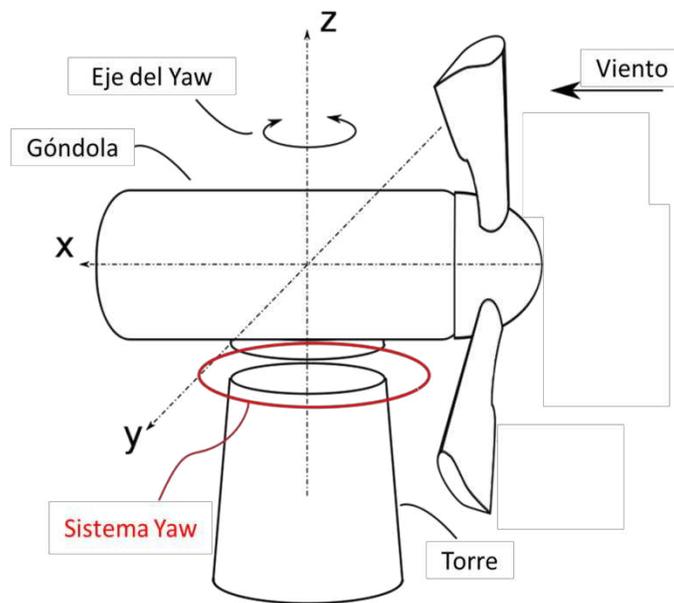


Figura 11: Estructura del funcionamiento de un aerogenerador (Wind Turbine, 2019).

4.3.2 Sistema de control de *pitch*

Es un Sistema de control o regulación de inclinación para cambiar el ángulo de inclinación de las palas para controlar la rotación o la salida de una turbina eólica. El objetivo del sistema de control del *pitch* es el ajustar el ángulo en función de la velocidad del viento para cambiar la fuerza aerodinámica en las palas y controlar de esta manera la potencia eléctrica de salida. El sistema de posicionamiento de las aspas está equipado por tres servomecanismos de actuación independientes y están coordinados con la unidad de rotación del *pitch* para ajustar el ángulo de inclinación de las tres palas al mismo tiempo. Cuando la velocidad de viento es demasiado alta, el controlador envía una señal al sistema de *pitch*, que inmediatamente hace girar las palas del rotor ligeramente fuera del viento. En caso contrario cuando la velocidad de viento disminuye las palas son giradas hacia el viento. Las palas del rotor deben ser capaces de girar alrededor de su eje longitudinal, tal como se muestra en la Figura 12.

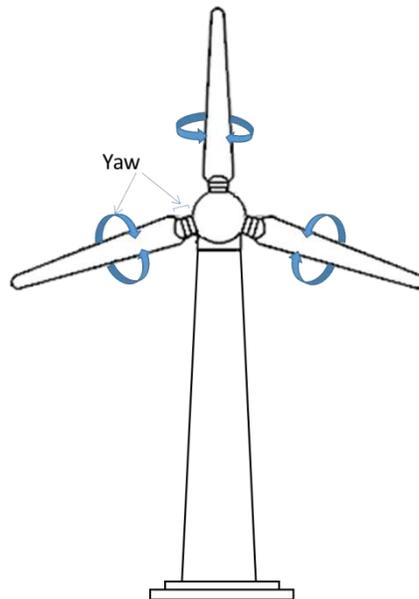


Figura 12: Sistema de control de *pitch*

4.3.3 Cortocircuito

Actualmente, se utilizan dos tipos de generadores eléctricos, los cuales se encargan de convertir la energía mecánica en energía eléctrica. El primero es el generador asíncrono del tipo rotor de devanado y el segundo el generador síncrono del tipo imanes permanentes. La MEM cuenta con un generador eléctrico síncrono con capacidad de 1500 KW con velocidad de 1800 rpm y una tensión de generación de 690 volts. El generador es controlado mediante un sistema de control diseñado con el objetivo de mantener el voltaje de salida en su valor nominal.

Los cortocircuitos son situaciones anómalas en los generadores eléctricos síncronos, donde sus componentes son sometidos a un estrés eléctrico, térmico y mecánico. Un cortocircuito en el generador ocasiona un estado de paro de generación de potencia, también puede ser provocado al detectar un incendio o cortocircuito en los cables principales del sistema eléctrico por medio de alta temperatura (arriba de 290°C) (González, y otros 2013)

4.3.4 Sensor de dirección y velocidad de viento

Dado a que el viento es capaz de transferir energía, existe la necesidad de medirlo por medio de sensores. Los sensores de dirección miden la dirección del viento en un rango de 0° a 360° con precisiones de medición elevadas y se utilizan en conjunto con sensores de velocidad que miden la velocidad del viento. Estos sensores son parte fundamental para el correcto funcionamiento del sistema de *yaw* y *pitch* en los aerogeneradores.

4.3.5 Descripción del catálogo de fallas

El catálogo de fallas consiste en una clasificación de las distintas fallas que se pueden simular en la MEM, y una breve descripción de operación donde se establece las posibles causas de cada una de las fallas, con el objetivo de facilitar la clasificación de fallas en el momento de hacer reconocimientos de patrones. En la Tabla 1 se presenta 7 fallas insertadas al funcionamiento normal del aerogenerador. La F0 describe la operación normal de la maquina a la cual se etiqueta como falla para ser comparada con el resto de las que si son fallas.

No.	Falla	Id falla	Descripción de la falla
1	Falla 0	F0	Operación normal.
2	Falla 1	F1	Sin velocidad de <i>pitch</i> en todas las aspas (control suprimido).
3	Falla 2	F2	Exceso de velocidad de <i>pitch</i> en todas las aspas (falla de sensor de velocidad).
4	Falla 3	F3	Sin velocidad de <i>pitch</i> en aspa No. 2.
5	Falla 4	F4	<i>Pitch</i> fuera de control en aspa No. 2.
6	Falla 5	F5	Cortocircuito en el generador.
7	Falla 6	F6	Falla en el sistema <i>yaw</i> .

Tabla 1: Catálogo de fallas insertada al aerogenerador

La F0 es la simulación del comportamiento normal de la MEM con un modelo normal de turbulencia con variaciones de viento.

La F1 se utiliza para modelar fallas del sistema de control de *pitch* de todas las palas. La velocidad de lanzamiento de todas las palas se establece en 0. Además, el controlador recibe un valor abajo del 13% para la velocidad del rotor. Esta falla se simula parcialmente en el SCADA, utilizando el valor correcto de la velocidad del rotor, pero alimentando al controlador *pitch* con un valor incorrecto.

La F2 se utiliza para crear una situación de exceso de velocidad. El controlador recibe 2/3 del valor del sensor de velocidad. Esto normalmente resultará en que el controlador intente alcanzar 1.5 veces la velocidad nominal. El SCADA dispara un paro una vez que la velocidad del rotor se excede activando los frenos de exceso de velocidad. Esta falla se simula en el SCADA, utilizando el valor correcto de la velocidad del rotor, pero alimenta al controlador pitch con un valor incorrecto.

La F3 se usa para modelar una sola falla del sistema de control de *pitch*, donde no se producen más alineaciones (giro o inclinación de la pala). La velocidad de *pitch* de la pala 2 se establece en 0.

La F4 se usa para modelar una sola falla del sistema de control de *pitch*, donde el controlador le indica a una pala que se incline hacia la posición de trabajo también conocido como inclinación de embalamiento¹. La F4 se usa para simular una falla del controlador permitiendo que el SCADA detecte y anule la velocidad de *pitch*, una vez que se detecta el problema.

F5: Se utiliza para modelar un cortocircuito al generador.

F6: La turbina comienza a orientarse o girar incondicionalmente. La dirección es la misma que el valor inicial establecido en el SCADA. Si no se modela la orientación, la tasa de giro mantiene lo especificado en el SCADA.

4.4 Descripción del modelo de comportamiento de la MEM

En esta sección se describe el modelo de comportamiento (García, y otros 2016) usando datos simulados del funcionamiento correcto de la turbina, generado con un modelo dinámico estructural.

¹ Embalamiento: Incremento significativo de la velocidad de un generador al reducirse la potencia eléctrica como consecuencia de un defecto en el sistema, o de la desconexión del generador de la red (real academia de ingeniería).

El comportamiento de un equipo puede verse como las variaciones en algunos de sus parámetros cuando existen cambios en el ambiente del equipo. Entonces, se puede describir el comportamiento de un proceso industrial como las variaciones que tienen sus variables principales dadas las variables de entrada al proceso y el entorno. Para la modelación del comportamiento de una turbina eólica se requiere entonces modelar el proceder adecuado de la turbina de acuerdo al entorno suyo y a los estímulos que puede recibir. Entonces, el modelo deberá indicar los cambios que se registran en las partes de la turbina cuando se resienten cambios en el entorno. En otras palabras, algunas variables tendrán valores diferentes cuando otras variables cambien de valor.

Se considera un modelo de comportamiento con producción normal como un aerogenerador conectado a la red eléctrica y operando normalmente. Con el modelo de comportamiento de la MEM es posible la identificación de desviaciones al comportamiento normal, es decir realizar un diagnóstico en tiempo real.

4.4.1 Configuración del modelo de comportamiento con producción normal

Para construir un modelo de comportamiento normal de la MEM se necesita de un conjunto de datos simulados con diferentes características de operación, dichas simulaciones requieren como datos de entrada diferentes parámetros, tales como la velocidad y dirección de viento definidos en los DLC.

Para configurar el simulador se usaron diferentes casos de carga de diseño. Por ejemplo, la Tabla 2 muestra la configuración de la nomenclatura de un caso de carga.

U	V	W	X	Y
1	2	w	x	y

Tabla 2: Nomenclatura de un caso de carga

Donde:

U: Representa el primer dígito de un caso de carga y corresponde al diseño de producción normal, con base en la norma: IEC 61400. Se identifica con el número 1.

V: Representa el segundo dígito de un caso de carga y corresponde al diseño de operación normal con variación del viento, con base en la norma: IEC 61400. Se identifica con el número 2.

W: Representa el tercer dígito de un caso de carga y corresponde al primer dígito de la velocidad de viento a evaluar.

X: Representa el cuarto dígito de un caso de carga y corresponde al segundo dígito de la velocidad de viento a evaluar.

Y: Representa el quinto dígito de un caso de carga y corresponde al ángulo de Azimut² del aerogenerador, para este caso de estudio el ángulo se estableció en 0°.

DLC	Vel (m/s)	Ángulo de orientación (°) Φ_{yaw}	Condición de operación	Ángulo azimutal (°)	No. De variables	Modelo
12120	12	-8	Normal	0	96	NTM
12130	13					
12140	14					
12150	15					

Tabla 3: Características de operación de las simulaciones para modelos de comportamiento

La Tabla 3 describe las características de operación de las simulaciones generadas para construir los modelos de comportamiento normal. Se generaron simulaciones de 5 min con una frecuencia de 16 muestras por segundo de 96 variables (ver anexo). Cada simulación se configuró con velocidades promedios de

² Azimut: Angulo de posición de cada pala de 0°-180°

12, 13, 14 y 15 m/s como parámetro entrada. Cada velocidad promedio con 3 semillas diferentes (ver Figura 13), con estas configuraciones el simulador genera datos aleatorios de velocidad en un rango de 4-25 m/s.

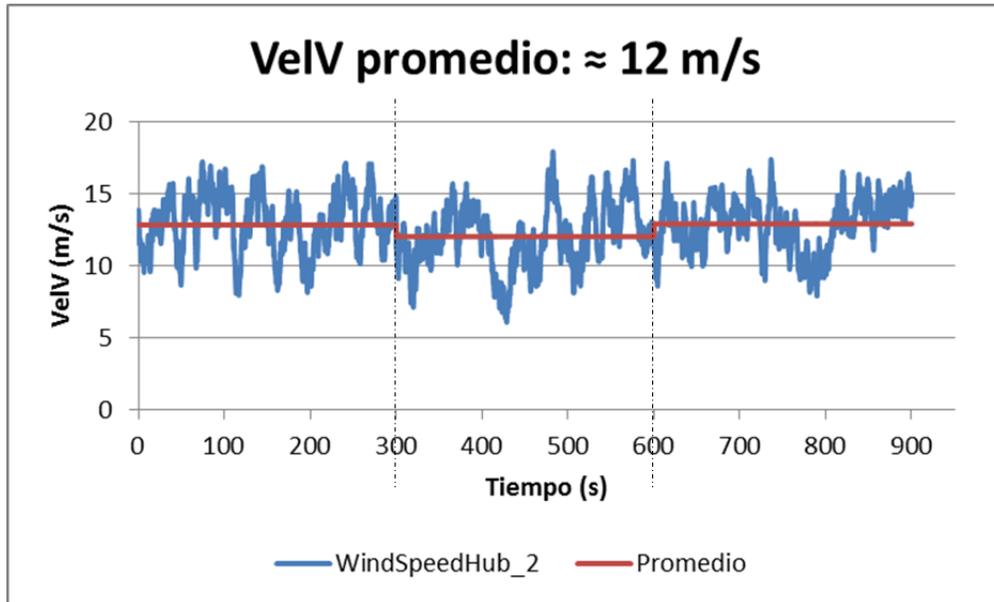


Figura 13: Velocidad promedio de 12 m/s con 3 semillas diferentes

En la Figura 13 se grafica en color azul la variable velocidad de viento del aerogenerador y en color rojo el promedio de las velocidad de ≈ 12 m/s. En el eje x, se observa a la escala de tiempo en segundos (s) y en el eje y, a la escala de la velocidad de viento en m/s. Se puede apreciar la variación de velocidades con sus respectivos promedios de 3 diferentes semillas, cada una separada por la línea punteada vertical.

4.4.2 Modelo de comportamiento con el sistema DxAG

Para la generación del modelo de comportamiento de la MEM en operación normal se utilizó el sistema DxAG. Para ello, se necesitó del conjunto de datos simulados de 96 variables (ver anexo) en operación normal del aerogenerador

adquiridos del simulador Focus6. Se consideró para el conjunto de prueba la simulación de las fallas con diferentes condiciones de operación para generar los patrones de variables en comportamiento anormal.

La Figura 14 representa el modelo de comportamiento normal construido con 96 variables involucradas en las diferentes configuraciones de operación de la turbina establecidos por el Modelo Dinámico Estructural de la MEM; en este conjunto de variables se incluyen las mediciones de esfuerzos, posición de palas, orientación de la góndola, dirección de viento, velocidades, potencia, entre otras .

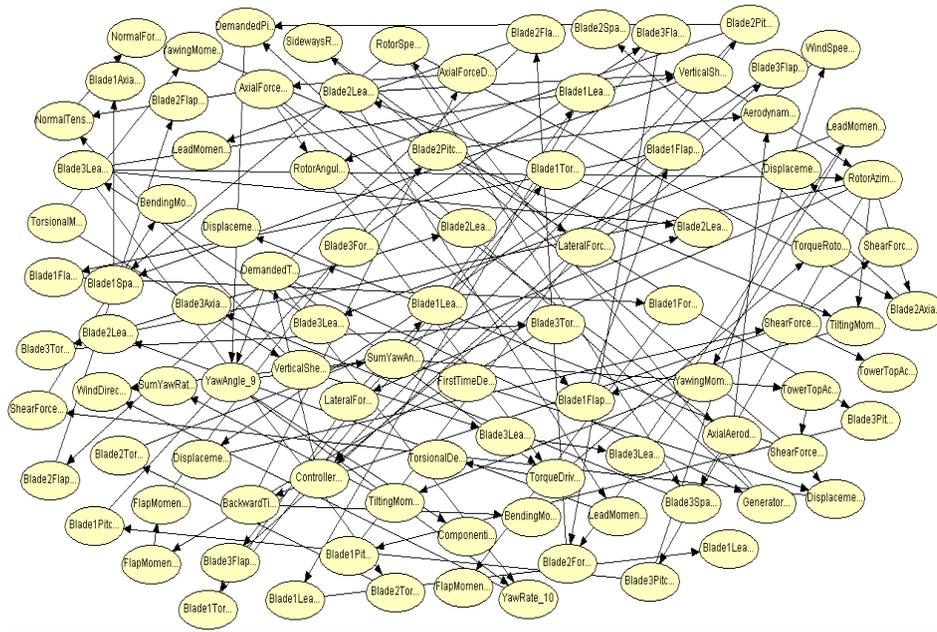


Figura 14: Modelo de comportamiento de la MEM en operación normal

4.5 Simulación de fallas en la MEM

Para generar simulaciones de 7 fallas incluyendo la no falla (ver Tabla 1) insertadas a la MEM se utilizó las características de operación de la Tabla 3. Para estas simulaciones se configuró una ventana de tiempo de 10 segundos donde se presenta la falla, el inicio de la falla es a partir del segundo 30, tal y como se puede observar en la Figura 15. Cada una de las fallas simuladas se configuraron con

velocidades promedio de 12, 13, 14 y 15 m/s como parámetros de entrada, cada velocidad promedio con 300 semillas diferentes (ver Figura 16). Con estas configuraciones el simulador genera datos aleatorios de velocidad en un rango de 4-25 m/s.

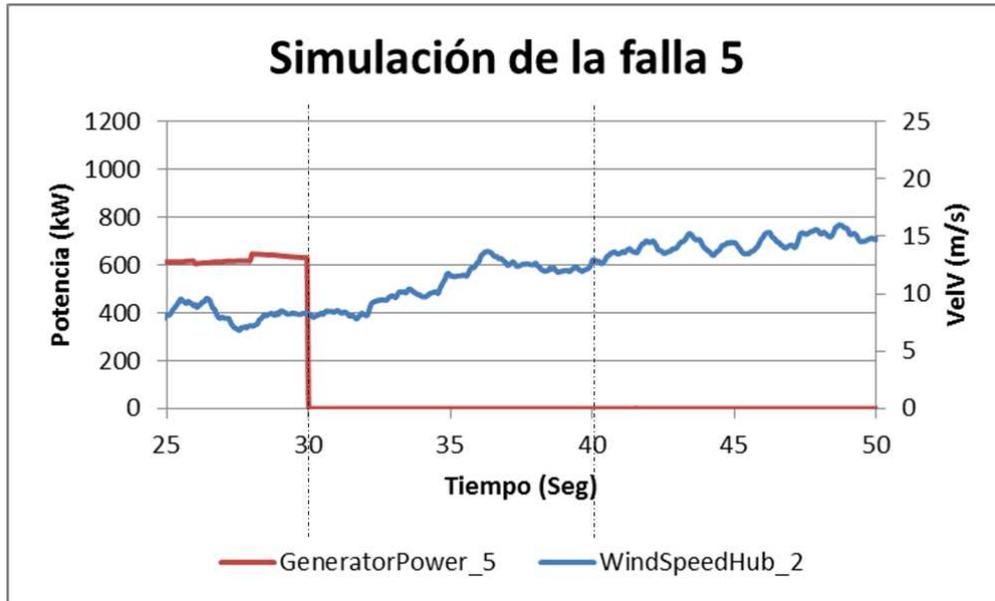


Figura 15: Simulación de cortocircuito en la MEM

En la Figura 15 se grafican dos variables del aerogenerador. La primera en color rojo, la generación de potencia y la segunda en color azul, la velocidad de viento. Se puede apreciar la falla del aerogenerador ocasionada por un cortocircuito en el segundo 30 cuando la generación se pone en cero aún con viento favorable. En el eje x, se observa a la escala de la velocidad en m/s, en el eje y, a la escala de potencia en KW. Las dos líneas punteadas verticales es la ventana de tiempo donde se presenta la falla.

