

Detección y Diagnostico de fallas en turbinas de viento usando  
técnicas de minería de datos

Rubén Francisco Manrique

Tesis presentada como requisito parcial para optar al título de:  
Máster en Sistemas y Computación

Ph.D Jorge Sofrony Esmeral, Director.

Universidad Nacional de Colombia

Departamento de Sistemas y Computación

Octubre 2012

Bogotá, Colombia



# Resumen

Detección y Diagnostico de fallas en turbinas de viento usando técnicas de minería de datos

Rubén Francisco Manrique

Departamento de Sistemas y Computación

La energía eólica es una de las fuentes de energía alternas más importantes con las que cuenta el planeta. En la presente década se ha evidenciado un gran interés en este tipo de energía lo que ha conducido al desarrollo de turbinas de viento más sofisticadas y de mayor capacidad de generación. Un punto crítico en las turbinas de viento es la implementación de sistemas efectivos de detección de fallas que logren minimizar sus costos de mantenimiento. Según estudios recientes para una turbina con 20 años de vida útil, los costos de mantenimiento y reemplazo de partes se estiman en al menos 10-15% de los ingresos totales de la generación, un costo muy alto si se compara con otros tipos de energía. El desarrollo de sistemas de detección y aislamiento de fallas (FDI) para turbinas de viento es un tema de investigación actual, fundamental para posicionar aún más este tipo de energía.

La mayoría de los sistemas FDI se basan en modelos matemáticos construidos a partir del conocimiento a priori del sistema, estos realizan análisis de residuos para determinar la existencia o no de una falla. La efectividad de un sistema con este enfoque depende directamente de la capacidad del modelo matemático para interpretar la dinámica del sistema, en plantas complejas como las turbinas de viento actuales obtener modelos con la calidad suficiente como para ser empleados en detección de fallas es una tarea compleja. Como alternativa a este enfoque se encuentran los sistemas de detección de fallas basados en datos 'data-driven', los cuales a partir de una colección de datos provenientes de un sistema de monitoreo identifican patrones que le permiten hacer inferencias sobre comportamientos anormales. En un enfoque 'data-driven' disminuye en gran medida la cantidad de información analítica necesaria acerca de la planta, sin embargo son necesarios algoritmos y técnicas que sean capaces de identificar y extraer patrones sobre grandes sets de datos (extracción de conocimiento). En ese sentido el área de minería de datos nos provee de grandes herramientas, algoritmos y metodologías para la construcción de sistemas FDI 'data-driven'.

En este trabajo se implementaron diferentes sistemas FDI sobre modelos simulados de turbinas de viento, todos ellos construidos bajo un proceso KDD (Knowledge Discovery in Databases) y la aplicación de algoritmos de clasificación/regresión. Su evaluación se realizó en términos y métricas de desempeño industrial comúnmente usadas en la literatura del área. Los resultados obtenidos demuestran que el uso de técnicas de minería de datos en sistemas FDI para turbinas de viento es un camino prometedor alternativo a las técnicas tradicionales basadas en modelos.

Keywords: detección y diagnostico de fallas, minería de datos, turbina de vientos, data-driven

# Abstract

Fault Detection and Diagnosis for Wind Turbines using  
Data-Mining Approach

Rubén Francisco Manrique

Wind energy is one of the alternative energy sources more important with the planet counts. In the present decade has increased the interest in this type of energy, which has led to the development of more sophisticated wind turbines and increased generation capacity. A critical point in wind turbines is the implementation of effective fault detection that manage to minimize maintenance costs. According to recent research for a turbine with 20 years of service life, maintenance costs and replacement parts are estimated at least 10-15 % of total revenue generation, a high cost compared with other types of energy. The development of systems fault detection and isolation (FDI) for wind turbines is a current research topic, key to position this type of energy.

Most FDI systems are based on mathematical models constructed from a priori knowledge of the system for residue analysis to determine the existence of a fault. The effectiveness of this approach depends directly on the ability of the mathematical model to 'interpret the dynamics of the system', in complex plants like current wind turbines get models with sufficient quality for use in detection of failures is a complex task. As an alternative to this approach are 'data-driven' fault detection systems, which from a collection of data from a monitoring system, identify patterns that allow you to make inferences about abnormal behavior. In a 'data-driven' approach, it reduces the amount of necessary analytical information about the plant, however algorithms and techniques are needed to identify and extract patterns on large data set (extraction of knowledge). In that sense data mining area provides us with great tools, algorithms and methodologies for building FDI 'data-driven' systems.

In this work different systems were implemented FDI on simulated models of wind turbines, all based on KDD (Knowledge Discovery in Databases) process and the application of algorithms classification / regression. The evaluation was made in terms of industrial performance metrics commonly used in the literature of the area. The results show that the use of data mining techniques in FDI systems for wind turbines is a promising way, alternative to traditional techniques based on models.