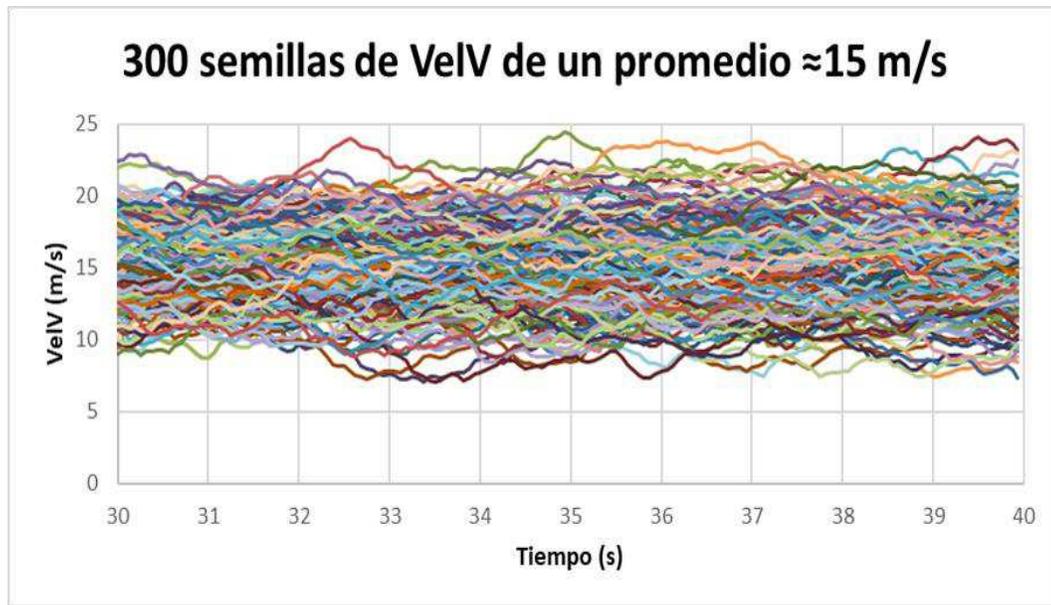


Figura 16: Velocidad de viento en diferentes variaciones

En la Figura 16 se grafican 300 variaciones de velocidades (semillas) con un promedio de ≈ 15 m/s. En el eje x, se observa a la escala de tiempo en segundos (s) y en el eje y, a la escala de la velocidad de viento en m/s.

4.6 Patrón de comportamiento

Las fallas ocurren de diferentes formas y por muchas razones, las cuales pueden ser al azar o provocadas. El término falla generalmente se define como una desviación de un rango aceptable de una variable observada o un parámetro calculado (Md Raihan y Syed A. 2013). Estas desviaciones llegan a presentarse como sucesos frecuentes a las cuales se les denomina patrones. En la Figura 17 se observa el patrón de falla de la variable *RWindDirectionHub_6* donde el eje x, es el tiempo en segundos (s) de muestreo de la variable y, el eje y es la probabilidad de comportamiento anormal.

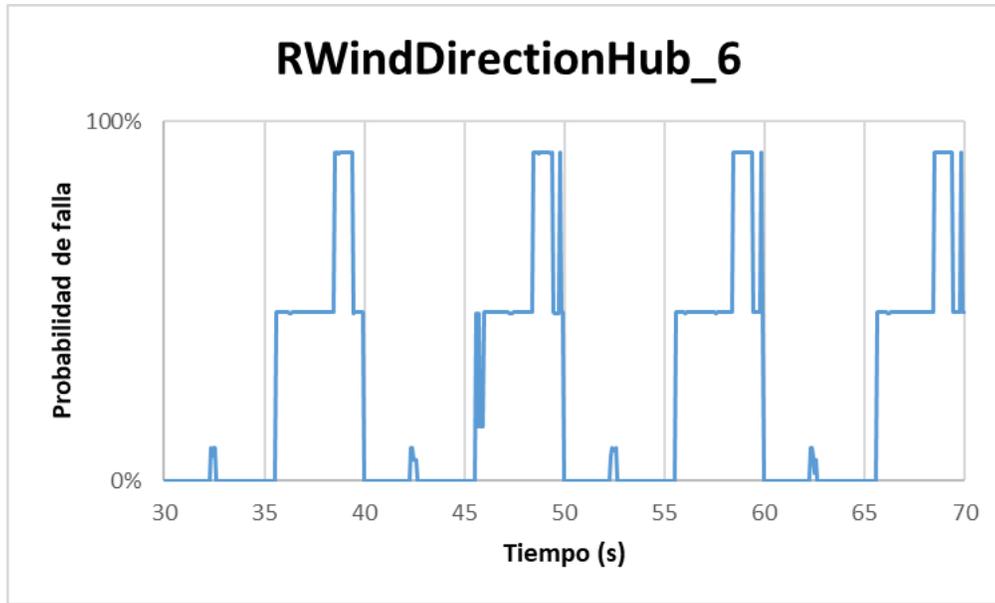


Figura 17: Serie de temporal de la variable *RWindDirectionHub_6*

Por lo tanto, un patrón de comportamiento describe el conjunto de variables con comportamiento anormal que se produce a partir de una falla raíz o un mal funcionamiento del AG.

En la Figura 18 se describe el patrón de comportamiento que define la F1. En el eje x, se observa un periodo de 10 segundos donde se presenta la frecuencia con la que ocurren las variables con comportamiento anormal. Por ejemplo, las variables *RBlade3FlapwiseBendingMoment_333* y *RyawingMomentDriveTrain_53*, y en el eje y, la probabilidad de comportamiento anormal detectada en un conjunto de variables.

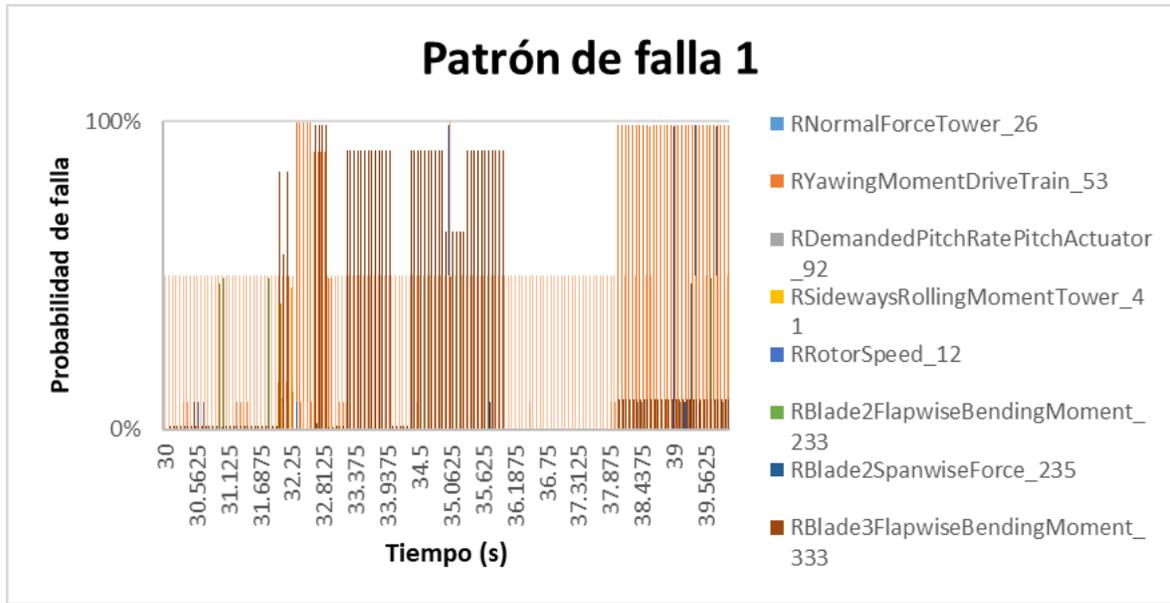


Figura 18: Patrón de comportamiento de la F1

A partir del catálogo de fallas se generaron un conjunto de patrones de comportamiento (ver Figura 19) que representan las fallas insertadas al aerogenerador, estos patrones se generan con el sistema DxAG.

Debido a que diferentes componentes de un mismo aerogenerador pueden registrar diferentes tipos de patrones de comportamiento, el problema se da en identificar correctamente la falla raíz. Para resolver este problema se utiliza la técnica de reconocimiento de patrones para la clasificación de la falla raíz dado un patrón mapeado a un catálogo de fallas definido.

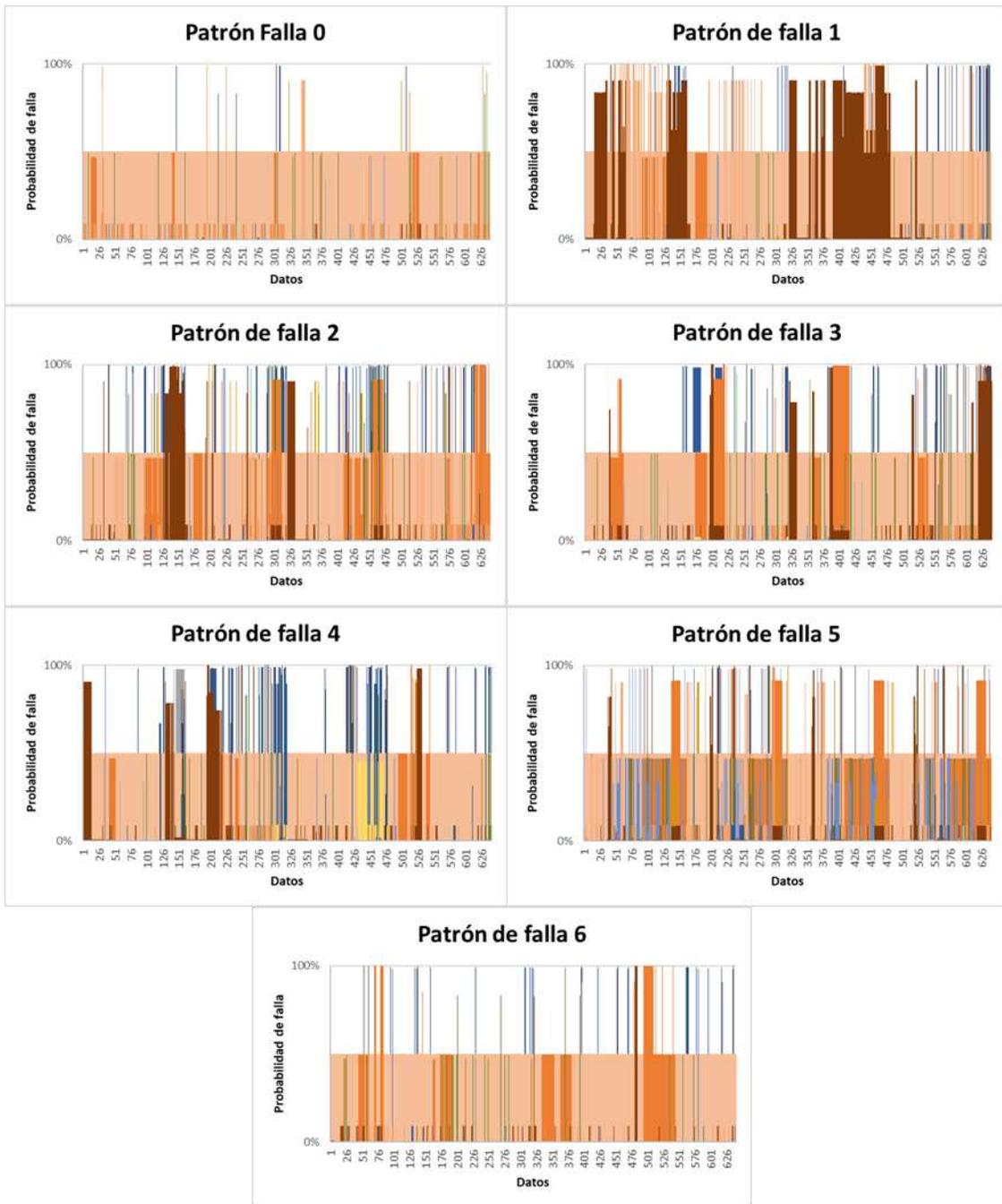


Figura 19: Patrones de comportamiento dado un catálogo de fallas

La Figura 19 representa un ejemplo de los patrones de comportamiento de las fallas descritas en el catálogo de fallas.

4.7 Procedimiento de extracción datos de fallas y no falla

En esta sección se describe el procedimiento para extraer datos de entrenamiento y prueba para los modelos usados en el DxAG.

Para extraer la información de las fallas y no falla del aerogenerador se generó el siguiente procedimiento de 6 pasos (ver Figura 20). A continuación, se describe la función de cada uno de los pasos en el procedimiento.

- i) Configuración MEM: consiste en configurar el simulador Focus6 con las características de operación de la MEM para cargar los DLC utilizados para generar simulaciones del comportamiento del aerogenerador.
- ii) Selección de operación: Consiste en seleccionar el tipo de operación de la MEM, se configuraron dos tipos, el primero es la operación de producción normal y la segunda es la operación de producción con falla. En esta sección se debe configurar el número de simulaciones y tiempo de simulación requerido y posterior a esas configuraciones iniciar la simulación.
- iii) Seleccionar variables y exportar datos: Posterior a la simulación se debe extraer la información de las 111 variables (ver anexo) descriptivas del funcionamiento del aerogenerador para su reprocesamiento, por lo cual se debe exportar los datos simulados a archivos de valores separados por coma (csv).
- iv) Pre-procesamiento: En esta sección se realizan los siguientes tres pasos básicos:
 - a. Selección de variables: Se reducen las variables de 111 a 95 eliminando todas aquellas que su valor promedio de operación sea igual a cero o valor nulo.
 - b. Formato de variable: El nombre de las variables deben tener un formato válido para ser usados en la herramienta de IA *Hugin Research* (Andersen, y otros 1989) para generar los modelos de comportamiento. El nombre de las variables debe ser modificado con la estructura que se muestra en la siguiente tabla.

Formato de variable original	Formato de variable modificada
(2) <i>Wind speed hub [m/s] @ 0 m</i>	<i>WindSpeedHub_2</i>
(5) <i>Generator power [W] @ 0 m</i>	<i>GeneratorPower_5</i>
(6) <i>Wind direction at hub [deg] @ 0 m</i>	<i>WindDirectionHub_6</i>
(9) <i>Yaw angle [deg] @ 0 m</i>	<i>YawAngle_9</i>
(12) <i>Rotor speed[rpm] @ 0 m</i>	<i>RotorSpeed_12</i>
(17) <i>Rotor angular acceleration [deg/s^2] @ 0 m</i>	<i>RotorAngularAcceleration_17</i>

Tabla 4: Formato de la variable: velocidad de viento

- c. Discretización: Se discretizan los valores continuos de las variables para la formación de los modelos de comportamiento usados en el DxAG. La discretización de las variables en cada conjunto de datos se realizó con base en la discretización equidistante (2) de acuerdo a los rangos de cada conjunto de datos. Con esto se garantiza que todas las variables que forman los modelos probabilistas de comportamiento estén formadas con número fijo de intervalos.

$$w = \frac{\max - \min}{i} \quad (2)$$

Por lo tanto, los intervalos discretizados son:

$$\min + w, \min + 2w, \dots, \min + (i - 1)w \quad (3)$$

donde:

w es la distancia o tamaño del intervalo

\min es el valor máximo del conjunto de datos.

\min es el valor mínimo del conjunto de datos

i es el número de intervalo equidistante

- v) Datos entrenamiento: Se genera un subconjunto de datos para entrenar los modelos de comportamiento normal con datos discretos.

- vi) Datos de prueba: Se genera un subconjunto de datos para probar el modelo de comportamiento normal, en este caso los datos de prueba serán los datos con fallas simuladas. Los conjuntos de datos de prueba deberán tener las mismas condiciones de operación que los datos de entrenamiento.

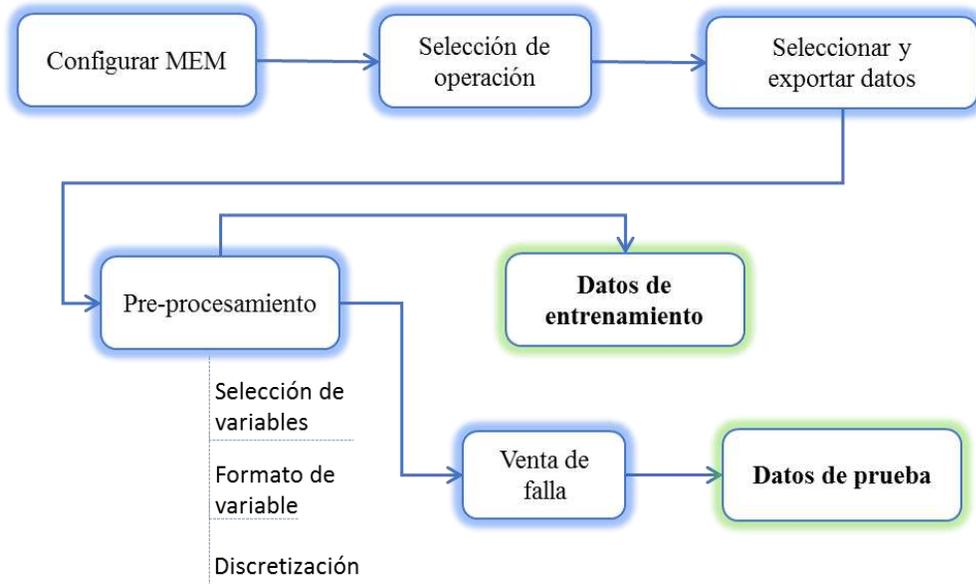


Figura 20: Procedimiento de extracción de datos

Se han planteado los experimentos para adquirir los datos simulados del MEM en operación normal y con falla. A continuación, se ejecutan los experimentos para el reconocimiento de la falla raíz.

Capítulo 5

Reconocimiento de falla raíz

Las causas implícitas en las anomalías de los aerogeneradores como *pitch* fuera de control o cortocircuito en el generador eléctrico se le conoce como falla raíz, pudiendo ser del tipo funcional donde el equipo deja de cumplir algunas funciones o del tipo catastrófico donde el equipo queda fuera de operación. En este capítulo se describe el procedimiento de reconocimiento de patrones para la identificación de falla raíz, así como el conjunto de experimentos realizados con algoritmos de aprendizaje automático comúnmente conocido como *Machine Learning*.

5.1 Reconocimiento de patrones

El Reconocimiento de patrones (RP) es la disciplina científica que trata los métodos para la descripción y clasificación de objetos (Marques de Sá 2001). El término reconocimiento de patrones tiene sus orígenes en la ingeniería y abarca una amplia gama de problemas de procesamiento de información de gran importancia práctica, desde el reconocimiento de voz y la clasificación de caracteres escritos a mano, hasta la detección de fallas en maquinaria y el diagnóstico médico. A menudo, estos son problemas que muchos humanos resuelven de una manera aparentemente sin esfuerzo. Sin embargo, su solución utilizando computadoras ha resuelto en muchos casos problemas extremadamente complicados (C. Bishop 1995).

Algunas de las técnicas clásicas para el RP más conocidas son las siguientes (Gómez Gil 2018):

- Métodos de estimación paramétricos
- Métodos de estimación no paramétricos

- Métodos estocásticos
- Métodos basados en aprendizaje supervisados y no supervisados

5.1.1 Procedimiento de reconocimiento de patrones

La búsqueda automática de patrones en los datos mediante el uso de algoritmos de aprendizaje es fundamental para poder tomar acciones tales como clasificar los patrones para la identificación de falla raíz. Para clasificar de manera correcta la falla raíz se establece un procedimiento de reconocimiento de patrones tal y como se muestra en la Figura 21 la cual consiste en los siguientes 6 pasos:

- i) Patrón de falla: Obtener el conjunto de variables con comportamiento anormal que se produce a partir de una falla del AG. El patrón de comportamiento se obtiene del sistema DxAG tal como se describe en capítulos anteriores.
- ii) Formación de objetos: Extraer las características partiendo de un patrón de comportamiento de falla.
- iii) Reducción de dimensionalidad: Obtener el mejor subconjunto de variables en un conjunto de datos mediante una búsqueda automática.
- iv) Modelo de clasificación: Aplicar el algoritmo de clasificación al conjunto de patrones de fallas definidos para la clasificación de la falla raíz.
- v) Evaluación del modelo: Verificar que los modelos de clasificación se estén desempeñando de manera correcta.
- vi) Clasificación de falla raíz: Establecer la precisión y exactitud del modelo para clasificar futuras fallas.

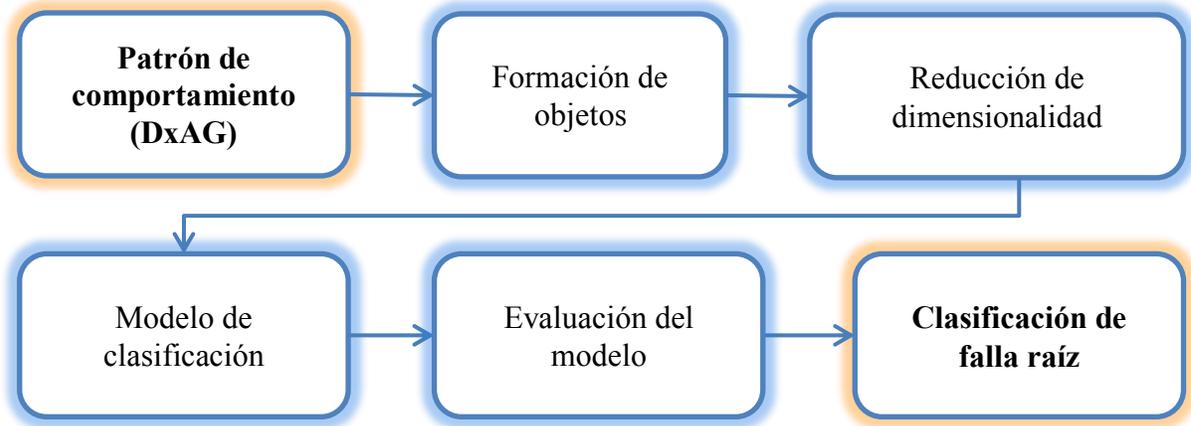


Figura 21: Procedimiento de reconocimiento de patrones

Considerando el problema de clasificación de falla raíz del aerogenerador de la Figura 18, podemos resolverlo representando cada patrón de comportamiento generado por el DxAG como una imagen de un tamaño $n \times m$, donde n sería los registros simulados y m representa el conjunto de variables con comportamiento anormal. Por lo tanto, puede representarse mediante un vector X de características descriptivas de $(n \times m) \times 1$ que sería el patrón de falla. El objetivo es usar un algoritmo de aprendizaje automático supervisado que reciba como entrada el vector de características y obtenga como resultado la identificación de la falla raíz de manera correcta.

Para el enfoque de aprendizaje automático supervisado, llamaremos conjunto de entrenamiento a los sub-conjuntos de N patrones de falla, donde las clases de las fallas son conocidas de antemano dado a que se analizaron cada una de las fallas de manera individual y se etiquetaron manualmente basándose en un catálogo de falla definido en el capítulo 4. Entonces para cada patrón de falla x (de la F0-F6) tenemos un vector objetivo f .

Por ejemplo, se expresa como una función $y(x)$ al resultado de ejecutar un algoritmo de Machine Learning (ML) el cual toma nuevos patrones de falla x como entrada y genera un vector de salida con la misma estructura que el vector objetivo.

La forma precisa de la función $y(x)$ se determina durante la fase de entrenamiento o aprendizaje. Una vez que el modelo está entrenado, puede predecir la identidad de los nuevos patrones de falla. A la capacidad de clasificar correctamente nuevos ejemplos que no fueron utilizados para el entrenamiento, en ML se le conoce como generalización.

5.2 Aprendizaje Automático

Se define el aprendizaje de maquina como el campo de estudio que brinda a las computadoras la capacidad de aprender sin ser programadas explícitamente (Arthur L. 1959).

Se dice que un programa de computadora aprende de la experiencia E con respecto a una de tareas T y alguna medida de rendimiento P , si es que su rendimiento en T , medido por P , mejora con la experiencia E (Tom M. 1998).

El ML es una rama de la inteligencia artificial, basada en la idea de que las maquinas aprenden de forma autónoma a partir de un conjunto de datos, identificar patrones y tomar decisiones con poca intervención humana. La base del aprendizaje son los datos y cuantos más datos, mejores y más precisos serán los resultados. Existen muchas razones por la cual se justifica el usar ML. Por ejemplo, nos permite crear herramientas para reducir tiempos de programación, permite personalizar tareas para fines específicos y además permite resolver problemas que sería imposible resolver a mano.

En la Figura 22 se describe los principales tipos de aprendizajes y los problemas de aprendizaje, los cuales nos ayudan a resolver distintos problemas con cierta facilidad dependiendo la complejidad y el tipo de problema a resolver.

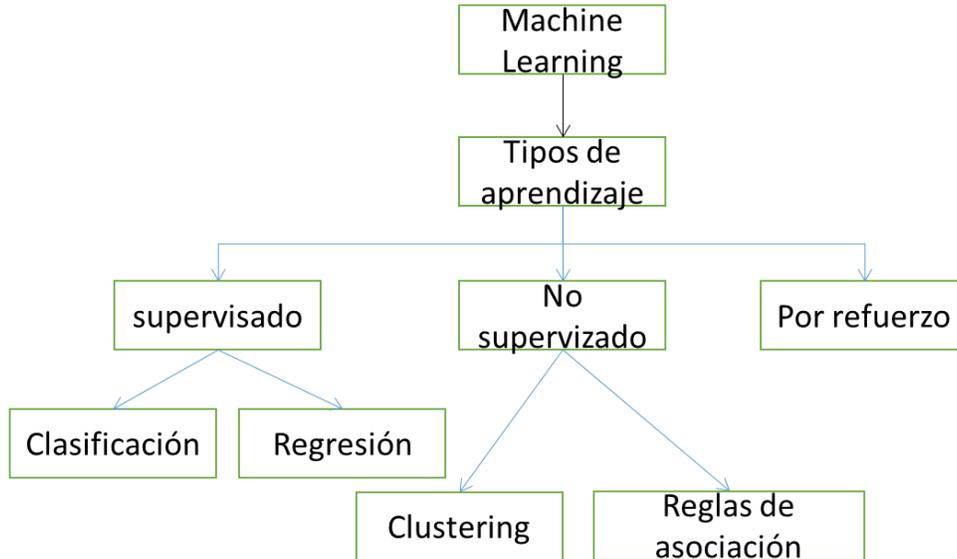


Figura 22: Tipos de aprendizaje

5.3 Principales tipos de aprendizaje automático

5.3.1 Aprendizaje supervisado

El aprendizaje supervisado es el primer tipo de aprendizaje y el más usado que involucra un conjunto de técnicas donde la principal característica es tener datos etiquetados los cuales permiten realizar clasificaciones o predicciones basadas en comportamientos o características analizadas en datos históricos. Este proceso es considerado aprendizaje supervisado porque el algoritmo aprende a partir de un conjunto de datos de entrenamiento de manera supervisada. Se conoce la etiqueta de salida y el algoritmo itera sobre los datos de entrenamiento hasta obtener un rendimiento óptimo.

La Figura 23 describe el procedimiento de aprendizaje supervisado

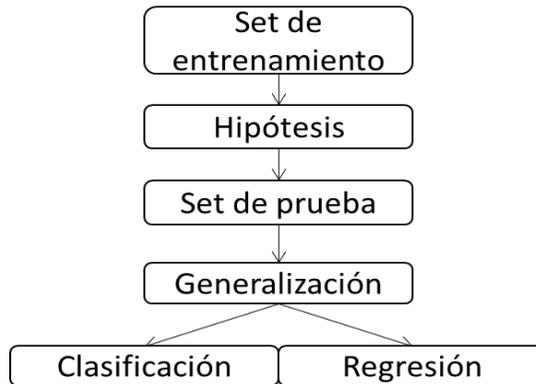


Figura 23: Procedimiento de aprendizaje supervisado

El procedimiento del aprendizaje supervisado se basa en un conjunto de datos de entrenamiento de N ejemplos de pares entrada-salida y aprende una función que se asigna de entrada a salida. Un par es: $(x, f(x))$, donde x es la variable de entrada, y $f(x)$ es la salida de la función f aplicada a x .

Por ejemplo: $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$, donde cada y_j fue generado por una función desconocida $y = f(x)$

Por lo tanto, la tarea de aprender una función es: Dada una colección de ejemplos de f , devolver una función h que se aproxima a la verdadera función f , donde la función h se denomina hipótesis. La razón por la cual el aprendizaje es difícil, desde un punto de vista conceptual, es que no es fácil determinar si una función h es una buena aproximación de f .

El aprendizaje es una búsqueda en el espacio de posibles hipótesis para alguna que se desempeñara bien, incluso en nuevos ejemplos fuera del conjunto de entrenamiento. Para medir la exactitud de una hipótesis, le damos un conjunto de datos de prueba que son distintos del conjunto de entrenamiento. Decimos que una hipótesis generaliza bien si predice correctamente el valor de y para ejemplos novedosos. A veces, la función f es estocástica; no es estrictamente una función de

x , y lo que tenemos que aprender es una distribución de probabilidad condicional (P) dado por: $P(Y|x)$. (Russell y Norving 2010)

El aprendizaje supervisado se agrupa en 2 problemas:

- i) Clasificación: Este problema de aprendizaje se le denomina clasificación cuando la salida de la variable y es uno de un conjunto finito de valores y puede ser binaria si solo hay dos clases o multiclase si existen más de 2 clases.

El Figura 24 se muestra un ejemplo de clasificación binaria de lado derecho, y multiclase de lado izquierdo, donde se encuentran diferentes patrones (objetos de color rojo, azul y verde) y se tiene por objetivo clasificar los objetos de cada color a la clase que pertenezcan relacionando características etiquetadas.

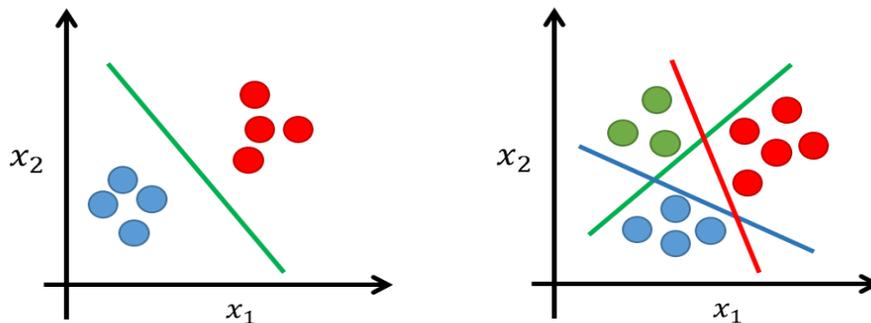


Figura 24: Problema de clasificación binaria y multiclase

- ii) Regresión: Cuando y es un número (como la temperatura del siguiente día), el problema de aprendizaje se llama regresión. Técnicamente, resolver un problema de regresión es encontrar una expectativa condicional o un valor promedio de y , porque la probabilidad de que hayamos encontrado exactamente el número correcto de valores reales para y es cercano a 0. (Russell y Norving 2010). La característica principal de un problema de regresión es que la variable de salida debe ser un valor real ($y \in \mathbb{R}$).

En la Figura 25 se describe un ejemplo de regresión lineal con una variable donde en el eje x , representa la variable de entrada denotada por x y en el eje y , representa la variable de salida denotada por y , siendo la recta de color rojo la hipótesis $h(x)$.

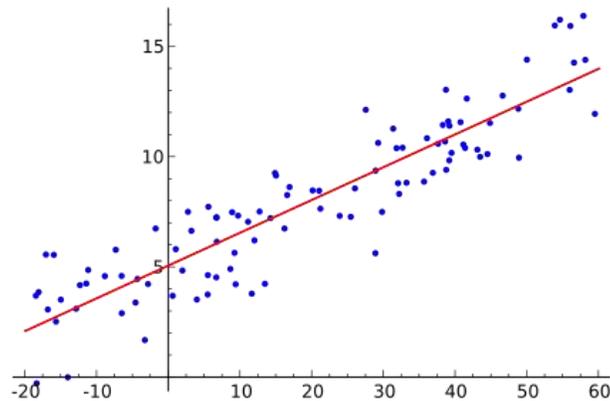


Figura 25: Problema de regresión lineal

Algunos ejemplos de algoritmos de aprendizaje supervisado son:

- Regresión
- Redes Neuronales
- KNN(K-vecinos cercanos)
- Bayesianos
- Árboles de decisión
- Ensamble
- SVM

5.3.2 Aprendizaje no supervisado

El problema de aprendizaje no supervisado consiste en aprender a partir de patrones de entradas en los que no se especifican los valores de sus salidas (no hay etiquetas de salida). Estos algoritmos, pueden por sí mismos descubrir una estructura que represente los datos.

El aprendizaje no supervisado se agrupa en 2 problemas:

- i) *Clustering*: La tarea de aprendizaje no supervisada más común es la de agrupación, es cuando desea descubrir las agrupaciones inherentes en los datos, Por ejemplo, un conductor de taxis podría desarrollar gradualmente un concepto de "días de buen tráfico" y "días de mal tráfico" sin haber recibido nunca ejemplos etiquetados por un maestro (Russell y Norving 2010).
- ii) Reglas de Asociación: Estas reglas de asociación son reglas *if – then* similares a las reglas de clasificación, excepto que la parte *then* no está restringida a una variable de clase en particular y puede contener cualquier característica (o incluso varias características) (Peter 2012).

Las reglas de asociación se utilizan para descubrir patrones que ocurren en común dentro de un determinado conjunto de datos. Este tipo de regla es popular en las aplicaciones de marketing. Un ejemplo de reglas de asociación son los algoritmos de detección de anomalías.

5.3.3 Aprendizaje por refuerzo

Es una forma de aprender políticas para decidir qué acción tomar dado un evento. Esta técnica se ocupa del problema de encontrar acciones adecuadas para tomar en una situación determinada con el fin de maximizar una recompensa. Aquí, el algoritmo de aprendizaje no recibe ejemplos de resultados óptimos, como el de aprendizaje supervisado, sino que debe descubrirlos mediante un proceso de prueba y error. Normalmente, hay una secuencia de estados y acciones en las que el algoritmo de aprendizaje interactúa con su entorno. En muchos casos, la acción actual no solo afecta la recompensa inmediata, sino que también tiene un impacto en la recompensa en todos los pasos de tiempo subsiguientes. Por ejemplo, una red neuronal puede aprender a jugar el juego de *backgammon* a un alto nivel al usar técnicas de aprendizaje por refuerzo apropiadas (C. Bishop 2006).

5.4 Algoritmos de clasificación

Para construir un modelo de clasificación de múltiples fallas se debe seleccionar el algoritmo de clasificación considerando los siguiente dos tipos:

- i) Tipo paramétrico: Se basan en la aproximación de los datos de entrenamiento a la variable objetivo con un número de parámetros o clases, independientes de la cantidad de datos con los que se entrena;
- ii) Tipo no paramétrico: No da por hecho una forma concreta en el modelo a generar, realizan la clasificación sin suponer el número de clases ni la forma de las distribuciones de probabilidad resultando más flexibles y dando generalmente mejores resultados si se tienen muchos más datos para su entrenamiento.

Para la clasificación de la falla raíz se generó un conjunto de experimentos empleando 5 algoritmos diferentes .de los cuales uno es paramétrico: *Naive Bayes* y los otros cuatro son no paramétricos: *K-Nearest-Neighbor*, *Support Vector Machines*, Árboles de decisión, *Random Forest*.

A continuación, se describe brevemente el funcionamiento general de cada uno de los algoritmos.

5.4.1 *K-Nearest-Neighbor (K-NN)*

K-Nearest Neighbors (K Vecinos más cercanos), es un algoritmo de tipo supervisado de ML no paramétrico. Basado en instancias, puede usarse para clasificación) y regresión. Este algoritmo no aprende explícitamente un modelo, simplemente almacena instancias de los datos de entrenamiento y clasifica nuevos datos basados en medida de similitud.

Como ventaja se puede decir que es un algoritmo sencillo de aprender e implementar, pero como desventaja es que utiliza todo el conjunto de datos para entrenar cada punto (vecino).

Un objeto se clasifica por mayoría de votos de sus vecinos y el objeto se asigna a la clase más común entre sus K vecinos más cercanos, por lo tanto, la salida es una pertenencia a una clase. Donde K es un número entero positivo, normalmente pequeño y representa la cantidad de vecinos para clasificar. Si por ejemplo K se le asignara el valor de 1, el objeto se asignaría a la clase de ese vecino más cercano.

En la Figura 26 se tiene como entrada $K=4$, como vecinos más cercanos, por lo tanto, el *objeto a clasificar* se clasifica en la clase a la que pertenece la mayoría de vecinos de acuerdo a una búsqueda con base en distancias Euclidianas (4) o de Manhattan. En este caso, la clase mayoritaria es la clase A.

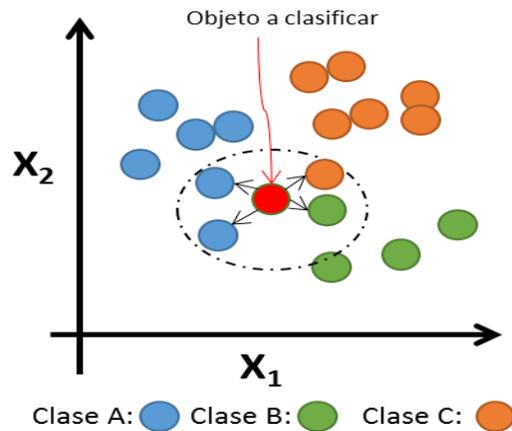


Figura 26: Algoritmo KNN

$$D(i, j) = \sqrt{\sum_{q=1}^N (x_{iq} - x_{jq})^2} \quad (4)$$

En la Figura 27 se observa la interpretación de la distancia euclidiana entre dos puntos P_1 y P_2 de coordenadas (x_1, x_1) y (x_2, x_2) respectivamente.

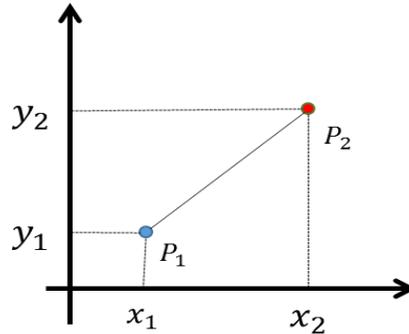


Figura 27: Distancia Euclidiana entre dos puntos

Procedimiento de búsqueda del algoritmo K -NN:

- i) Calcular la distancia entre el objeto a clasificar y el resto de objetos del conjunto de datos de entrenamiento.
- ii) Seleccionar los K elementos más cercanos (con menor distancia).
- iii) Realizar una votación de mayoría de vecinos más cercanos. Los K vecinos que dominen de una clase decidirá la clasificación final.

5.4.2 Support Vector Machines

Máquinas de soporte vectorial comúnmente conocido como *Support Vector Machines* (SVM) consiste en un conjunto de algoritmos de tipo aprendizaje supervisado basado en funciones. Se puede usar como clasificador discriminatorio definido por un hiperplano de separación. El hiperplano es una línea que divide el plano en dos partes para separar las clases, tal y como se muestra en la Figura 28, donde dado un conjunto de datos de entrenamiento, el algoritmo genera un hiperplano óptimo para clasificar los nuevos ejemplos en dos espacios dimensionales. Los vectores de soportes definen los límites de decisión.

En este ejemplo los objetos pertenecen a la clase A o B, la línea de separación define un límite en el lado derecho (clase B) e izquierdo (clase A). Cualquier objeto nuevo será clasificado a partir esta línea de separación.