Keywords: Fault Detection and Isolation, Data Mining, Wind Turbine, Data-Driven





# Contenido

<b>Tabla de contenido</b>	<b>VII</b>
<b>Lista de Figuras</b>	<b>IX</b>
<b>Lista de Tablas</b>	<b>XI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. El problema de la generación de energía eólica . . . . .	2
1.2. Enfoque de la Investigación . . . . .	3
1.3. Organización de la tesis . . . . .	4
<b>2. Sistemas de Detección y Diagnóstico de Fallas: Estado del Arte</b>	<b>5</b>
2.1. Definiciones . . . . .	5
2.2. Métodos de Detección y Aislamiento . . . . .	7
2.2.1. Métodos Basados en Modelos . . . . .	8
2.2.2. Métodos Libres de Modelos . . . . .	10
<b>3. Minería de datos y su aplicación a sistemas FDI</b>	<b>13</b>
3.1. Minería de Datos . . . . .	13
3.1.1. Clustering . . . . .	15
3.1.2. Reglas de Asociación . . . . .	16
3.2. Clasificación . . . . .	16
3.3. Algoritmos de Clasificación . . . . .	18
3.3.1. K-nearest neighbours (KNN) . . . . .	19
3.3.2. Clasificadores Bayesianos (CB) . . . . .	22
3.3.3. Máquinas de Vectores de Soporte (SVM) . . . . .	24
3.3.4. Redes Neuronales Artificiales (ANN) . . . . .	33
3.4. Minería de datos aplicada al desarrollo de sistemas FDI . . . . .	37

<b>4. Caso de estudio: Turbina de viento</b>	<b>39</b>
4.1. Modelo 1: Benchmark de turbina de viento	40
4.1.1. Descripción general del sistema	40
4.1.2. Estrategia de Control	42
4.1.3. Escenarios de falla	43
4.2. Modelo 2: FAST	47
4.2.1. Características Generales	47
4.2.2. Sensores	48
4.2.3. Actuadores	48
4.2.4. Escenarios de Falla	49
4.3. Retos de los Modelos	50
<b>5. Construcción del sistema FDI mediante clasificadores</b>	<b>51</b>
5.1. Modelo 1: Benchmark	51
5.1.1. Análisis exploratorio de los datos	51
5.1.2. Pre procesamiento de los datos	58
5.1.3. Aplicación de los algoritmos de clasificación	62
5.1.4. Construcción del sistema FDI (Arquitectura a dos niveles)	68
5.1.5. Pruebas experimentales	73
5.2. Modelo 2: FAST	78
5.2.1. Análisis exploratorio de los datos	78
5.2.2. Construcción del sistema FDI (Arquitectura a dos niveles)	83
5.2.3. Pruebas experimentales	84
<b>6. Construcción de un sistema FDI híbrido</b>	<b>87</b>
6.1. Construcción de los modelos de regresión con redes neuronales	89
6.1.1. Selección de entradas/salidas de las redes neuronales	90
6.1.2. Entrenamiento de las redes neuronales.	91
6.2. Generación de Residuos.	94
6.3. Evaluación de Residuos: Construcción de Clasificadores	95
6.4. Pruebas Experimentales	97
<b>7. Conclusiones y Trabajo Futuro</b>	<b>101</b>
7.1. Referentes a los sistemas FDI diseñados	101
7.2. Referente a las Fallas tratadas y las turbinas de viento	102
7.3. Referente a los algoritmos de clasificación	102
7.4. Trabajo Futuro	102
<b>Bibliografía</b>	<b>105</b>



# Lista de Figuras

1.1. Procesos de operación y mantenimiento cuando son empleados sistemas FDI . . .	2
1.2. Esquema general de detección de fallas basado en modelo . . . . .	3
2.1. Clasificación de Fallas de acuerdo a su comportamiento en el tiempo. La Falla ocurre en $t = \tau_f$ . . . . .	6
2.2. Clasificación de los métodos FDI. . . . .	7
3.1. Áreas base de la Minería de Datos . . . . .	14
3.2. El proceso de descubrimiento de conocimiento . . . . .	15
3.3. Clasificación: tarea de asignar a conjunto de atributos de entrada ( $x$ ) una etiqueta de clase ( $y$ ). . . . .	17
3.4. Clasificación de un nuevo ejemplo (estrella) dentro de un conjunto de objetos per- tenecientes a dos clases. . . . .	19
3.5. Los 3 vecinos mas cercanos al nuevo ejemplo en base a la distancia euclidiana. . .	20
3.6. (a) Predicción de la clase de un ejemplo nuevo con respecto al vecino más cercano. (b) Vecinos Cercanos para $K=3$ . (c) Vecinos Cercanos para $K=8$ . . . . .	21
3.7. Aproximación de una función de distribución de probabilidad mediante una suma ponderada de Gaussianas . . . . .	24
3.8. Hiperplano separador en $\mathbb{R}^2$ , para dos clases . . . . .	26

3.9. Ejemplo de una SVM para dos clases linealmente separables . . . . .	27
3.10. SVM, caso no linealmente separable . . . . .	29
3.11. SVM, uso de la función de kernel para transformar a un espacio nuevo $H$ . . . . .	32
3.12. SVM, Clasificación múlticlasa . . . . .	34
3.13. Perceptrón Simple . . . . .	36
3.14. Perceptrón Multicapa . . . . .	37
4.1. Modelo de turbina de viento propuesto en [1]. Vista en Simulink. . . . .	41
4.2. Interconexión del modelo . . . . .	42
4.3. Curva de potencia turbina de viento . . . . .	43
4.4. Falla en los sensores de valor fijo . . . . .	44
4.5. Falla en los sensores de valor escalado . . . . .	45
4.6. Componentes del modelo de turbina simulado en FAST . . . . .	47
4.7. Componentes de la interfaz de simulink construida con FAST . . . . .	48
5.1. Grafica de las características: WindSp, RotorSp M1, RotorSp M2, GenSp M1, GenSp M2, y GenTorq. . . . .	52
5.2. Grafica de las características: GenPower, B11Pitch M1, B11Pitch M2, B12Pitch M1, B12Pitch M2, B13Pitch M1 y B13Pitch M2. . . . .	52
5.3. Histogramas a 10 bins de cada una de las características en el set de datos. . . . .	56
5.4. Estructura de un Diagrama de Caja. . . . .	57
5.5. Graficas de Caja (Identificación de Outliers). . . . .	59
5.6. Outliers en las medidas de los pitch. . . . .	60
5.7. Efecto del Muestreo. . . . .	60
5.8. KNN encontrando el valor de $k$ . . . . .	64
5.9. SVM encontrando el valor de $C$ . . . . .	65

---

5.10. SVM encontrando el valor de $\sigma$ . . . . .	65
5.11. Arquitectura FDI a dos niveles. . . . .	69
5.12. Ejemplo del funcionamiento del FCTW. . . . .	71
5.13. Predicción realizada por SVM+FCTW en el Nivel 1. . . . .	72
5.14. Clasificación realizada por las redes Neuronales en el Nivel 2 . . . . .	74
5.15. Comportamiento de la Falla 7 . . . . .	76
5.16. Grafica de las características: Velocidad del Rotor, Velocidad-Torque del Generador y Potencia Generada (Modelo Fast). . . . .	79
5.17. Grafica de las características: Angulos del Pitch y Error de desplazamiento. . . . .	79
5.18. Modelo de viento usado en FAST . . . . .	80
5.19. Histogramas a 10 bins de cada una de las características en el set de datos (Modelo Fast). . . . .	81
5.20. Graficas de Caja, identificación de outliers en modelo FAST. . . . .	82
6.1. Arquitectura híbrida propuesta . . . . .	88
6.2. Red de base radial (NN-RBF) con conexión completa. $x, \lambda, \hat{y}$ son las entradas, los pesos de las conexiones y la salida respectivamente. . . . .	89
6.3. Comparación entre la salida real y la estimación realizada por las redes neuronales (I). . . . .	92
6.4. Comparación entre la salida real y la estimación realizada por las redes neuronales (II). . . . .	93
6.5. Comportamiento de los residuos $B_2 - r_i$ (grafica inferior) al ser introducida una falla de valor escalado a los 50 Ts de ejecución en la señal $B_2$ (grafica superior). . .	95
6.6. Evaluación de residuos mediante clasificadores. . . . .	96
6.7. Ejemplo de conjunto de residuos de entrada al clasificador. . . . .	98



# Lista de Tablas

3.1. Ejemplo de un conjunto de datos de entrenamiento . . . . .	18
4.1. Energía eólica instalada a nivel mundial (1998-2008) . . . . .	39
4.2. Fallas en los sensores consideradas . . . . .	46
4.3. Fallas en los actuadores consideradas . . . . .	46
4.4. Señales medidas (sensores) seleccionadas en el modelo de turbina FAST . . . . .	49
4.5. Fallas consideradas en el modelo FAST . . . . .	49
5.1. Conjunto inicial de atributos del set de entrenamiento . . . . .	53
5.2. Matriz de Correlación . . . . .	54
5.3. Conjunto de características posterior a la eliminación de redundancia física . . . . .	61
5.4. Resultados de cada arquitectura de red neuronal empleada (CO capas ocultas, Nx Número de Neuronas por Capa, FA Función de Activación, TE Tiempo de Entrenamiento) . . . . .	63
5.5. Evaluación de los clasificadores . . . . .	67
5.6. Evaluación de los clasificadores Nivel 1 . . . . .	70
5.7. Evaluación de los clasificadores Nivel 1 a la salida del FCTW . . . . .	72
5.8. Evaluación de los clasificadores Nivel 1 a la salida del FCTW . . . . .	72
5.9. Evaluación de los clasificadores Nivel 2 . . . . .	73