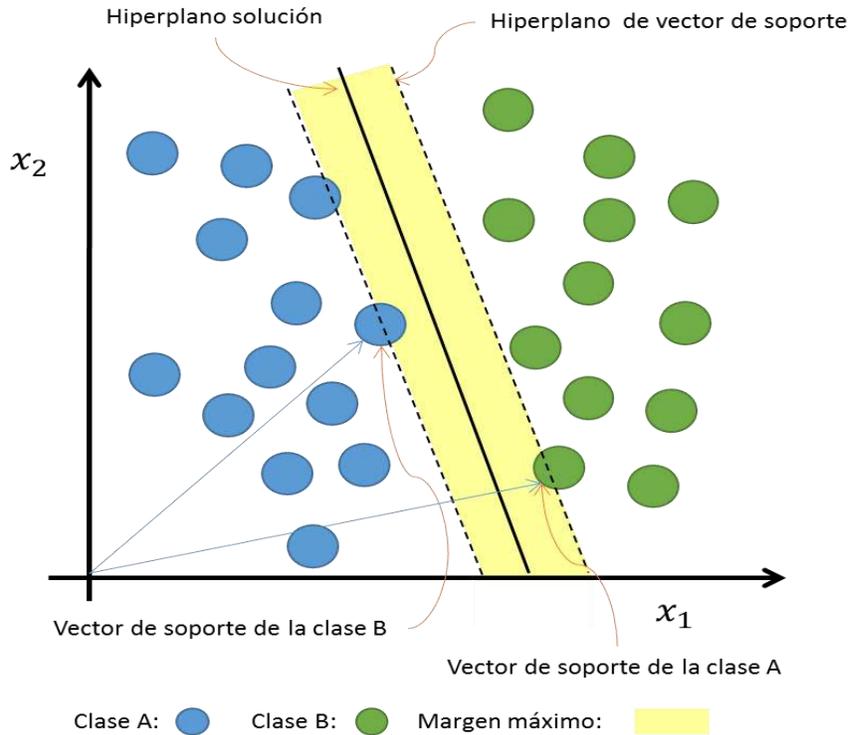


Figura 28: Algoritmo SVM

En este ejemplo, es fácil obtener un hiperplano lineal entre estas dos clases, pero en otros casos, cuando los datos no son linealmente separables, se deben utilizar Funciones Kernel (núcleo) para transformar el espacio de los datos de entrada a un espacio de mayor dimensionalidad y ahí encontrar el hiperplano que mejor se ajuste al conjunto de entrada.

Algunas ventajas del SVM son:

- Es versátil, se pueden especificar diferentes funciones del núcleo para la función de decisión.
- Diferentes funciones Kernel evitan el sobreajuste usando parámetros de regularización.

Algunas desventajas del SVM:

- Son ineficientes para entrenar si el número de características es mucho mayor que el número de muestras (problema de dimensionalidad).
- No proporcionan directamente estimaciones de probabilidad, éstas se calculan a partir de una validación cruzada quíntuple.

5.4.3 Naive Bayes (NB)

Es una técnica de clasificación estadística basado en el teorema de Bayes (5). Se le conoce como algoritmo “ingenuo o simple” porque calcula las probabilidades condicionales de cada clase por separado, como si fueran clases independientes. Esta técnica supone que el efecto de una característica particular en una clase es independiente de otras características.

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (5)$$

Para hacer una predicción, podemos calcular las probabilidades de la instancia que pertenece a cada clase y seleccionar el valor de la clase con la probabilidad más alta.

Con el teorema de Bayes, usando independencia condicional, dado una clase de variable donde y es la clase a estimar y x_1 un vector de características dependientes obtenemos lo siguiente:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \rightarrow \text{posteriori} = \frac{\text{priori} \times \text{likelihood}}{\text{evidencia}} \quad (6)$$

donde:

$P(y)$ es la probabilidad *priori* de y basada en los datos históricos, es decir, es la probabilidad de que la hipótesis y sea verdadera (independientemente de los datos).

$P(y|x_1, \dots, x_n)$ es la probabilidad *posteriori* de la clase (target), esto es la probabilidad de la hipótesis y dado los datos x_1, \dots, x_n (la distribución de probabilidad final para la hipótesis).

$P(x_1, \dots, x_n|y)$ es la probabilidad de predecir una clase determinada dado que la hipótesis fue verdadera; es decir la probabilidad de observar los valores del predictor.

$P(x_1, \dots, x_n)$ es la probabilidad *priori*, es decir, la probabilidad de los datos (independiente de la hipótesis).

Pasos para calcular la probabilidad condicional de un evento:

- i) Calcular la probabilidad previa para las etiquetas de clase dadas.
- ii) Encontrar la relación de probabilidad (likelihood) con cada atributo para cada clase.
- iii) Con el resultado del paso ii, calcular la probabilidad posterior con el teorema de Bayes.
- iv) Seleccionar la clase que tiene la probabilidad más alta.

Algunas ventajas del NB:

- Es rápido y facilita la predicción de la clase (es) dado un conjunto de datos de prueba.
- Necesita pocos datos de entrenamiento.
- Tiene un buen desempeño con variables de entrada categóricas en comparación con variables numéricas (en las cuales, se supone una distribución normal).

Una desventaja del NB es que si en el conjunto de datos de prueba, la variable categórica tiene una categoría que no se observó en el conjunto de datos de entrenamiento, el modelo asignará un cero, por lo tanto, no podrá hacer una predicción.

5.4.4 Árboles de decisión

Decision Tree Classification (DT) es comúnmente conocido como Árbol de Decisión. Es un algoritmo de tipo aprendizaje supervisado usado para problemas de clasificación y regresión. Podemos decir que los arboles de decisión desarrollan una secuencia de pruebas para obtener una decisión. Los arboles están formados por nodos de decisión que son los nombres de los atributos. Los atributos son las características que intervienen en la clasificación, las ramas son los posibles valores del atributo asociado al nodo y las hojas son conjuntos clasificados de ejemplos y etiquetados con el nombre de una clase, donde las clases es el valor previsto de la salida dada la entrada.

Para construir árboles que representen los datos y que contribuyan a reducir el tamaño del árbol de decisión, se utiliza el algoritmo ID3 (Induction Decision Trees) (QUINLAN 1986) el cual usa la teoría de información de Shannon (Shannon 1948).

Este algoritmo es recursivo, no utiliza *backtracking*, realiza ganancia de información y el resultado puede expresarse como un conjunto de reglas *if – then*.

En el ejemplo presentado en la Tabla 5 se describe la construcción y el funcionamiento de un árbol de decisión dado los siguientes tres pasos:

- i) Calcular la entropía total (7) de todas las clases.
- ii) Seleccionar el mejor atributo basado en la reducción de la entropía. Se debe calcular la ganancia de información (9) la cual se basa en la disminución de la entropía (8) cuando un conjunto de datos se divide en los valores de un atribuido seleccionado.
- iii) Iterar hasta que todos los atributos sean clasificados.

$$H(P_1, \dots, P_m) = - \sum_{i=1}^m P_i \log_2 P_i \quad (7)$$

donde:

$H(P_1, \dots, P_m)$ es la cantidad de Información necesaria dado una tupla.

P_i es la probabilidad de posibles valores.

m es el número de ejemplos de entrenamiento dado un variable.

Por lo tanto, la entropía H de la clase C dado un atributo A es dada por:

$$H(C|A) = \sum_{j=1}^m P(a_j) * H(C|a_j) \quad (8)$$

donde, a_j representa a los valores de un atributo.

Y la ganancia de información se expresa por:

$$Gain(A) = H(C) - H(C|A) \tag{9}$$

Para generar el ejemplo se utiliza la siguiente tabla de valores.

No.	Programación	Comunicación	Presentación	Decisión
1	bueno	bien	elegante	si
2	promedio	bien	normal	si
3	malo	mentiroso	fachoso	no
4	promedio	mentiroso	normal	no
5	bueno	aburrido	fachoso	si
6	bueno	bien	fachoso	si
7	promedio	aburrido	fachoso	no
8	bueno	mentiroso	fachoso	no
9	malo	aburrido	normal	no
10	malo	aburrido	fachoso	no

Tabla 5: Atributos y clases para generar un árbol d decisión

Para generar el árbol de decisión se necesita elegir un nodo raíz calculando la ganancia de información de todos los atributos. Para la primera iteración el número de ejemplos de entrenamiento será 10.

Decisión	
si	no
4	6

Tabla 6: Conteo total de la clase “si” y la clase “no”

Dado el conteo de las clases se calcula la entropía total: $H(Decisión) =$

$$I(4,6) = 0.971$$

		Decisión		$P(aj)$	$H(Decisión aj)$
		si	no		
Programación	aj	$P(si aj)$	$P(no aj)$		
	bueno	3/4	1/4	4/10	0.811
	promedio	1/3	2/3	3/10	0.918
malo	0/3	3/3	3/10	0	

Tabla 7: Cálculo de la entropía de la clase Decisión dado Programación y su ganancia de información

$$H(Decisión|Programación) = 0.6$$

$$Gain(Programación) = 0.371$$

		Decisión		$P(aj)$	$H(Decisión aj)$
		si	no		
Comunicación	aj	$P(si aj)$	$P(no aj)$		
	bien	3/4	1/4	4/10	0
	mentiroso	1/3	2/3	3/10	0
aburrido	0/3	3/3	3/10	0.811	

Tabla 8: Cálculo de la entropía de la clase Decisión dado Comunicación y su ganancia de información

$$H(Decisión|Comunicación) = 0.325$$

$$Gain(Comunicación) = 0.646$$

		Decisión		$P(aj)$	$H(Decisión aj)$
		si	no		
Presentación	aj	$P(si aj)$	$P(no aj)$		
	elegante	3/4	1/4	4/10	0
	normal	1/3	2/3	3/10	0.918
fachoso	0/3	3/3	3/10	0.918	

Tabla 9: Cálculo de la entropía de la clase Decisión dado Presentación y su ganancia de información

$$H(Decisión|Presentación) = 0.826$$

$$Gain(Presentación) = 0.144$$

Calculando la ganancia de información para cada variable se observa que la variable “Comunicación” es la de mayor puntuación, por lo tanto, se toma como nodo raíz, y las ramas con entropía cero se convierten en hojas dando como resultado el siguiente árbol de decisión como primer paso.

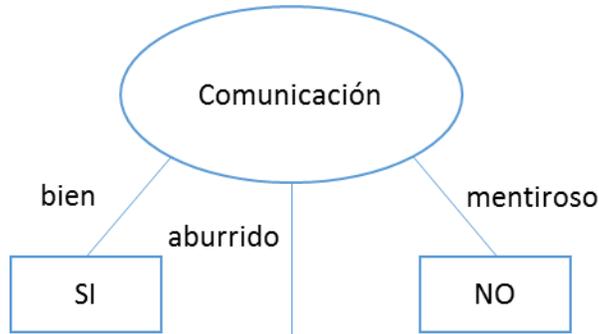


Figura 29: Árbol de decisión identificando como nodo raíz a la variable “Comunicación”

Se observa en los cálculos que el atributo aburrido no concluye a una hoja, por lo tanto, se necesita seguir calculando nuevos valores para seguir sub-dividiéndose para una mejor clasificación. Para continuar con la construcción del árbol, el número de ejemplos de entrenamiento ahora será 4 de la variable “Comunicación”

aburrido ∈ Comunicación	
si	no
1	3

Tabla 10: Conteo total del atributo aburrido para la clase “si” y para la clase “no”

Dado el conteo de las clases se calcula la entropía total: $H(aburrido) = I(1,3) = 0.811$

		aburrido ∈ Comunicación		P(aj)	H(aburrido aj)
		si	no		
Programación	aj	P(si aj)	P(no aj)		
	bueno	1/1	0/1	1/4	0
	promedio	0/1	0/1	1/4	0
malo	0/2	2/2	2/4	0	

$$H(\text{aburrido}|\text{Programación}) = 0$$

$$\text{Gain}(\text{Programación}) = 0.811$$

		aburrido ∈ Comunicación		P(aj)	H(aburrido aj)
		si	no		
Presentación	aj	P(si aj)	P(no aj)		
	elegante	0/0	0/0	0/4	0
	normal	0/1	1/1	1/4	0
fachoso	1/3	2/3	3/4	0.918	

$$H(\text{aburrido}|\text{Presentación}) = 0.689$$

$$\text{Gain}(\text{Presentación}) = 0.122$$

Calculando la ganancia de información para cada variable se observa que la variable “Programación” es la de mayor puntuación, por lo tanto, se toma como el siguiente nodo, y las ramas con entropía cero se convierten en hojas. Por lo tanto, el árbol decisión final es el siguiente:

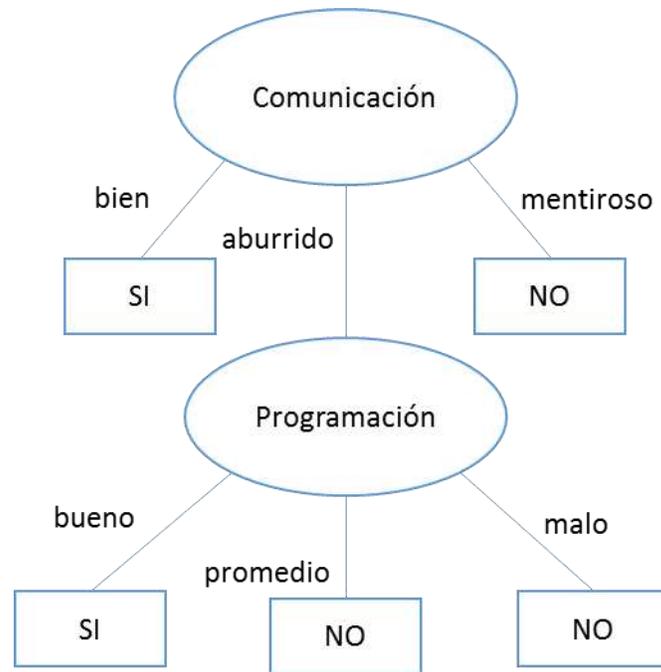


Figura 30: Árbol de decisión resultante

Entonces tenemos que el funcionamiento del árbol construido es el siguiente:

IF (Comunicación = bien) THEN SI

IF (comunicación = mentiroso) THEN NO

IF (Comunicación = aburrido) AND (Programación = bueno) THEN SI

IF (Comunicación = aburrido) AND (Programación = promedio) THEN NO

IF (Comunicación = aburrido) AND (Programación = malo) THEN NO

Algunas ventajas de los DT son:

- Cuantifica el costo de un resultado y la probabilidad de que suceda.
- Puede usarse para la selección de variables.

Algunas desventajas de los DT son:

- Puede incurrir en un sobre entrenamiento (*overfitting*).
- Los DT están sesgados con un conjunto de datos desequilibrados, se recomienda equilibrar el conjunto de datos de entrenamiento antes de crear el árbol.
- Requiere datos completos

5.4.5 *Random Forest*

Random Forest (RF) comúnmente conocido como árboles aleatorios. Es un algoritmo de tipo aprendizaje supervisado usado para problemas de clasificación o regresión. Es un algoritmo de ensamble (combinación de modelos), el cual consiste en un gran número de árboles de decisión individuales que operan como un conjunto, donde cada árbol arroja una predicción de clase, y a través de una votación, la clase con la mayor cantidad de votos se convierte en la predicción del modelo. Este algoritmo utiliza “*bagging*” (empaquetamiento) y realiza la selección de características de forma aleatoria para construir cada árbol individual, con la finalidad de crear un bosque no correlacionado de árboles.

Este modelo se considera aleatorio por las siguientes características:

Muestreo aleatorio sobre los datos de entrenamiento

Al entrenar cada árbol aprenden de un subconjunto de muestras aleatorias del conjunto de datos de entrenamiento. Los subconjuntos se extraen con reemplazo, conocido como (*bootstrapping* o *bagging*) ver Figura 31. En el momento de la prueba, las predicciones se realizan promediando las predicciones de cada árbol de decisión. El objetivo al entrenar a cada árbol con diferentes muestras es que el todo el bosque tenga una varianza menor, pero no a cambio de aumentar el sesgo (*bias*).

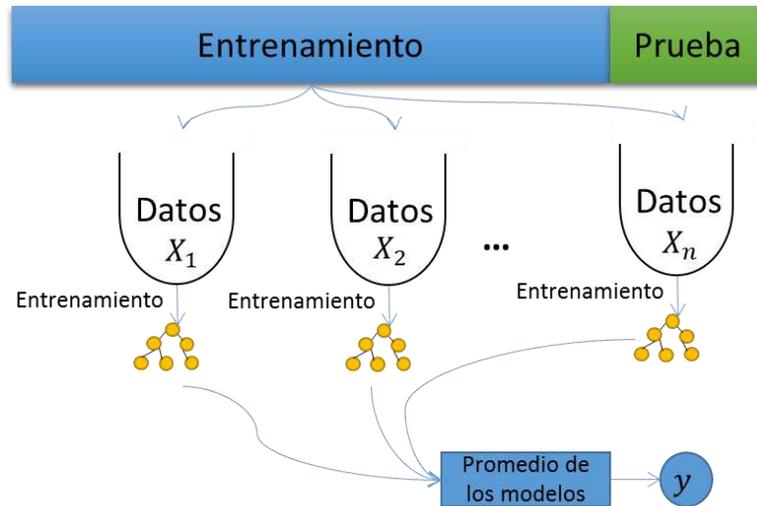


Figura 31: Idea general de *Bagging*

Subconjuntos aleatorios de características considerados al seleccionar nodos

Es donde cada árbol obtiene el conjunto completo de características, pero en cada nodo, solo se considera un subconjunto aleatorio de características.

Procedimiento general del funcionamiento del algoritmo RF:

- i) Seleccionar un subconjunto de muestras aleatorias del conjunto de datos de entrenamiento.
- ii) Generar un árbol de decisión para cada muestra con su respectivo resultado de predicción.
- iii) Hacer una votación para cada resultado de cada árbol.
- iv) Seleccionar la predicción con más votos como resultado.

Algunas ventajas de los RF son:

- Cuantifica el costo de un resultado y la probabilidad de que suceda.
- Puede usarse para la selección de variables.
- Se consideran como un método muy preciso y robusto por la forma de operar.
- Se utiliza para la selección de características importantes

Algunas desventajas de los DT son:

- En algunos casos puede presentar sobreajuste en ciertos grupos de datos en problemas de clasificación.
- Pueden llegar a ser tardado por los múltiples arboles de decisión.
- La clasificación hecha por el RF es difícil de interpretar.

5.5 Evaluación de modelo

La evaluación y la validación cruzada son procedimientos básicos en el aprendizaje automático para medir el rendimiento de un modelo de clasificación o predicción sobre nuevos datos, los cuales no han sido vistos por el modelo, indicando el nivel de aciertos de las predicciones en este nuevo conjunto de datos. Debido a que el rendimiento de un modelo sobre el conjunto de entrenamiento es una estimación optimista de su verdadero rendimiento con nuevos datos, se debe evaluar con datos estadísticamente independientes con los que se entrenó, con la finalidad de obtener buena generalización del modelo.

Los diferentes tipos de aprendizaje automático tienen diferentes métricas de evaluación de rendimiento de un problema (Hossin y Sulaiman 2015). Para evaluar un modelo correctamente mediante distintas métricas de evaluación debemos contar con muestras de datos lo suficientemente grandes para generar dos conjuntos de

datos. El primer conjunto se etiqueta como datos de entrenamiento y el segundo como conjunto de datos de prueba. Generalmente suelen tenerse muestras con un porcentaje de separación del 70-30 %, 80-20% o 90-10%.

Podemos realizar la evaluación de tres formas distintas:

- i) Generar predicciones sobre los datos de entrenamiento y evaluar el modelo con base en estas predicciones.
- ii) Generar predicciones en el modelo, y compararlas con las predicciones generadas con un conjunto de datos de prueba.
- iii) Comparar las puntuaciones de dos modelos diferentes, pero relacionados, utilizando el mismo conjunto de datos.

Para el caso de clasificación binaria las métricas más utilizadas se describen a continuación. Estas métricas pueden ser escaladas para problemas de clasificación multiclase.