---

5.10. Sensibilidad a Fallas Nivel 2 . . . . .	73
5.11. Resultados de la Evaluación al sistema FDI . . . . .	75
5.12. Resultados de la Evaluación al sistema FDI . . . . .	77
5.13. Conjunto de atributos del set de entrenamiento, Modelo FAST . . . . .	80
5.14. Evaluación de los clasificadores Nivel 1 . . . . .	83
5.15. Evaluación de los clasificadores Nivel 2 - Sensibilidad a Fallas . . . . .	83
5.16. Evaluación de los clasificadores Nivel 2 - Precisión Global . . . . .	84
5.17. Resultados de la Evaluación al sistema FDI modelo FAST . . . . .	85
6.1. Resumen de las características de las redes neuronales entrenadas . . . . .	94
6.2. Valores de $\Omega$ y $\bar{w}$ utilizados para la generación de residuos . . . . .	95
6.3. Evaluación de los clasificadores construidos para la evaluación de residuos . . . . .	97
6.4. Resultados de las pruebas experimentales aplicadas al sistema FDI híbrido . . . . .	100

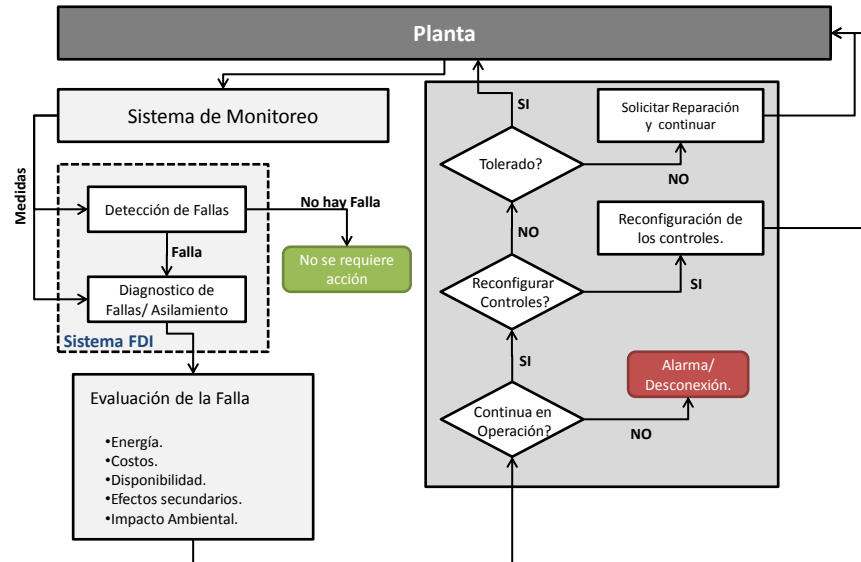
# Capítulo 1

## Introducción

Uno de los mayores problemas que presentan los sistemas industriales es la identificación de la ocurrencia de una falla. En muchos casos su no identificación oportuna puede ser extremadamente costosa en términos de vidas humanas, impacto ambiental y pérdidas económicas. En los últimos años se evidencia una creciente demanda de sistemas industriales de mayor rendimiento, mejor seguridad y fiabilidad, que sean capaces de operar autónomamente y tomar la mejor decisión en presencia de una falla. Son estas razones por la que los sistemas de detección y aislamiento de fallas (FDI 'Fault Detection and Isolation') se han convertido en un componente esencial en el desarrollo de sistemas inteligentes que estén en la capacidad de detectar, aislar (ubicar) y estimar autónomamente la severidad de una o varias fallas.

El proceso de detección es básicamente una decisión binaria, el sistema está funcionando correctamente o se evidencia la ocurrencia de una falla. Por su parte, el aislamiento se enfoca en la determinación del tipo (fuente/elemento) de fallo. Una vez que el fallo ha sido detectado y aislado, se requiere alguna acción. En la Figura 1.1 se muestra el proceso típico de operación y mantenimiento empleando sistemas de detección y aislamiento de fallas: el primer paso lo realiza el sistema de monitoreo sensando físicamente el sistema, los datos suministrados pasan al sistema FDI donde son detectadas condiciones anormales (problemas), se evalúa si es una falla y luego se determina su origen y posibles causas (aislamiento). Un sistema de valoración evalúa el impacto de la falla en términos de energía, costo, efectos secundarios y disponibilidad. Con base en esta información se determina cual va a ser la respuesta ante esa falla.