- Exactitud
- Precisión
- Recall
- F1 score
- ROC-AUC
- Matriz de confusión
- Curvas de Precisión-Recall

Exactitud o “Accuracy”

Es el porcentaje total de elementos clasificados correctamente, la cual es dada por el número total de predicciones correctas entre el número total de predicciones.

Para el caso de datos balanceados, podemos suponer que la medición de exactitud del modelo es confiable, en caso contrario se debe evaluar el modelo con otras métricas dado a que puede generar conclusiones no correctas.

Para clasificación binaria:

$$acc = \frac{tp + tn}{tp + tn + fp + fn} \quad (10)$$

Para clasificación multiclase:

$$acc = \frac{\text{No. total de predicciones correctas}}{\text{No. total de predicciones}} \quad (11)$$

donde:

tp = Cuando la clase real es verdadera y lo que se predijo fue verdadero.

tn = Cuando la clase real es falso y lo que se predijo fue falso.

fp = Cuando la clase real es falso y lo que se predijo fue verdadero.

fn = Cuando la clase real es verdadera y lo que se predijo fue falso.

Precisión

Es la proporción con la que un modelo predijo correctamente la clase positiva. Se define como la relación de las observaciones positivas predichas correctamente dividido entre el total de observaciones positivas predichas. La precisión responde a la siguiente pregunta: de todas las etiquetas positivas posibles, ¿Cuántas identifico correctamente el modelo?

Para calcular la precisión de modelo se realiza en función a la siguiente formula:

$$Precisión = \frac{tp}{tp + fp} \quad (12)$$

Recall (sensitivity)

También conocida como la tasa de verdaderos positivos (TPR), es el porcentaje de instancias positivas correctamente clasificada, es decir, la capacidad del modelo para predecir la clase correctamente.

Se define como el número de muestras estimadas correctamente dividido entre el número de todos los elementos relevantes y es dada por:

$$Recall = \frac{tp}{tp + fn} \quad (13)$$

Con la medición correcta de la Precisión y Recall se tienen cuatro posibles escenarios para cada clase:

- i) Alta Precisión y alto Recall, indican que el modelo clasifica perfectamente esa clase.
- ii) Alta Precisión y bajo Recall, el modelo no detecta muy bien la clase, pero cuando lo hace es altamente confiable.
- iii) Baja Precisión y alto Recall, el modelo detecta bien la clase, pero también incluye muestras de otras clases.
- iv) Baja Precisión y bajo Recall, el modelo no clasifica la clase correctamente.

Cuando se tiene un conjunto de datos desbalanceados, podemos obtener un valor alto en la precisión en la clase con la mayoría de los datos y un bajo Recall en la clase con la minoría de datos.

F1-measure (F-Score)

Es una métrica para cuantificar el rendimiento del modelo. Es útil para datos desbalanceados porque intenta encontrar el equilibrio entre la precisión y el Recall. Se calcula como el promedio ponderado de la Precisión y Recall dentro de un rango entre 0 y 1, donde el valor ideal de la puntuación F es 1. Indica qué tan preciso es el modelo de clasificación y qué tan robusto. F1-Score intenta encontrar el equilibrio entre precisión y recuperación y está dado por:

$$F1 = 2 * \frac{\text{Precisión} * \text{Recall}}{\text{Precisión} + \text{Recall}} \quad (14)$$

Para problemas multiclase la Precisión y el Recall son valores promedio de todas clases.

ROC-AUC

ROC (*Receiver Operating Characteristic*) (Bewick, Cheek y Ball 2004) es una gráfica para clasificación binaria la cual permite mostrar el rendimiento de un modelo de aprendizaje de clasificación. La gráfica representa la distribución de la tasa de verdaderos positivos (TPR) conocida como sensibilidad en el eje y, frente a la tasa de falsos positivos (FPR) conocida como (1- especificidad) en el eje x. Cada resultado de predicción representa un punto en el espacio ROC.

Donde la especificidad es la probabilidad de clasificar correctamente a un caso cuyo estado real sea clasificado como negativo, y es dado por la siguiente formula:

$$\text{Especificidad} = \frac{tn}{tn + fp} \quad (15)$$

$$FPR = \frac{fp}{fp + tn} = 1 - \text{Especificidad} \quad (16)$$

La curva ROC nos permite comparar modelos a través del área bajo de la curva ROC (AUC), donde el rendimiento del modelo se representa con un puntaje entre 0 y 1, por lo tanto, mide la separabilidad entre las probabilidades de las clases negativas y las clases positivas. Para ejemplos, en multiclase se puede generar curvas por cada clase o generar un promedio micro o macro de todas las clases.

5.6 Validación cruzada

La validación cruzada (*cross-validation*) es una técnica que reduce el sobreajuste (*overfitting*). Puede ser aplicada a cualquier algoritmo de aprendizaje. La idea básica es estimar la calidad de cada hipótesis en la predicción de datos no observados. Esto se hace separando una parte de datos conocidos y utilizándola para medir la calidad de la predicción de una hipótesis inducida a partir de los datos restantes. La validación cruzada de k pliegue (*K-fold cross-validation*) consiste en realizar k experimentos, dejando a un lado cada vez $1/k$ de los datos para prueba y promediando los resultados. Los valores comúnmente usados de k son 5 y 10. El extremo es $k = n$, también conocido como: “validación cruzada omitiendo uno” (*leave-one-out-cross-validation*) (Russell y Norving 2010). En la Figura 32 se describe la validación cruzada usando el valor de k igual a 5, donde el porcentaje de entrenamiento y prueba normalmente son de 70-30%, 80-20, 90-10%.

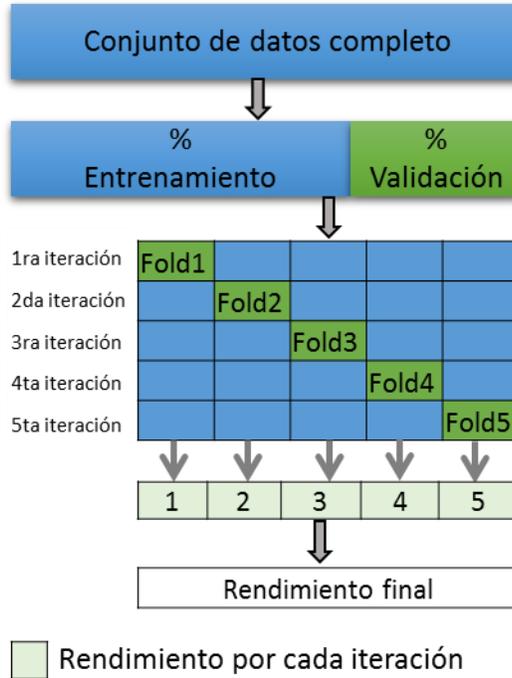


Figura 32: Validación cruzada con $k=5$

5.7 Experimentación

En el presente capítulo se describen el procedimiento de reconocimiento de patrones de la Figura 21 así como los resultados obtenidos y comparaciones entre los experimentos.

Para la experimentación se utilizó el patrón de falla generado por el DxAG, el cual se describe en el capítulo 2.

5.7.1 Formación de objetos

Partiendo de un conjunto de patrones de comportamiento generado por el sistema DxAG (ver Figura 19) se continúa con el proceso de generar características

presentes en patrones adquiridos de cada falla, los cuales pueden ser pasados directamente a la etapa de clasificación de fallas.

Tomando como ejemplo el patrón de comportamiento de la falla F1 de la Figura 18 donde el eje de las “x” representa el tiempo en segundos del conjunto de variables y el eje de las “y” el porcentaje de probabilidad de variables con comportamiento anormal, se supone como una imagen de un tamaño $n \times m$ donde n sería el número de los registros simulados y m el conjunto de variables con comportamiento anormal, tal como se describe en capítulos anteriores. Tomando en cuenta que el objetivo del reconocimiento de patrones es asignar un patrón a la clase a la que pertenece, es necesario extraer características que permitan representarse y describirse como un vector de características cuantitativas globales. En la Figura 33 se describe una matriz de m registros por n variables involucrados en un patrón de falla a la cual se procede a generar un vector de características cuantitativas donde X es el patrón y x son las características descriptivas.

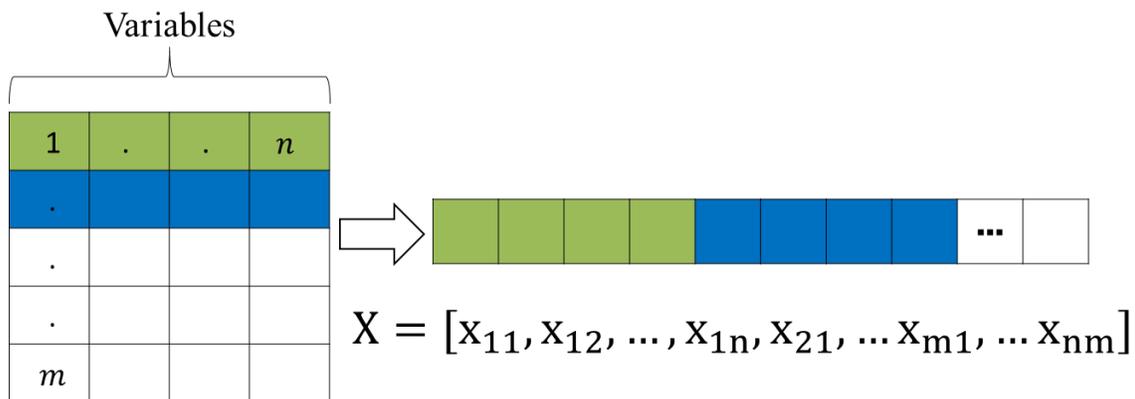


Figura 33: Vector de características cuantitativas

La extracción de características partiendo del patrón de comportamiento de cada falla tiene como objetivo obtener un único un patrón de falla a clasificar que contenga la mayor información representativa de ésta. Para los siguientes experimentos es necesario etiquetar los patrones generados, para eso se cuenta con

un catálogo de falla (ver Tabla 1) donde se describe la función de cada falla simulada en la MEM. Una vez formados todos objetos se genera un conjunto de datos completos al cual se le denominara conjunto de patrones de falla (ver Figura 34) formado por atributos y clases que son las etiquetas de cada patrón de falla, el cual representara en su momento a dos sub-conjuntos de datos uno para entrenar el modelo de clasificación y otro para probar su rendimiento.

Atributos				Clases
$X_{1,1}$.	.	$X_{1,n}$	F0
.				.
.				.
.				.
$X_{m,1}$				F6

Figura 34: Conjunto de Patrones de Falla

Donde:

n es el número total de atributos

m es el número total de patrón de falla (ejemplos de entrenamiento)

El conjunto de patrones de falla se formó con datos balaceados donde todas las clases tienen la misma cantidad de datos, tal y como se muestra en la Figura 35, donde en el eje de las x son las clases, las cuales representan las fallas etiquetadas y el eje de las y, la cantidad de patrones etiquetados.

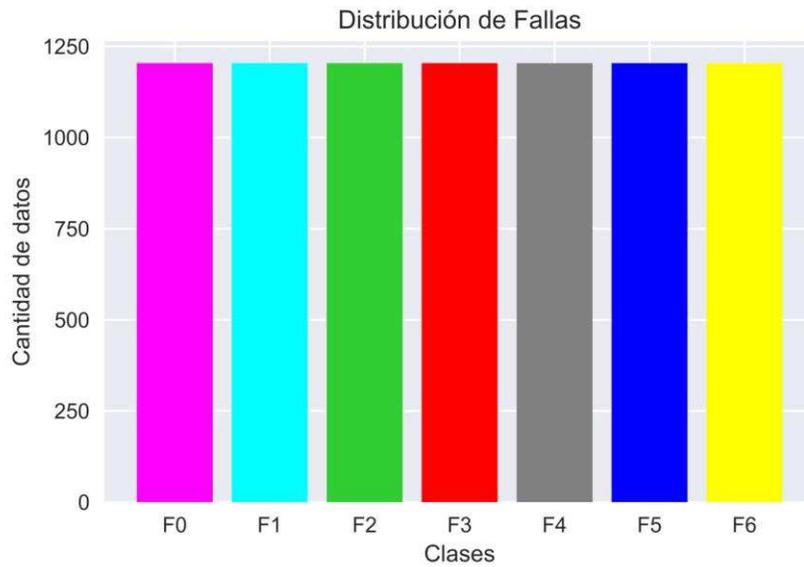


Figura 35: Distribución de Fallas

5.7.2 Reducción de dimensionalidad

Antes de evaluar los algoritmos de clasificación es importante realizar un análisis previo del conjunto de datos para eliminar atributos redundantes e irrelevantes. La selección de características o variables importantes es un proceso donde se busca obtener el mejor subconjunto de características dentro de un conjunto de datos mediante una búsqueda automática, por lo tanto, es necesario extraer aquellas variables que sean útiles para distinguir los patrones de cada falla. Además, ayuda a mejorar la precisión del modelo de clasificación, ayuda a disminuir el tiempo de entrenamiento y colabora a obtener menos datos redundantes.

La selección de características mediante el algoritmo de aprendizaje *Random Forest* (RF) se implementa mediante algoritmos que tienen sus propios métodos de selección de características incorporados. Por lo tanto, cuando se entrena un árbol es posible calcular cuánto disminuye la entropía de cada variable, cuanto mayor es la disminución de la entropía más significativa de la variable. Hacer una selección de

variables mediante las características importantes derivadas de árboles es una forma muy sencilla, rápida y generalmente precisa.

Para ejemplificar el funcionamiento de selección de características importantes, en la Figura 36 se muestran 29 características diferentes del conjunto de datos de Patrones de Fallas, las cuales se clasifican por sus respectivas medidas de importancia relativa. Para este proceso las características importantes se normalizan para que la suma de sus valores de cada característica sea de igual a 1. Para este ejemplo se observa que la variable “Var_3748” es la de menos importancia y la variable “Var_234” es la de mayor importancia.

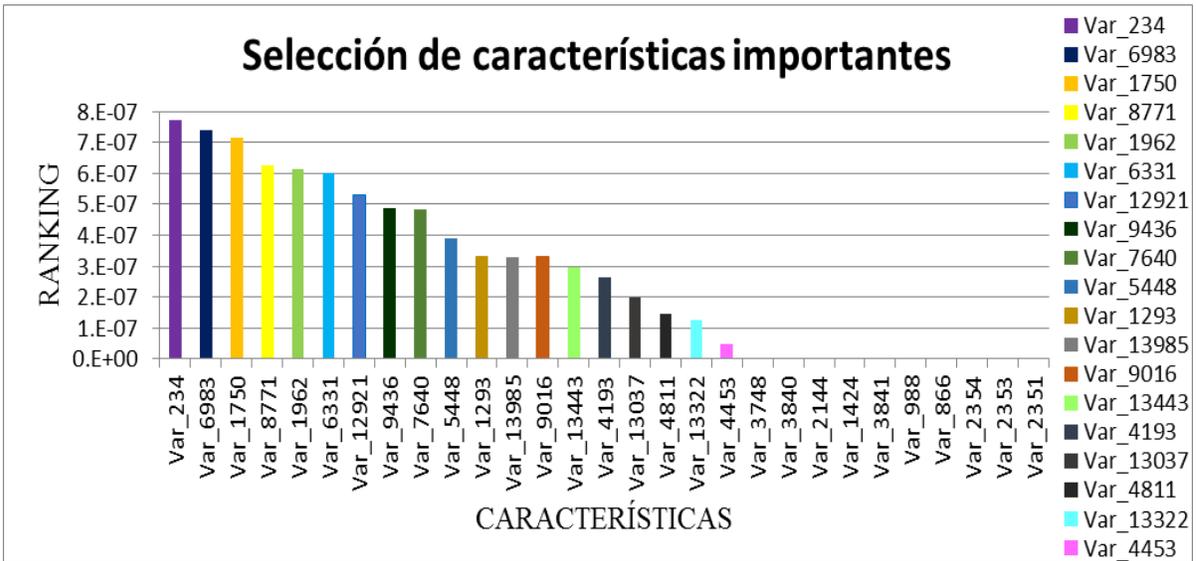


Figura 36: Ejemplo de Selección de características

Para reducir la dimensionalidad del conjunto de Patrones de Falla se creó un modelo de selección de características importantes basado en el algoritmo RF mediante una visualización de los datos (ver Figura 37). Se seleccionaron todas aquellas variables con puntaje mayor a cero logrando una reducción del 37 % sobre el total de las variables seleccionadas.

En la Figura 37 se clasifican 15,200 características donde en el eje x, describen las variables a evaluar y el eje y describe la importancia de la variable, dado por valores normalizados donde el valor mínimo de importancia es de 0 y el valor máximo de importancia es 0.0062.

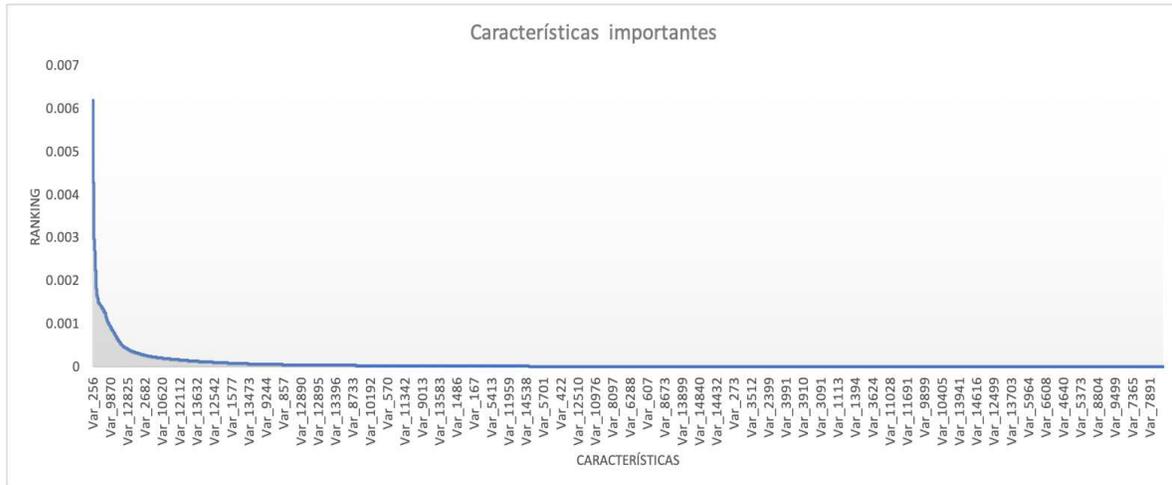


Figura 37: Características importantes

5.7.3 Modelo de clasificación

Se seleccionaron 5 algoritmos de clasificación para determinar qué modelo es el mejor para identificar relaciones y patrones entre variables para saber a qué clase pertenece un patrón de falla dado. Para estas comparaciones se utilizó el conjunto de patrones de falla sin discriminar ninguna variable, la validación de cada modelo se realizó con la técnica de validación cruzada usando 10 interacciones (90 % para entrenar y 10 % para probar).

Para esta etapa de experimentos se eligieron los siguientes 5 algoritmos de clasificación:

- i) *Random Forest (RF)*
- ii) *K-Nearest Neighbors (KNN)*
- iii) *Decision Tree (DT)*

- iv) *Gaussian Naive Bayes (GNB)*
- v) *Support Vector Classification (SVC)*

Para cada algoritmo se utilizó los siguientes parámetros de configuración

RandomForestClassifier:

- *n_estimators* = 100 árboles.
- *max_features* = Automático. Número máximo de características consideradas para dividir un nodo
- *max_depth* = Automático. Número máximo de niveles en cada árbol de decisión
- *bootstrap* = sí. Método para muestrear puntos de datos (con o sin reemplazo)

KNeighborsClassifier:

- *n_neighbours*: 5,10 y16 Número de vecinos.
- *algorithm*: Automático, Se elige el algoritmo más apropiado en función de los valores
- métrica: *Minkowski*. Métrica de distancia a utilizar.