El desarrollo de sistemas FDI que respondan de manera eficiente es un tema de investigación actual que tiende a involucrar diversas áreas del conocimiento. Aplicaciones recientes han empleado técnicas de inteligencia computacional y metodologías de minería de datos para realizar procesos de 'extracción de conocimiento' sobre colecciones de datos y así encontrar patrones que le permitan identificar y clasificar un conjunto de fallas. Los resultados obtenidos en estas investigaciones son muy prometedores: baja sensibilidad al ruido [2], bajos niveles de no-detección [3,4], y una buena capacidad no solo para detectar la falla si no para identificar su origen [5].



**Figura 1.1** Proceso típico de operación y mantenimiento cuando son empleados sistemas FDI.

## 1.1. El problema de la generación de energía eólica

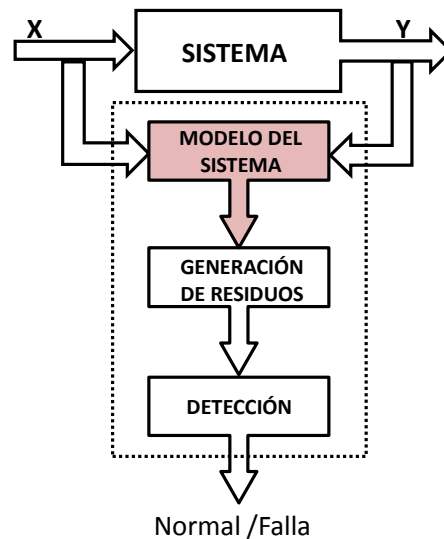
Un problema actual de la humanidad es la generación de energía, sin ella los sistemas sociales y económicos actuales colapsarían. Es bien sabido que las energías fósiles son finitas y dado el actual crecimiento de la demanda mundial se pronostica que en un futuro muy cercano no se sea capaz de abastecerla. Sumado a esto, el uso creciente y continuo de energías fósiles ha causado un gran daño en nuestro medio ambiente, problemas como el calentamiento global y el derretimiento de los polos son consecuencias de su uso.

Ante estos inminentes problemas es necesario integrar otros tipos de energías que sean rentables, accesibles y que no representen ningún daño para el planeta. Dentro de este contexto, la energía eólica se presenta como una de las alternativas con más auge a nivel mundial, en los últimos 10 años ha tenido un crecimiento promedio del 30% convirtiéndose así, en la segunda energía alternativa con más prospectiva [6].

Este gran crecimiento ha aumentado el interés por desarrollar turbinas de viento (aerogeneradores) con una mayor capacidad de generación y con una más larga vida útil. En este sentido uno de los mayores retos que se enfrentan son los elevados costos de mantenimiento los cuales se estiman en al menos un 10-15% de los ingresos totales de la generación [7].

Para suplir esta deficiencia es necesario integrar sistemas que permitieran la detección temprana de fallas. El desarrollo de sistemas FDI que respondan de manera eficiente en aerogeneradores es un tema de investigación actual que tiende a involucrar diversas áreas del conocimiento.





**Figura 1.2** Esquema general de detección de fallas 'basado en modelos'.

## 1.2. Enfoque de la Investigación

Los métodos de detección y aislamiento de fallas se pueden clasificar de manera amplia en dos tipos: los basados en modelos (model based) o los libres de modelos controlados por los datos (data driven model free). En términos generales, los basados en modelos usan el conocimiento a priori del sistema para desarrollar modelos, los cuales son usados como referencia para evaluar diferencias (residuos) entre el estado actual de operación, determinado por las mediciones del sistema de monitoreo, y el estado de operación esperado obtenido del modelo [8, 9], Figura 1.2.

La calidad de un sistema FDI con enfoque basado en modelos depende directamente de la precisión del modelo construido para describir la dinámica del proceso real siendo por lo tanto, potencialmente sensibles a errores de modelado, variación paramétrica, ruido y perturbaciones [10].

Por otro lado, están los métodos controlados por los datos los cuales han ganado gran aceptación en los últimos años debido en gran parte al desarrollo de mejores técnicas de aprendizaje que permiten definir y detectar patrones [11]. Para estos, a diferencia de los anteriores no es necesario especificar un modelo matemático del sistema para los procesos de detección y aislamiento, en su lugar usan colecciones de datos provenientes del sistema de monitoreo para descubrir patrones que permitan caracterizar e identificar fallas en el sistema. Este enfoque emplea diferentes técnicas de minería de datos y algoritmos de aprendizaje maquina: sistemas FDI basados en redes neuronales [12, 13], métodos estadísticos multivariados basados en análisis de componente principal (PCA) [2], redes bayesianas [14], clustering [15, 16], entre otros, han demostrado resultados pro-

medidores en el área superando algunos de los problemas de los métodos basados en modelos [17].