DecisionTreeClassifier:

- *criterion*: *Gini*
- *max_features* : *Default*
- *max_depth* : *Default*

GaussianNB:

- No se establece ningún parámetro

SVC:

- *kernel*: lineal, gaussiano y sigmoide. Tipo de núcleo que se utilizó en el algoritmo. Este caso se probó con 3 tipos de los cuales se eligió el mejor.
- Gamma: Automático. Coeficiente de *kernel* para ‘*rbf*’, y ‘*sigmoid*’.

En la Tabla 11 se contrastan las diferentes medidas de exactitud de los modelos que se obtuvieron con cada algoritmo mediante la validación cruzada.

Algoritmo	% Test Accuracy	[+/-] STD
RF	90.5278	2.8066
KNN	37.8433	1.5085
DT	83.0108	1.4444
GNB	63.7699	1.8997
SVM	52.5845	1.4842

Tabla 11: Medidas de rendimiento

Para una mejor visualización del rendimiento de cada modelo se representan los resultados en un diagrama de caja, tal y como se describe en la Figura 38, donde en el eje x se representan los algoritmos a evaluar y en el eje y, la exactitud de la prueba. Cada diagrama grafica el valor máximo, mínimo y la mediana de la exactitud del modelo que se obtuvo en las 10 iteraciones de la validación cruzada. De esta forma podemos observar que, para nuestro conjunto de datos dado, el algoritmo RF mostró el mejor rendimiento con el 90.52 % de exactitud

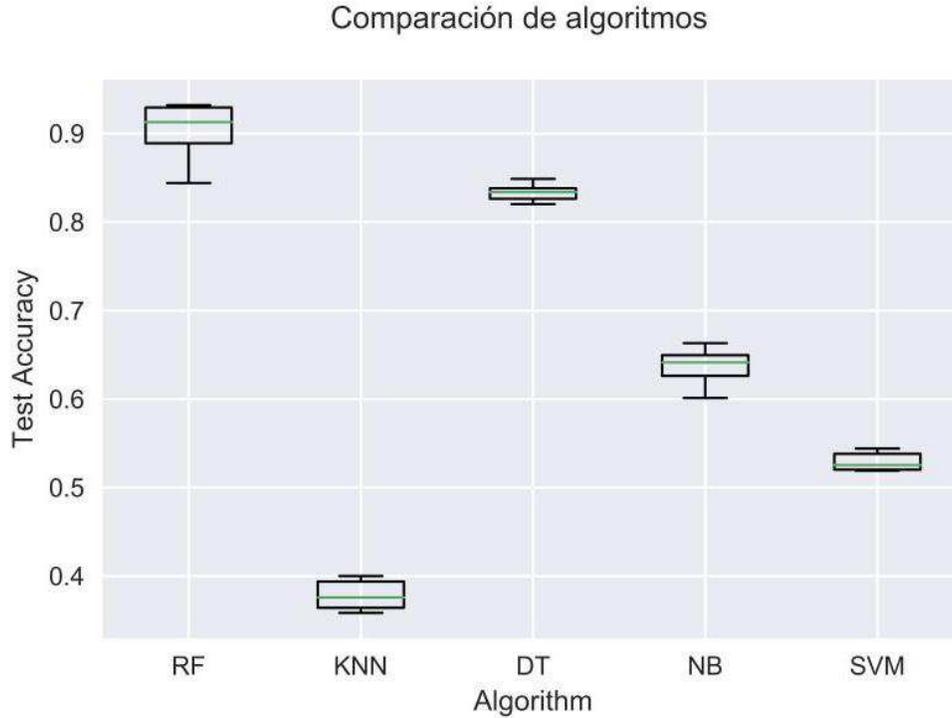


Figura 38: Comparación de algoritmos

5.7.3.1 *Random Forest*

El algoritmo de RF es uno de los más populares y estudiados en cuanto a clasificación. En general, brindan un buen desempeño gracias al ensamble de muchos árboles de decisión. Mientras más árboles en el bosque, más robusto podría ser el algoritmo.

La Figura 39 describe de manera general el funcionamiento del algoritmo como clasificador. Con un número n de árboles de decisión, donde X son las características seleccionadas y c es la predicción de la clase. La capacidad de clasificar con una buena precisión los patrones dados es de vital importancia para diversas aplicaciones en el sector energético, en particular para clasificar fallas raíz de los AGs. Para evaluar y mejorar el desempeño del modelo generado por el algoritmo RF, se realizaron una serie de experimentos donde se utilizaron métricas

de rendimiento, reducción de características, técnicas de análisis de componentes principales (PCA), técnicas de validación cruzada y “*hold-out*”. Para llevar a cabo los siguientes 5 experimentos:

- i) RF sin validación cruzada y sin reducción de características.
- ii) RF sin validación cruzada y con reducción de características
- iii) RF sin validación cruzada y PCA con dos componentes.
- iv) RF sin validación cruzada y PCA con cincuenta componentes.
- v) RF con validación cruzada y con reducción de características.

se usó el conjunto de patrones de falla con clases balanceadas.

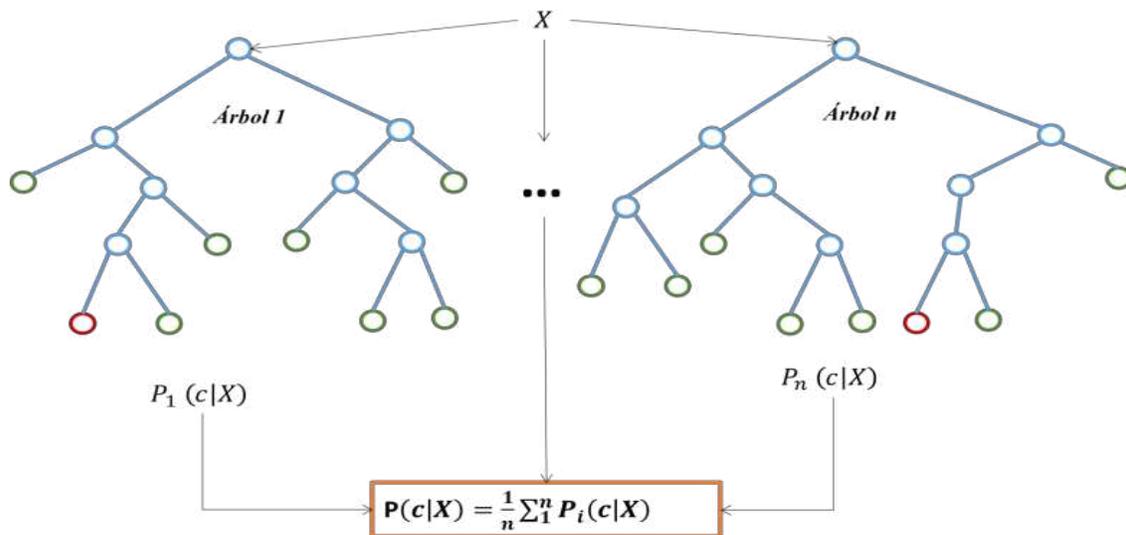


Figura 39: Algoritmo de clasificación *Random Forest*

A continuación, se detalla los experimentos realizados.

5.7.3.2 Experimento 1: RF sin validación cruzada y sin reducción de características

Para este experimento dividimos el conjunto de patrones de falla en 2 sub-conjuntos, técnica conocida como “*hold-out*”, el primer sub-conjunto con el 80% del total de los datos al cual se le etiqueta como conjunto de datos de entrenamiento y al segundo sub-conjunto restante del 20 % etiquetado como conjunto de prueba. Para estos experimentos se utilizó el conjunto de patrones de falla sin discriminar ninguna variable.

La Tabla 12 describe los resultados del primer experimento mediante las métricas de Precisión, Recall, y F1-Score de cada uno de las clases. Con el 88.90% de exactitud y un 89% de precisión generado por el modelo, se puede predecir las distintas fallas evaluadas correctamente. Se supone que el modelo tiene un buen rendimiento de clasificación dado a que la Precisión y el Recall son significativamente altos. Además, se tiene la medida del F1-Score del 89% el cual también nos indica la precisión y robustez del modelo de clasificación.

- Exactitud de la prueba: 88.90%

Clase	Precisión	Recall	F1-Score
F0	0.8	0.84	0.82
F1	0.86	0.89	0.88
F2	0.98	0.97	0.98
F3	0.91	0.84	0.88
F4	0.84	0.88	0.86
F5	1	1	1
F6	0.81	0.78	0.79
Promedio	89%	89%	89%

Tabla 12: Reporte de clasificación sin validación cruzada y sin reducción de características.

La Figura 40 muestra la matriz de confusión donde se describe el éxito de las predicciones de cada clase en términos de porcentaje. Se observa que en el eje x se

describen las fallas estimadas por el modelo y el eje de y las fallas reales que en su momento se etiquetaron manualmente basándose en el catálogo de fallas de la Tabla 1. Tomando como ejemplo la F5 (falla real) etiquetado con el número 5 en la matriz de confusión podemos observar que fue clasificada correctamente como F5 (falla estimada) con el 100%. Esto resulta ser aceptable debido a que es una falla que no presenta un comportamiento de operación similar con el resto de las fallas, por tratarse de un cortocircuito en el generador eléctrico de la MEM. Para el caso F6 se clasifica correctamente con el 77% pero a la vez el modelo lo clasifica incorrectamente como F0 con el 16.1%, con el 4.9 % lo clasifica como F1, con el 0.9% como F3 y con el 0.4% como F4. A medida de seguir realizando experimentos con diferentes configuraciones se puede mejorar el rendimiento de clasificación.

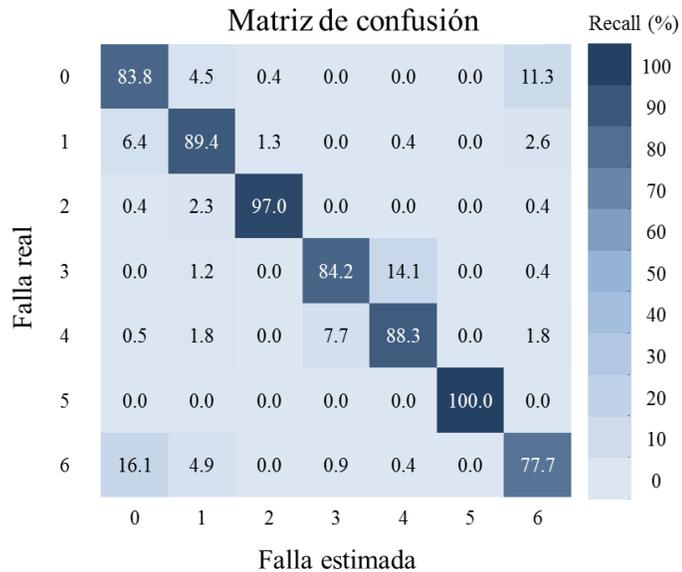


Figura 40: Matriz de confusión sin validación cruzada y sin reducción de características

5.7.3.3 Experimento 2: RF sin validación cruzada y con reducción de características

Este experimento contrasta las ventajas de seleccionar características importantes del conjunto de patrones de fallas. Se reflejan estas ventajas en los

resultados obtenidos de la Exactitud del modelo, así como la Precisión, Recall y f1-score del conjunto de clases analizadas. Para este segundo experimento dividimos el conjunto de patrones de falla en 2 sub-conjuntos, el primero con el 80% para entrenamiento y el segundo con el 20 % para probar. Para este experimento se discrimino el 37% de características dado a que su aportación significativa al conjunto de patrones de falla era mínima.

En la Tabla 13 se describe los resultados del segundo experimento mediante las métricas de Precisión, Recall, y F1-Score de cada uno de las clases. Se observa que, dado a la reducción de características, el modelo presenta un macro-promedio de Precisión, Recall del 90%. Unas de las principales ventajas de este experimento fue reducir el tamaño de características del conjunto de patrones de falla, parámetro que favorece en tiempo de entrenamiento y rendimiento del modelo de clasificación.

La exactitud reportada para este modelo es del 89.67 %. Se supone que el modelo tiene un buen rendimiento de clasificación dado a que la precisión y el Recall son significativamente altos. Además, se tiene un score del 90% el cual también nos indica que tan de preciso y robusto es el modelo de clasificación.

- Exactitud de la prueba: 90.39%

Clase	Precisión	Recall	F1-Score
F0	0.81	0.87	0.84
F1	0.86	0.9	0.88
F2	0.98	0.96	0.97
F3	0.92	0.84	0.88
F4	0.84	0.89	0.86
F5	1	1	1
F6	0.87	0.8	0.84
Promedio	90%	90%	90%

Tabla 13: Reporte de clasificación sin validación cruzada y con reducción de características

La Figura 41 muestra la matriz de confusión donde se describe el éxito de las predicciones de cada clase. Se observa que el eje x describe las fallas estimadas, las cuales se encargan de predecir el modelo y el eje de y las fallas reales que en su momento se etiquetaron manualmente basándose en el catálogo de fallas de la Tabla 1.

Tomando como ejemplo a F5 se observa que sigue conservando su capacidad de ser estimada correctamente, esto debido a su comportamiento de operación. Para la F6 con este modelo se observa un incremento en su capacidad de ser estimada correctamente con el 80.4%, con el 13.8% lo clasifica incorrectamente como F0, con el 4.5.9% lo clasifica como F1, con el 0.4% como F3 y con el 0.9% como F4.

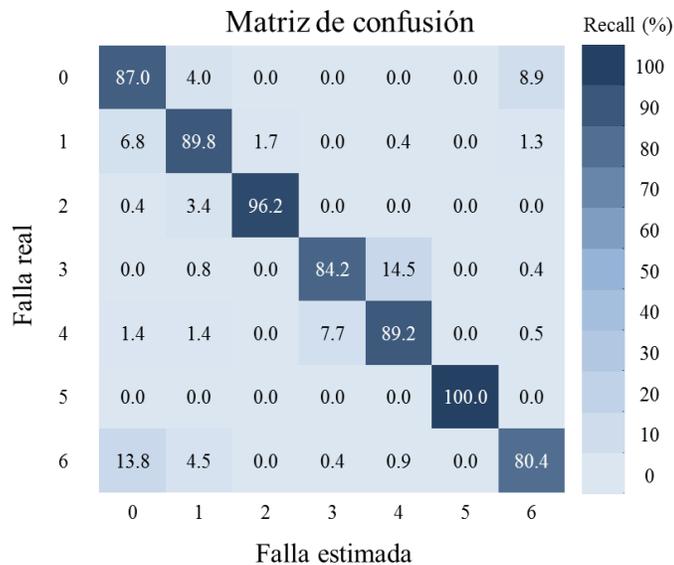


Figura 41: Matriz de confusión sin validación cruzada y con reducción de características

En la Figura 42 se muestra la gráfica de curvas ROC, la cual representa la distribución de la tasa de verdaderos positivos (TPR) conocida como sensibilidad en el eje y, frente a la tasa de falsos positivos (FPR) conocida como (1- especificidad) en el eje x. El trazo de una curva ROC nos indica la capacidad de discriminar una

falla del resto de las demás, este trazo debe de estar lo más cerca posible del lado izquierdo y superior de la gráfica. La medición del área bajo la curva (AUC) indica que el modelo puede clasificar a la gran mayoría de Patrones de Falla de manera correcta. La línea diagonal negra es una línea de referencia o de no-discriminación de fallas.

Para modo de ejemplo observamos a la clase 5 (línea amarilla) donde el eje y corresponde a la probabilidad de ser clasificada correctamente como F5 y el eje x, es la clasificación identificada incorrectamente sobre el resto de las de más fallas reales (TP).

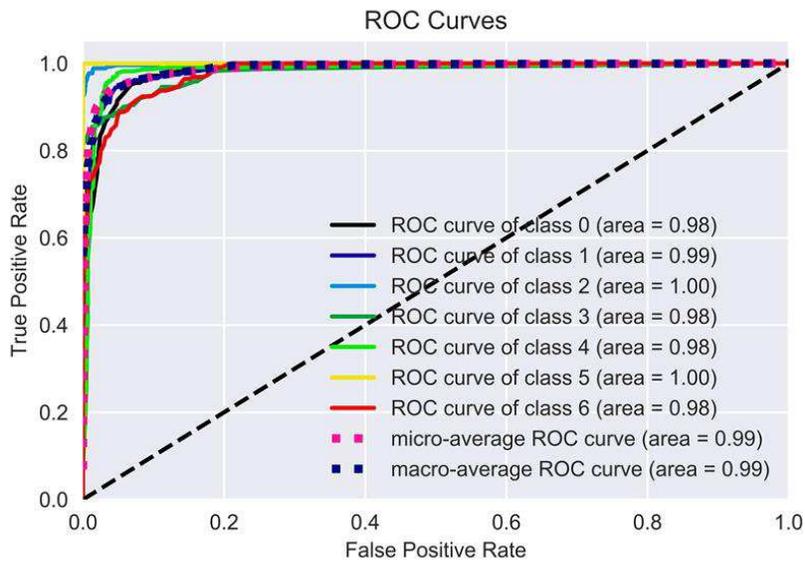


Figura 42: Curvas ROC sin validación cruzada y con reducción de características

5.7.3.4 Experimento 3: RF sin validación cruzada y PCA con dos componentes

Para este tercer experimento dividimos el conjunto de patrones de falla en 2 sub-conjuntos, el primero con el 80% para entrenamiento y el segundo con el 20 % para probar. Para este experimento se utiliza la técnica de análisis de componentes

principales (PCA) aplicado al conjunto de patrones de falla sin discriminar ninguna variable.

El PCA es una técnica para reducir la dimensionalidad de grandes conjuntos de datos, incrementando la capacidad de interpretación, pero al mismo tiempo minimizando la pérdida de información. Lo hace creando nuevas variables no correlacionadas llamadas componentes que maximizan sucesivamente la varianza, para encontrar tales variables se reduce a resolver un problema de eigen-valores eigen-vectores, por lo tanto las nuevas variables están definidas por el conjunto de datos analizados (Ian T. y Cadima 2016).

Esta técnica es útil para visualizar datos de múltiples dimensiones, dado a que el conjunto de patrones de fallas tiene un espacio de alta dimensión que hace imposible visualizar el comportamiento de los datos. Aplicamos esta técnica para proyectar los datos a un espacio de dos dimensiones para comprender el comportamiento de las fallas. En la Figura 43 se grafica los valores de dos componentes principales que representan el conjunto de patrones de falla. Podemos observar claramente el comportamiento de las 7 Fallas involucradas donde la falla 5 no presenta un comportamiento de operación similar al resto de las fallas, pero la falla 3 y 4 presentan comportamientos similares, la falla 1 y 2, 0 y 6 también presentan esta relación. Con esto comprobamos que posiblemente los rendimientos de nuestros modelos de clasificación se deban a estas relaciones. Para comprender las relaciones de estos datos podemos contrastarlas con la descripción de cada falla descritas en la Tabla 1.