Esta investigación se centra en métodos 'data-driven' que requieran la mínima cantidad de información analítica acerca del sistema. Como caso de estudio se trabaja con modelos simulados de turbinas de viento para los cuales se diseñan diferentes sistemas FDI empleando para ello una metodología propia de minería de datos y diferentes técnicas de inteligencia computacional. Para su construcción se proponen dos arquitecturas novedosas: la primera basada en dos niveles de clasificación que separan la tarea de detección y aislamiento, y la segunda de carácter híbrido que toma las cualidades de los métodos basados en modelos y los combina con un enfoque 'data-driven'.

### **1.3. Organización de la tesis**

Esta tesis está organizada en 7 capítulos. En este capítulo, se discutió la introducción a sistemas FDI y el interés de la investigación. La literatura relacionada a las diferentes metodologías para el desarrollo de sistemas FDI son revisadas en el Capítulo 2, los algoritmos de clasificación a emplearse así como su forma de aplicación en el Capítulo 3. El Capítulo 4 presenta como caso de estudio dos modelos de turbina de viento. En el Capítulo 5 se construye un sistema FDI con enfoque 'data-driven', los resultados obtenidos son evaluados y comparados contra otras investigaciones en el área. En el Capítulo 6 se procede a construir un sistema FDI híbrido que emplea tanto técnicas basadas en el modelo como técnicas data-driven. Finalmente, en el Capítulo 7 se dan conclusiones y recomendaciones para trabajos futuros.

# Capítulo 2

## Sistemas de Detección y Diagnóstico de Fallas: Estado del Arte

El propósito de este capítulo es revisar los métodos mas usados para la detección y aislamiento de fallas, así como las investigaciones más recientes que aplican inteligencia computacional. Se definirán términos y conceptos que serán usados en los siguientes capítulos.

### 2.1. Definiciones

Una falla representa un cambio inesperado en el funcionamiento del sistema, que podrían en ultimas involucrar daño físico o estructural del sistema. De acuerdo a R. Isermann [18], se pueden realizar tres clasificaciones:

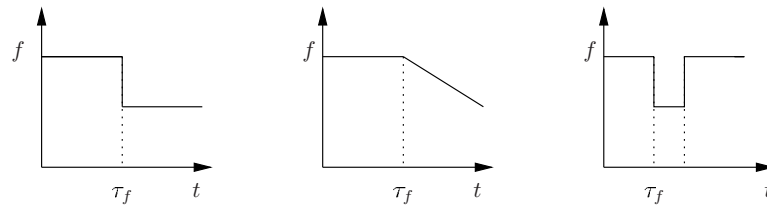
- De acuerdo a su localización física.
- De acuerdo a sus propiedades matemáticas.
- De acuerdo a su comportamiento en el tiempo.

#### **Clasificación de Fallas de acuerdo a su localización física.**

Se pueden encontrar fallas en los sensores, actuadores y componentes. Considerando un proceso físico, representado como un sistema en tiempo discreto LTI (Linear Time Invariant), despreciando las secuencias de ruido, se puede definir como:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.1}$$

$$y(k) = Cx(k) + Du(k) \tag{2.2}$$



**Figura 2.1** Clasificación de Fallas de acuerdo a su comportamiento en el tiempo. De izquierda a derecha: fallas abruptas, incipientes e intermitentes.

Con las matrices de estado  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{l \times n}$  y  $D \in \mathbb{R}^{l \times m}$ , el vector de estados  $x \in \mathbb{R}^n$ , el vector de entrada  $u \in \mathbb{R}^m$  y el vector de salida  $y \in \mathbb{R}^l$ , y  $k \in \mathbb{N}^+$  el instante de tiempo. Una falla en los sensores da como resultado una fallo en el vector de salida  $y(k)$ , según la falla se evidencia en las matrices  $C$  y  $D$ . Una falla en los actuadores da como resultado un fallo en la acción de control, las matrices  $B$  y  $D$  son afectadas. Una falla en los componentes hace referencia a cualquier falla ocurrida en cualquier componente diferente a los sensores o actuadores del sistema y se evidencia como un cambio en la dinámica física del proceso, afectando la matriz  $A$ .

### Clasificación de Fallas de acuerdo a sus propiedades matemáticas.

En este grupo se encuentran las fallas multiplicativas y aditivas. Considerando de nuevo el sistema LTI, un fallo aditivo se puede ver como un término (constante o variable) que se añade al vector de estado  $x(k)$  o el vector de salida  $y(k)$ . En un sentido físico, un fallo aditivo aparece cuando existe un desplazamiento en las medidas de un sensor o en la respuesta de un actuador. Un fallo multiplicativo influye los vectores  $x(k)$  ó  $y(k)$  como el producto de un variable y un término (constante o variable). En un sentido físico, los fallos multiplicativos pueden ser vistos como la degradación de un sensor o actuador. También puede ser utilizados para describir los fallos de componentes: cambios de parámetros dentro del proceso.