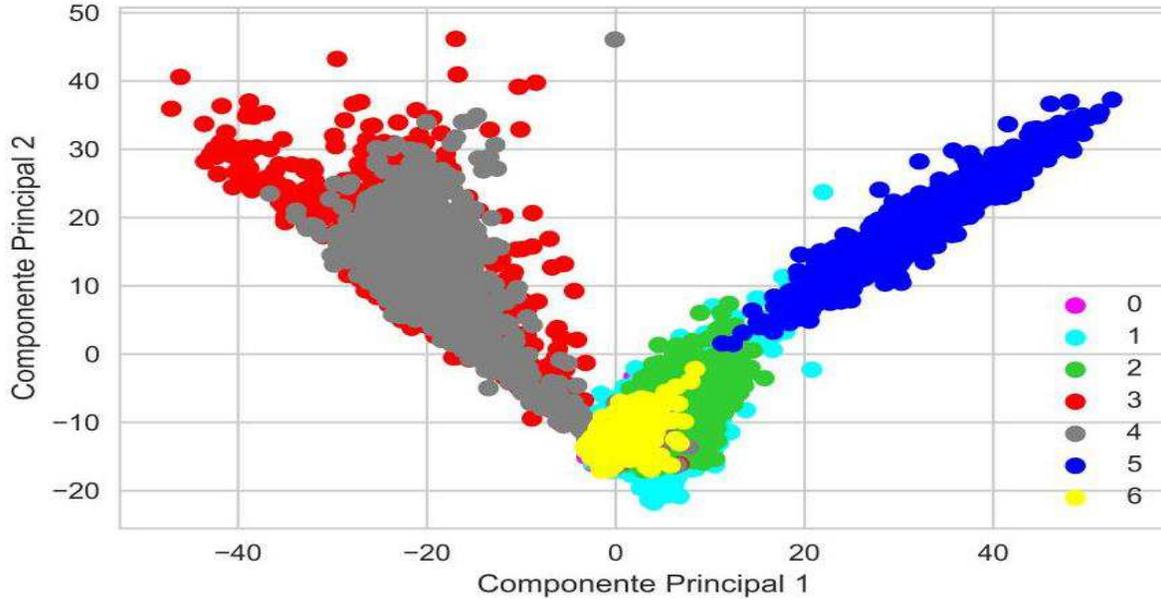


Figura 43: Visualización del conjunto de patrones de falla con PCA

En la Tabla 14 se describe los resultados de éste tercer experimento mediante las métricas de Precisión, Recall, y F1-Score de cada uno de las clases. Se observa que al aplicar PCA al conjunto de patrones de falla, el modelo reporta una exactitud del 60.67%, una Precisión y Recall del 60% lo cual nos indica que el modelo no clasifica las clases de manera correcta. Además, se tiene un F1-score del 60%, el cual también indica que el modelo de clasificación no es preciso ni robusto. A modo de ejemplos se observa en la matriz de confusión de la Figura 44 que la mayoría de las fallas no se podrían clasificar de manera correcta. Por ejemplo, para clasificar la falla 4 el modelo indica un 54.1% de probabilidad de que sea la falla 4 y con un 42.8% de sea la falla 3. Esto prácticamente es un resultado indeterminado, para lo cual se considera como un pésimo clasificador.

- Exactitud de la prueba: 60.67%

Clase	Precisión	Recall	F1-Score
F0	0.49	0.51	0.5
F1	0.45	0.4	0.43
F2	0.67	0.7	0.68
F3	0.6	0.6	0.6
F4	0.55	0.54	0.55
F5	0.99	0.99	1
F6	0.44	0.46	0.45
Promedio	60%	60%	60%

Tabla 14: Reporte de clasificación. 2 PCA sin validación cruzada y sin reducción de características

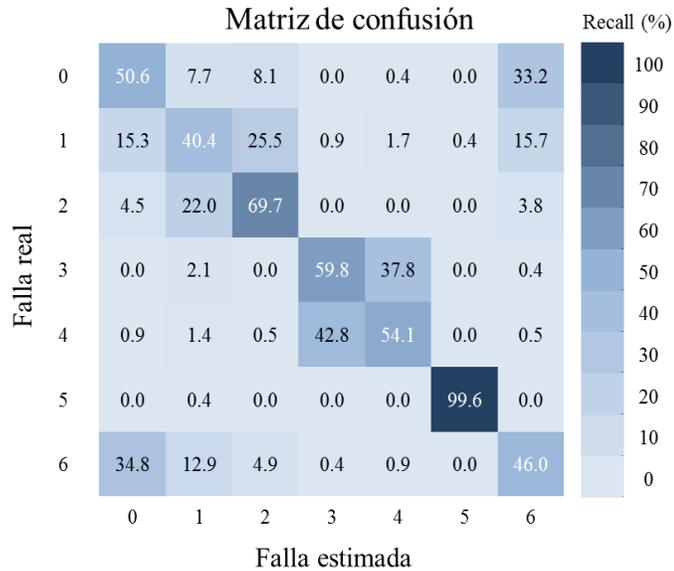


Figura 44: Matriz de confusión. 2 PCA sin validación cruzada y sin reducción de características

Debido a que el modelo construido con dos componentes principales no fue exitoso y tomando en cuenta que el objetivo del análisis de componentes es maximizar sucesivamente la varianza, podemos seleccionar una cantidad de componentes necesarios que representen la mayor cantidad de información y así

evitar perderla. En la Figura 45 se observa que la mayor información de los datos se captura en los primeros 50 componentes donde se presenta la mayor variación de los datos. En el siguiente experimento se comprobará el rendimiento del modelo entrenado con esta cantidad de componentes.

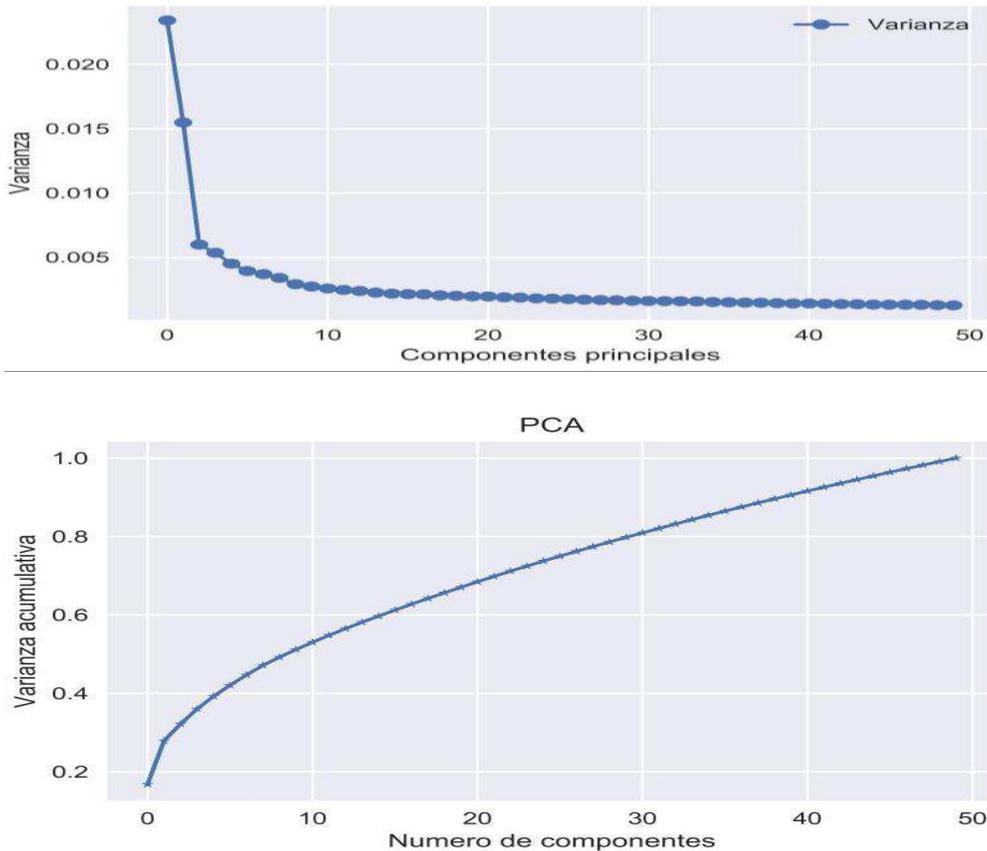


Figura 45: Varianza acumulativa. El eje x, representa número de componentes y el eje y, la varianza a acumulativa

5.7.3.5 Experimento 4: RF sin validación cruzada y PCA con cincuenta componentes

Para este cuarto experimento dividimos el conjunto de patrones de falla en 2 sub-conjuntos, el primero con el 80% para entrenamiento y el segundo con el 20 %

para probar. Para este experimento se generó un modelo con 50 componentes principales, el resultado del rendimiento se describen en la Tabla 15. Se observa que el modelo obtuvo una exactitud y precisión del 81 %, mucho mayor a utilizar solo dos componentes. Teniendo en cuenta que tenemos medidas de precisión y f1-score relativamente aceptables es necesario analizar la matriz de confusión (ver Figura 46) en la cual se describe el rendimiento de clasificación de cada una de las fallas. Por ejemplo, la probabilidad de clasificar correctamente la falla 1, 2, 3, 4, 5 están por arriba del 80 % pero para la falla 6 y 0 sigue siendo un poco ineficiente. Esto podría llegar a tener solución, ahora que ya se conoce el rendimiento del modelo podemos configurar hiper-parámetros del algoritmo RF, por ejemplo, aumentar el número de árboles en el bosque hasta encontrar un rendimiento mucho mejor al ya obtenido.

- Exactitud de la prueba: 81.37%

Clase	Precisión	Recall	F1-Score
F0	0.66	0.65	0.66
F1	0.77	0.78	0.77
F2	0.94	0.91	0.93
F3	0.89	0.81	0.85
F4	0.8	0.9	0.84
F5	0.96	1	0.98
F6	0.62	0.61	0.61
Promedio	81%	81%	81%

Tabla 15: Reporte de clasificación. 50 PCA sin validación cruzada y sin reducción de características

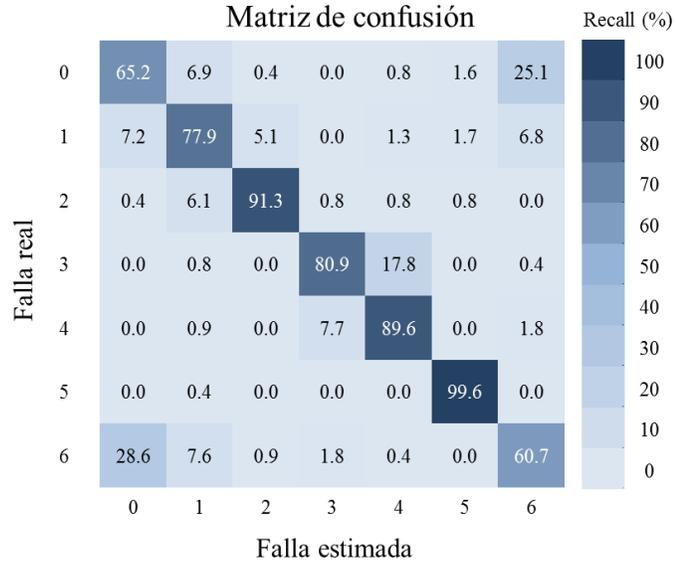


Figura 46: Matriz de confusión. 50 PCA sin validación cruzada y sin reducción de características

5.7.3.6 Experimento 5: RF con validación cruzada y reducción de características

Para este quinto experimento se utilizó el conjunto de patrones de falla conformado por un conjunto de características importantes. Para evaluar el desempeño del modelo en términos de precisión de sus predicciones cuando se tenga nuevos patrones que no hayan sido vistos previamente y obtener un modelo más robusto, se utilizó la técnica de validación cruzada.

De acuerdo con los resultados expuestos en la Tabla 16 se demuestra que el modelo de clasificación generado por algoritmo RF presenta un rendimiento ideal para clasificar distintas fallas simuladas en la MEM, con el 91.14% de exactitud y un 91 % de precisión del modelo generado. Se comprueba que tendrá un buen rendimiento en relación con las nuevas fallas sin analizar que no hayan sido usadas para el entrenamiento del modelo de clasificación de fallas múltiples.

- Exactitud de la prueba: 91.14%

Clase	Precisión	Recall	F1-Score
F0	0.82	0.85	0.83
F1	0.83	0.91	0.87
F2	0.98	0.97	0.97
F3	1	0.87	0.93
F4	0.91	0.96	0.93
F5	1	1	1
F6	0.86	0.83	0.84
Promedio	91%	90%	91%

Tabla 16: Reporte de clasificación con validación cruzada y reducción de características

En la Figura 47 se describe en términos de porcentaje el resultado de la clasificación de cada falla raíz evaluada. Por ejemplo, para F1 (falla real) el modelo clasifica la falla como F1 (falla estimada) con el 90.2%, con el 6.5% lo clasifica como F0, con el 1.6% lo clasifica como F2 y con el 1.7% lo clasifica como F6. La variación de porcentaje con el cual se clasifican las fallas se debe en gran parte al comportamiento de operación similar que existe entre ellas. En estos experimentos podemos observar que la F5 fue clasificada con el 100% debido a que es una falla que no presenta un comportamiento de operación similar con el resto de las fallas, por tratarse de un cortocircuito en el generador. Se supone que el modelo tiene un buen rendimiento de clasificación dado a que la precisión y el Recall son significativamente altos. Además, se tiene una probabilidad del 90 % de que el modelo clasifique correctamente una falla dada. La medida de F1-Sscore del 91% nos indica que tan de preciso y robusto es el modelo de clasificación.

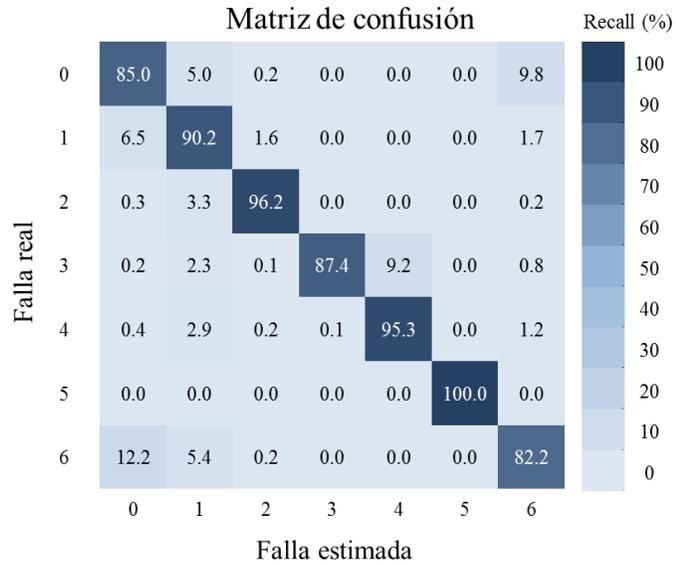


Figura 47: Matriz de confusión con validación cruzada y reducción de características

Conociendo la cantidad de casos presentes en la matriz de confusión podemos calcular distintas métricas de evaluación mencionadas en el capítulo anterior. La Figura 48 describe el rendimiento de clasificación de múltiples fallas, en el eje de las x se representa las fallas estimadas por el modelo y en el eje de las y representa las fallas reales que se sometieron a evaluación.

Donde:

tp (verdaderos positivos): Dado una Falla real etiquetada como F5, se define como la cantidad de casos en que el modelo lo estima como F5.

fn (falsos negativos): Dado una Falla real etiquetada como F6, se define como la cantidad de casos en que el modelo lo clasifica incorrectamente diciendo que puede ser F0, F1, F2.

fp (Falsos positivos): Cuando no hay falla (F0), el modelo de acuerdo a la cantidad de casos analizados lo clasifica como falla, diciendo que puede ser F1, F2, F3, F5, F6.

tn (verdaderos negativos): Predicción rechazada correctamente para la Falla real etiquetada como F0.

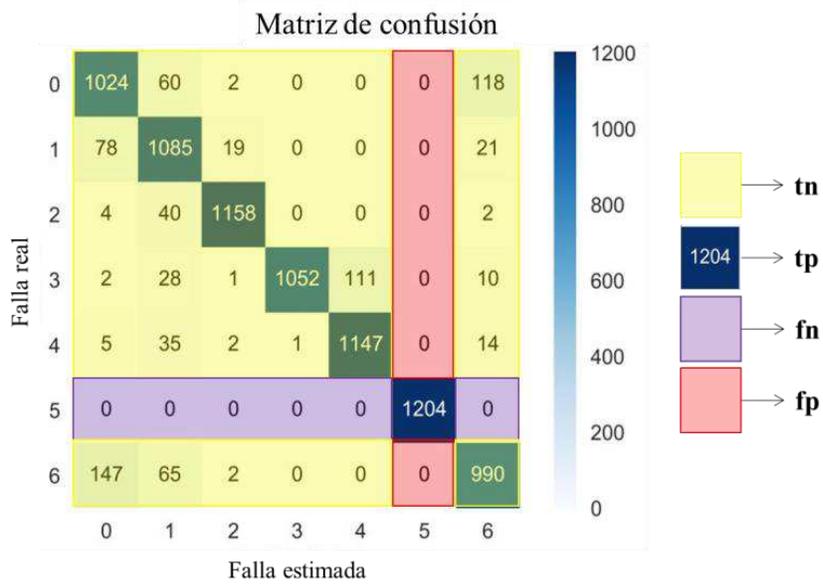


Figura 48: Matriz de confusión del experimento 5

Capítulo 6

Conclusiones y trabajo futuro

Este trabajo de investigación es resultado de las actividades en el proyecto P12 del CEMIE-Eólico. La idea fue utilizar técnicas de Inteligencia Artificial (IA) en apoyo al sector eólico. En este trabajo se utilizó un sistema de diagnóstico de aerogeneradores utilizando modelos probabilistas de comportamiento para poder detectar desviaciones a ese comportamiento normal y se desarrolló una técnica para la identificación de fallas al detectar desviaciones al comportamiento. La identificación de la falla se hace con un procedimiento de reconocimiento de patrones.

Se presenta una metodología para la creación de modelos de comportamiento en operación normal, así como el procedimiento de extracción de datos simulados de operación normal y con falla del aerogenerador.

Se estableció un catálogo de fallas simuladas de la MEM con ayuda de la herramienta computacional Focus6. Se generó un procedimiento de reconocimiento de patrones para tratar los patrones de comportamiento generado por el DxAG así también se definió la forma de crear objetos para formar el conjunto de patrones de fallas los cuales fueron sometidos a técnicas de aprendizaje automático para la clasificación de fallas mediante algoritmos de clasificación.

Se evaluaron 5 algoritmos para determinar el mejor calificador para la identificación de la falla raíz. Como resultado se eligió el algoritmo de clasificación *Random Forest* para construir el modelo de clasificación de fallas múltiples

Como resultado final, se generó un modelo de comportamiento normal de la MEM y a través del sistema de diagnóstico DxAG se generaron los patrones de comportamiento anormal. Con base en lo anterior, se realizaron experimentos para la

clasificación multiclase de patrones de fallas usando el algoritmo de *Random Forest*. Finalmente, se evaluó el desempeño del algoritmo usando métricas de exactitud, precisión, Recall y f1-score logrando un 91% en la clasificación de patrones para identificar la falla raíz.

La aportación del trabajo de investigación es la clasificación de patrones para identificar falla raíz por lo que, dado las dimensiones del trabajo se contempla como trabajo futuro tres tareas importantes. Primero, realizar una comparación más exhaustiva de los diferentes algoritmos de clasificación tratando de mejorar los resultados o haciéndolos más eficientes. Segundo, implementar la integración del procedimiento de reconocimiento de patrones al sistema DxAG con la finalidad de presentarle al operador la clasificación de fallas del aerogenerador, esto mediante un conjunto de pruebas en laboratorio con diferentes fallas y en diferentes condiciones. Finalmente, probar el prototipo experimental con aerogeneradores reales donde se espera una serie de retos a resolver.

Bibliografía

AMDEE. Septiembre 20, 2017. <http://www.amdee.org> (accessed Mayo 30, 2019).

AMDEE. www.amdee.org. Noviembre 2016.

Andersen, S. K., K. G. Olesen, F. V. Jensen, and F. Jensen. "Hugin – a shell for building Bayesian belief universes for expert systems." *In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989: 1080-1085.

Arthur L., Samuel. "Some Studies in Machine Learning Using the Game of Checkers." *IBM Journal of Research and Development*, 1959: 210-229.

Bayes, Thomas, and Mr. Price. "An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S." *Philosophical Transactions of the Royal Society of London*, 1763: 370-418.

Bewick, Viv, Liz Cheek, and Jonathan Ball. "Statistics review 13: Receiver operating characteristic curves." *Journal of Biomedical Science*, November 04, 2004.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

Bishop, Christopher.M. *Neural Network for Pattern Recognition*. Clarendon press. OXFORD, 1995.

CEMIE-Eólico. 05 18, 2018. <http://www.cemieeolico.org.mx/> (accessed 07 30, 2019).

Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern Classification (Second edition)*. New York: Wiley-Interscience, 2000.