### Clasificación de Fallas de acuerdo a su comportamiento en el tiempo.

Se pueden clasificar como: abruptas, incipientes e intermitentes. Una falla incipiente (suave) se desarrolla continua y lentamente sobre el tiempo. Sus efectos sobre el sistema en principio son casi imperceptibles. Un fallo se denomina duro o abrupto si sus efectos sobre el sistema son mayores y pueden llevar el sistema muy cerca del límite de comportamiento aceptable.

Un fallo que se llama intermitente, si sus efectos en el sistema están ocultos por períodos de tiempo discontinuos [18]. A pesar de que este tipo de fallos pueden ser tolerables en el momento en que se producen, deben ser diagnosticados lo antes posible, ya que pueden conducir con el tiempo a graves consecuencias.

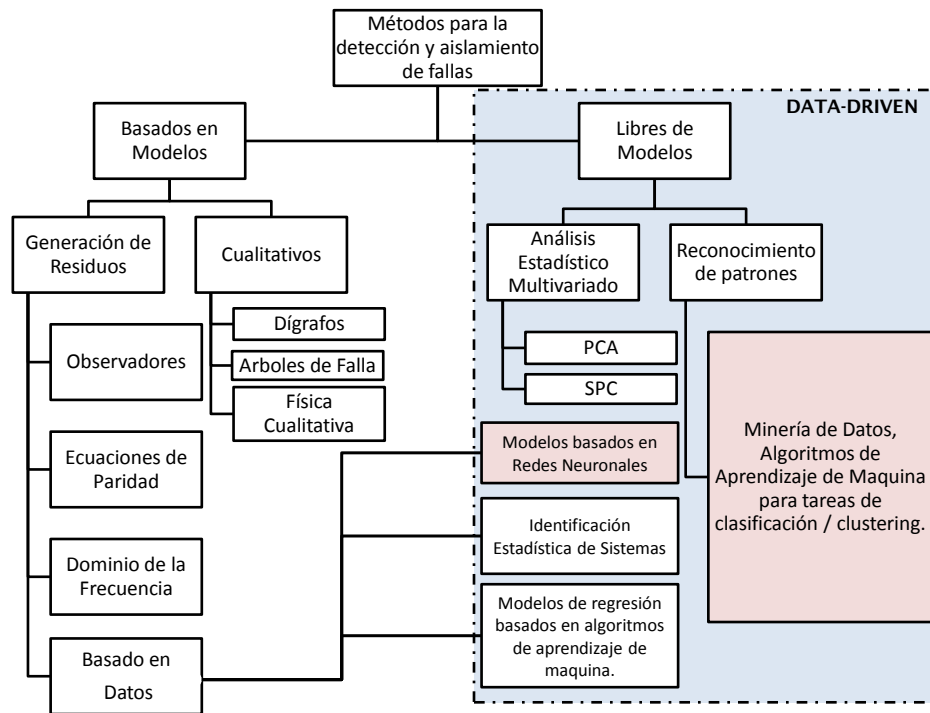


Figura 2.2 Clasificación de los métodos FDI

## 2.2. Métodos de Detección y Aislamiento

Existe en la literatura un gran espectro de métodos FDI que van desde enfoques meramente analíticos, pasando por procesos de identificación estadísticos hasta la inteligencia artificial. Desde el punto de vista de modelamiento y como se menciona en el Capítulo 1 se pueden dividir entre aquellos que necesitan un modelo preciso del sistema y aquellos que solo usan datos históricos del proceso, esta clasificación es general y es comúnmente usada en diferentes estudios realizados en el área [19], [20] y [21]. La Figura 2.2 muestra de forma jerárquica la clasificación de los diferentes métodos de detección y aislamiento de fallas. El foco de esta tesis se centra en los métodos libres de modelo que emplean técnicas y metodologías de minería de datos para el descubrimiento de patrones, sin embargo con el objeto de construir un modelo híbrido también se emplearon modelos de regresión basados en redes neuronales (recuadros en fondo rojo Figura 2.2).

En los métodos basados en modelos la idea clave yace en que es posible construir un modelo que refleje la estructura y las características del sistema con una alta exactitud. La estructura básica de muchos sistemas de detección basados en modelos es en paralelo, esto es tanto el sistema como el modelo recibiendo las mismas entradas y señales de control en los mismos instantes de tiempo, de esta forma los comportamientos inesperados que se presenten en el sistema real producto de una(s) falla(s) se puedan identificar mediante la comparación con el comportamiento predicho por

el modelo. Los enfoques libres de modelo se basan principalmente en las observaciones (datos históricos) que permiten la definición de una condición/señal normal por esta razón es común en la literatura encontrarlos como métodos 'data-driven', sin embargo como se muestra en la Figura 2.2 una nueva generación de métodos basados en modelos también caen en esta clasificación. Los métodos data-driven han ganado gran aceptación en los últimos años, debido a dos razones fundamentales:

1. La disponibilidad de grandes sets de datos históricos correspondientes al funcionamiento del sistema. Los procesos industriales modernos cuentan con avanzados sistemas de monitoreo y adquisición de datos.
2. La integración de efectivas técnicas de identificación de patrones empleadas en áreas como la minería de datos y el aprendizaje de máquina.