Fernández Delgado, Manuel, Eva Cernadas, Senén Barro, and Dinani Amorim. "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research*, 2014: 3133-3181.

- García Márquez, Fausto Pedro, Andrew MarkTobias, Jesús María Pinar Pérez, and Mayorkinos Papaelias. "Condition monitoring of wind turbines: Techniques and methods." *ELSEVIER*, 2012: 169-178.
- García, Uriel A., Pablo H. Ibargüengoytia, Alberto Reyes, and Mónica Borunda. "Modelo de Comportamiento de una turbina eólica." *Congreso Mexicano de Inteligencia Artificial*, 05 18, 2016: 119–129.
- Gómez Gil, Pilar. *El reconocimiento de patrones y su aplicación a las señales digitales*. amexcomp, 2018.
- González, Galarza, et al. "Desarrollo de la Máquina Eólica Mexica en el IIE: proyecto MEM." *Artículo técnico-Boletín IIE*, junio 2013.
- Hossin, H., and M.N. Sulaiman. "A review on evaluation metrics for data classification evaluations." *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, Marzo 2015.
- Ian H., Witten, and Frank Eibe. *Data Mining: Practical Machine Learning Tools and Techniques*. USA: Morgan Kaufmann, 2005.
- Ian T., Jolliffe, and Jorge Cadima. "Principal component analysis: a review and recent developments." *Mathematical, Physical and Engineering Sciences*, April 13, 2016.
- Ibargüengoytia, Pablo H., Sunil Vadera, and L. Enrique Sucar. "A Probabilistic Model for Information and Sensor Validation." *The Computer Journal*, 2006: 113-126.
- Ibargüengoytia, Pablo H., Uriel A. García, Alberto Reyes, and Mónica Borunda. "Anomalies Detection in the Behavior of Processes Using the Sensor Validation Theory." *IBERAMIA 2016*. San Jose, Costa Rica: Springer, 2016. 14-24.
- INEEL. *INEEL*. 12 21, 2016. <https://www.gob.mx/ineel> (accessed 05 30, 2019).

- International Standard IEC 61400-1*. International Electrotechnical Commission, 2007.
- Johnson, K.E., L.Y. Pao, M.J. Balas, and L.J. Fingersh. "Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture." *IEEE Control Systems Magazine*, Mayo 30, 2006: 70-81.
- Komaihaltec Inc.* 03 31, 2015. <http://komaihaltec.co.jp/english/>.
- Lindenburg, C. "PHATAS Release and USER'S MANUAL." 2003.
- Lopez, Elia E., Uriel A. Garcia, Pablo H. Ibargüengoytia, and Luis E. Serrano. "Evaluation of Artificial Intelligence Techniques for Vibration Analysis of Turbogenerators." *Congreso Internacional en Innovación y Desarrollo Tecnológico, CIINDET 2016*. Cuernavaca, Mor.: IEEE, 2016.
- Marques de Sá, J.p. *Pattern Recognition Concepts, Methods and applications*. Portugal: Springer, 2001.
- Md Raihan, Mallick , and Imtiaz Syed A. "A Hybrid Method for Process Fault Detection and Diagnosis." *10th IFAC International Symposium on Dynamics and Control of Process Systems*. Mumbai, India: IFAC, 2013.
- Meik, Schlechtingen, ilmar Ferreira Santos, and Sofiane Achiche. "Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description." *ELSEVIER*, 2013: 259-270.
- Nitin , Hardeniya, Perkins Jacob , Chopra Deepti , Joshi Nisheeth, and Mathur Iti . *Natural Language Processing: Python and NLTK*. birmingham: Packt Publishing, 2016.
- Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems: networks of plausible inferenc*. Palo Alto, Calif: Morgan Kaufmann, 1988.
- Peter, Flach. *MACHINE LEARNING: The Art and Science of Algorithms that Make Sense of Data*. New York: CAMBRIDGE, 2012.

- Poolsawad, N., C. Kambhampati, and J.G.F. Cleland. "Balancing Class for Performance of." *Proceedings of the World Congress on Engineering*, July 2, 2014.
- POWERS, D.M.W. "Evaluation: From Precision, Recall and F-Measure to Roc, informedness, markedness & correlation." *Journal of Machine Learning Technologies* (Bioinfo Publications), 2011: 37-63.
- QUINLAN , J.R. "Induction of Decision Trees." *Machine Learning 1*. Boston: Kluwer Academic Publishers, 1986. 81-106.
- Russell, Stuart, and Peter Norving. *Artificial Intelligence A Modern Approach (Third Edition)*. New Jersey: PEARSON, 2010.
- Shannon, C. E. "A Mathematical Theory of Communication." *The Bell System Technical Journal*, 1948: 379-592.
- Spirtes, P., C. Glymour, and R. Sheines. *Causation, Prediction and Search*. Cambridge, Massachusetts: MIT Press, 2000.
- Steven, Bird, Klein Ewan, and Loper Edward. *Natural Language Processing With Python First Edition*. United States of America: Safari, 2009.
- Tom M., Mitchell. *Machine Learning*. Ithaca, NY: McGraw-Hill, 1998.
- Wind Turbine*. Febrero 19, 2019. <https://www.windturbin.com/>.
- WMC. 01 26, 2010. <https://wmc.eu/focus6.php> (accessed 03 01, 2018).
- Zhou, Anmei, Dejie Yu, and Wenyi Zhang. "A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA." *ELSEVIER*, 2015: 115-125.

Anexo

Nombre y descripción de las variables generadas por el simulador Focus6

ID	Variabes (Señales)	Descripción
1	WindSpeedHub_2	Velocidad del viento en el cubo
2	AerodynamicPower_3	Potencia aerodinámica extraída del aire
3	AxialAerodynamicForceRotor_4	Fuerza axial aerodinámica en el rotor
4	GeneratorPower_5	Potencia del generador (después de restar las pérdidas)
5	WindDirectionHub_6	Dirección del viento en el cubo
6	TowerTopAccelerationInX_7	Aceleración flexionante de la torre en la dirección x
7	TowerTopAccelerationInY_8	Aceleración flexionante de la torre en la dirección y
8	YawAngle_9	Ángulo de orientación con respecto a la punta de la torre
9	YawRate_10	Velocidad del mecanismo de orientación
10	RotorAzimuth_11	Potencia del generador, después de restar las pérdidas
11	RotorSpeed_12	Velocidad angular del rotor
12	TorsionalDeformationOfRotorShaft_15	Deformación torsional de la flecha del rotor
13	FirstTimeDerivativeOf15_16	Aceleración de la deformación de la flecha del rotor
14	RotorAngularAcceleration_17	Aceleración angular del rotor
15	SumYawAngTowerTors_18	Suma del ángulo de orientación y la torsión en la punta de la torre
16	SumYawRateAndRateTowerTors_19	Suma de la velocidad del sistema de orientación y la velocidad de torsión de la torre
17	ComponentinRotorShaftDirectionTowerTop_20	Componente en la dirección de la flecha del rotor de la velocidad de deformación de la punta de la torre
18	BendingMomentInTowerAboutX_21	Momento flexionante en la torre con respecto al eje inercial x
19	BendingMomentInTowerAboutY_22	Momento flexionante en la torre con respecto al eje inercial y
20	TorsionalMomentintheTower_23	Momento torsional (del mecanismo de orientación) en la torre
21	ShearForceinTowerinX_24	Fuerza cortante en la torre en la dirección inercial x
22	ShearForceinTowerinY_25	Fuerza cortante en la torre en la dirección inercial y
23	NormalForceTower_26	Fuerza normal (tensión) en la torre
24	DisplacementofNacelleinX_27	Desplazamiento de la góndola en la dirección inercial x
25	DisplacementofNacelleinY_28	Desplazamiento de la góndola en la dirección inercial y
26	TorsionalDeformationTower_29	Deformación torsional de la torre

ID	Variabes (Señales)	Descripción
27	SidewaysRollingMomentTower_41	Momento lateral en la torre a lo largo de la componente horizontal del vector de rotación
28	BackwardTiltingMomentTower_42	Momento frontal en la torre
29	ShearForceTower_44	Fuerza cortante en la torre a lo largo de la componente horizontal del vector de rotación
30	ShearForceTowerPerpendicularRotor_45	Fuerza cortante en la torre, perpendicular al eje de rotación
31	NormalTensileForceTower_46	Fuerza de tensión normal en la torre
32	DisplacementNacelle_47	Desplazamiento de la góndola (dirección igual a 44)
33	DisplacementNacelle_48	Desplazamiento de la góndola (dirección igual a 45)
34	TorsionalDeformationTower_49	Deformación torsional de la torre
35	TorqueDriveTrainSupport_51	Torque en los soportes del tren de potencia (sin rotación)
36	TiltingMomentDriveTrain_52	Momento frontal del rotor, con respecto al eje y-GL (sin rotación)
37	YawingMomentDriveTrain_53	Momento torsional en el tren de potencia (sin rotación)
38	AxialForceDriveTrain_54	Fuerza axial (de compresión) en el tren de potencia (sin rotación)
39	LateralForceDriveTrain_55	Fuerza lateral (en la dirección y) en el tren de potencia (sin rotación)
40	VerticalShearForceDriveTrain_56	Fuerza cortante vertical en el tren de potencia (sin rotación)
41	TorsionalDeformationFlexibleGearboxSupport_57	Deformación torsional de los soportes flexibles de la caja de engranes (sin rotación)
42	TorqueRotorShaft_61	Torque en la flecha del rotor
43	TiltingMomentRotoraboutY_62	Momento frontal en el rotor, con respecto al eje y-GL (con rotación)
44	YawingMomentShaft_63	Momento torsional en la flecha (con rotación)
45	AxialForceShaft_64	Fuerza axial (de compresión) en la flecha
46	LateralForceShaft_65	Fuerza lateral (en la dirección y) en la flecha (con rotación)
47	VerticalShearForceShaft_66	Fuerza cortante vertical (hacia arriba) en la flecha (con rotación)
48	TorsionalDeformationFlexibleRotorShaft_67	Deformación torsional de la flecha flexible del rotor
49	ControllerInputValueRotorSpeed_91	Valores de entrada del controlador de la velocidad de rotación
50	DemandedPitchRatePitchActuator_92	Velocidad del ángulo de paso demandado por el actuador
51	DemandedTorqueGenerator_93	Torque demandado por el generador
52	LeadMomentBladeRoot3_94	Momento en la raíz del aspa 3, en el plano de rotación Mx (lead momento)

ID	Variabes (Señales)	Descripción
53	FlapMomentBladeRoot3_95	Momento en la raíz del aspa 3, perpendicular al plano/ <i>rot. My (flap moment)</i>
54	LeadMomentBladeRoot2_96	Momento en la raíz del aspa 2, en el plano de rotación Mx (<i>lead moment</i>)
55	FlapMomentBladeRoot2_97	Momento en la raíz del aspa 2, perpendicular al plano/ <i>rot. My (flap moment)</i>
56	LeadMomentBladeRoot1_98	Momento en la raíz del aspa 1, en el plano de rotación Mx (<i>lead moment</i>)
57	FlapMomentBladeRoot1_99	Momento en la raíz del aspa 1, perpendicular al plano/ <i>rot. My (flap moment)</i>
58	Blade1PitchAngle_104	Ángulo de pitch
59	Blade2PitchAngle_204	Ángulo de pitch
60	Blade3PitchAngle_304	Ángulo de pitch
61	Blade1PitchRate_105	Velocidad del pitch
62	Blade2PitchRate_205	Velocidad del pitch
63	Blade3PitchRate_305	Velocidad del pitch
64	Blade1LeadwiseBendingMoment_131	Momento en el aspa en el plano de rotación, con respecto al plano de rotación indeformable (<i>lead moment</i>)
65	Blade2LeadwiseBendingMoment_231	Momento en el aspa en el plano de rotación, con respecto al plano de rotación indeformable (<i>lead moment</i>)
66	Blade3LeadwiseBendingMoment_331	Momento en el aspa en el plano de rotación, con respecto al plano de rotación indeformable (<i>lead moment</i>)
67	Blade1TorsionalMoment_132	Momento torsional en el aspa, con respecto al plano de rotación indeformable
68	Blade2TorsionalMoment_232	Momento torsional en el aspa, con respecto al plano de rotación indeformable
69	Blade3TorsionalMoment_332	Momento torsional en el aspa, con respecto al plano de rotación indeformable
70	Blade1FlapwiseBendingMoment_133	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>), con respecto al plano de rotación indeformable
71	Blade2FlapwiseBendingMoment_233	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>), con respecto al plano de rotación indeformable
72	Blade3FlapwiseBendingMoment_333	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>), con respecto al plano de rotación indeformable
73	Blade1FlapwiseShearForce_134	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>), con respecto al plano de rotación indeformable

ID	Variables (Señales)	Descripción
74	Blade2FlapwiseShearForce_234	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>), con respecto al plano de rotación indeformable
75	Blade3FlapwiseShearForce_334	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>), con respecto al plano de rotación indeformable
76	Blade1SpanwiseForce_135	Fuerza a lo largo del aspa (<i>spanwise force</i>), con respecto al plano de rotación indeformable
77	Blade2SpanwiseForce_235	Fuerza a lo largo del aspa (<i>spanwise force</i>), con respecto al plano de rotación indeformable
78	Blade3SpanwiseForce_335	Fuerza a lo largo del aspa (<i>spanwise force</i>), con respecto al plano de rotación indeformable
79	Blade1LeadwiseShearForce_136	Fuerza cortante en el aspa en el plano de rotación (<i>leadwise shear force</i>), con respecto al plano de rotación indeformable
80	Blade2LeadwiseShearForce_236	Fuerza cortante en el aspa en el plano de rotación (<i>leadwise shear force</i>), con respecto al plano de rotación indeformable
81	Blade3LeadwiseShearForce_336	Fuerza cortante en el aspa en el plano de rotación (<i>leadwise shear force</i>), con respecto al plano de rotación indeformable
82	Blade1FlapwiseDeformation_137	Desplazamiento del aspa en el sentido perpendicular al plano de rotación (<i>flap displacement</i>), con respecto al plano de rotación indeformable
83	Blade2FlapwiseDeformation_237	Desplazamiento del aspa en el sentido perpendicular al plano de rotación (<i>flap displacement</i>), con respecto al plano de rotación indeformable
84	Blade3FlapwiseDeformation_337	Desplazamiento del aspa en el sentido perpendicular al plano de rotación (<i>flap displacement</i>), con respecto al plano de rotación indeformable
85	Blade1LeadwiseDeformation_139	Desplazamiento del aspa en el plano de rotación (<i>lead displacement</i>), con respecto al plano de rotación indeformable
86	Blade2LeadwiseDeformation_239	Desplazamiento del aspa en el plano de rotación (<i>lead displacement</i>), con respecto al plano de rotación indeformable
87	Blade3LeadwiseDeformation_339	Desplazamiento del aspa en el plano de rotación (<i>lead displacement</i>), con respecto al plano de rotación indeformable
88	Blade1LeadwiseBendingMoment_141	Momento en el aspa en el plano de rotación (<i>lead moment</i>)

ID	Variabes (Señales)	Descripción
89	Blade2LeadwiseBendingMoment_241	Momento en el aspa en el plano de rotación (<i>lead moment</i>)
90	Blade3LeadwiseBendingMoment_341	Momento en el aspa en el plano de rotación (<i>lead moment</i>)
91	Blade1TorsionalMoment_142	Momento torsional en el aspa
92	Blade2TorsionalMoment_242	Momento torsional en el aspa
93	Blade3TorsionalMoment_342	Momento torsional en el aspa
94	Blade1ForwardBendingMoment_143	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>)
95	Blade2ForwardBendingMoment_243	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>)
96	Blade3ForwardBendingMoment_343	Momento flexionante en el aspa en la dirección contraria al viento (<i>upwind bending moment</i>)
97	Blade1FlapwiseShearForce_144	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>)
98	Blade2FlapwiseShearForce_244	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>)
99	Blade3FlapwiseShearForce_344	Fuerza cortante en el aspa en la dirección perpendicular al plano de rotación (<i>flapwise shear force</i>)
100	Blade1AxialForce_145	Fuerza a lo largo del aspa (<i>spanwise force</i>)
101	Blade2AxialForce_245	Fuerza a lo largo del aspa (<i>spanwise force</i>)
102	Blade3AxialForce_345	Fuerza a lo largo del aspa (<i>spanwise force</i>)
103	Blade1LeadwiseShearForce_146	Fuerza cortante en el aspa en dirección del plano de rotación (<i>leadwise shear force</i>)
104	Blade2LeadwiseShearForce_246	Fuerza cortante en el aspa en dirección del plano de rotación (<i>leadwise shear force</i>)
105	Blade3LeadwiseShearForce_346	Fuerza cortante en el aspa en dirección del plano de rotación (<i>leadwise shear force</i>)
106	Blade1FlapwiseDeformation_147	Desplazamiento de la punta del aspa en dirección perpendicular al plano de rotación (<i>flap displacement</i>)
107	Blade2FlapwiseDeformation_247	Desplazamiento de la punta del aspa en dirección perpendicular al plano de rotación (<i>flap displacement</i>)
108	Blade3FlapwiseDeformation_347	Desplazamiento de la punta del aspa en dirección perpendicular al plano de rotación (<i>flap displacement</i>)
109	Blade1LeadwiseDeformation_149	Desplazamiento de la punta del aspa en el plano de rotación (<i>lead displacement</i>)
110	Blade2LeadwiseDeformation_249	Desplazamiento de la punta del aspa en el plano de rotación (<i>lead displacement</i>)
111	Blade3LeadwiseDeformation	Desplazamiento de la punta del aspa en el plano de rotación (<i>lead displacement</i>)



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Cuernavaca, Morelos a 22 de agosto del 2019.

DRA. LAURA PATRICIA CEBALLOS GILES
DIRECTORA DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de Maestro en Optimización y Cómputo Aplicado, del estudiante Uriel Alejandro García López, con matrícula 10010397, con el título **Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones al Comportamiento Normal Usando Modelos Probabilistas** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente

Dr. Pablo Héctor Ibargüengoytia González
Investigador
Instituto Nacional de Electricidad y Energías Limpias



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Cuernavaca, Morelos a 29 de agosto del 2019.

DRA. LAURA PATRICIA CEBALLOS GILES
DIRECTORA DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de Maestro en Optimización y Cómputo Aplicado, del estudiante Uriel Alejandro García López, con matrícula 10010397, con el título **Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones al Comportamiento Normal Usando Modelos Probabilistas** por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente

Dr. Alberto Reyes Ballesteros
Investigador
Instituto Nacional de Electricidad y Energías Limpias



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 30 de agosto del 2019.

DRA. LAURA PATRICIA CEBALLOS GILES
DIRECTORA DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de Maestro en Optimización y Cómputo Aplicado, del estudiante Uriel Alejandro García López, con matrícula 10010397, con el título Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones al Comportamiento Normal Usando Modelos Probabilistas por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.


Dra. Lorena Díaz González
Atentamente
Por una humanidad culta
Una universidad de excelencia

Nombre del docente
Profesor- investigador
Centro de Investigación en Ciencias



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 30 de agosto del 2019.

DRA. LAURA PATRICIA CEBALLOS GILES
DIRECTORA DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de Maestro en Optimización y Cómputo Aplicado, del estudiante Uriel Alejandro García López, con matrícula 10010397, con el título Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones al Comportamiento Normal Usando Modelos Probabilistas por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

Nombre del docente
Profesor- investigador
Centro de Investigación en Ciencias



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

Cuernavaca, Morelos a 30 de agosto del 2019.

DRA. LAURA PATRICIA CEBALLOS GILES
DIRECTORA DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de Maestro en Optimización y Cómputo Aplicado, del estudiante Uriel Alejandro García López, con matrícula 10010397, con el título Identificación de Fallas de Turbinas Eólicas al Detectar Desviaciones al Comportamiento Normal Usando Modelos Probabilistas por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que la estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

Nombre del docente
Profesor- investigador
Facultad de Ciencias Químicas e Ingeniería