El siguiente apartado ofrece una revisión general de la clasificación realizada. En el capítulo siguiente se centrará en la minería de datos y algunas de las técnicas empleadas en esta tesis.

### 2.2.1. Métodos Basados en Modelos

En estos métodos encontramos dos tipos: los que hacen uso de modelos para la generación explícita de residuos a modo de redundancia analítica y aquellos que hacen uso de modelos cualitativos para expresar relaciones causa-efecto. El modelo es usualmente desarrollado con base en el entendimiento previo de las leyes físicas que gobiernan el sistema, en los modelos construidos para generación de residuos este entendimiento es expresado en términos de relaciones matemáticas entre las entradas y salidas del sistema. En los modelos cualitativos dichas relaciones se expresan en términos de funciones cualitativas.

#### Métodos para la Generación de Residuos

La mayoría de la literatura referente a estos métodos usan modelos de entrada-salida y de espacio de estados para la generación de residuos. Esos enfoques pueden ser clasificados en observadores, ecuaciones de paridad, métodos en el dominio de la frecuencia y basado en los datos. Una revisión detallada de estos métodos se puede encontrar en [22], [23] y [24].

**Observadores:** la idea básica detrás de un observador es la estimación de las salidas del sistema a partir de las medidas (o un subconjunto de ellas) usando para ello observadores en un escenario determinista ([25], [26], [27], [28], [29]) o filtros estadísticos (e.g. Filtro de Kalman) en un escenario estocástico [30], [31], [32], [33]. En estos enfoques se usa la estimación del error como residuo. Dependiendo de las circunstancias se pueden usar observadores lineales ([34]), no lineales ([35], [36]), de orden completo o reducido, fijos o adaptativos ([37], [38]).

**Ecuaciones de paridad:** las ecuaciones de paridad son reordenamientos y usualmente variantes transformadas de los modelos de entrada-salida o de espacio de estados de la planta ([39]). La idea básica es comparar la paridad (consistencia) de lo(s) modelo(s) del sistema con las salidas de los sensores (medidas del sistema de monitoreo) y las entradas conocidas del sistema. A partir de estos análisis se reorganiza la estructura del modelo a fin de obtener la mejor capacidad de aislamiento a fallas. Las relaciones dinámicas de paridad fueron introducidas por Willsky [31], Gertler et al. [40], [41] propuso las llamadas 'ecuaciones de paridad ortogonales' en las cuales los vectores residuales son ortogonales entre sí para diferentes fallas mejorando así la capacidad de aislamiento, sin embargo su diseño no es sencillo. Los métodos de diseño de ecuaciones de paridad con propiedades direccionales se presentan en [42] y [41]. Chow y Willsky [43] propusieron un procedimiento para generar ecuaciones de paridad basados en la representación en espacio de estados de un sistema dinámico.

Se ha observado por varios investigadores la existencia de una cierta correspondencia entre los observadores y los enfoques basados en relaciones de paridad. Un análisis completo puede encontrarse en [44].

**Métodos en el dominio de la frecuencia:** los residuos también pueden ser generados en el dominio de la frecuencia mediante la factorización de la función de transferencia del sistema. Un amplio estudio fue realizado por Ding y Frank [45], el cual sirvió de base para posteriores desarrollos que incluyen temas de robustez [46]. Una analogía entre los observadores y los enfoques basados en factorización es realizada en [47] para sistemas lineales.

**Métodos basados en los datos:** pertenecen a la categoría 'data-driven' dado que requieren grandes set de datos históricos para la construcción de modelos. La idea básica yace en hacer un proceso de regresión sobre los datos con el objeto de obtener un conjunto de ecuaciones que describan de forma muy próxima el comportamiento de las mismas y por defecto las del sistema en cuestión. Muchas técnicas de regresión se pueden emplear, están las puramente estadísticas empleadas en identificación de sistemas y estimación paramétrica, y aquellas que emplean técnicas de inteligencia computacional como las redes neuronales. Trabajos relevantes con redes neuronales se pueden encontrar en [48] [12].

## Métodos Cualitativos

Se pueden clasificar en dígrafos, árboles de fallas y los métodos basados en la llamada física cualitativa (qualitative physics).

**Modelos Causales usando dígrafos:** un dígrafo es un grafo con arcos dirigidos conectando diferentes nodos. Cuando dichas conexiones además están sujetas a un signo (+/-) se conocen como SDG's. La dirección de los arcos va de nodos 'causa' a nodos 'efecto' representados de manera grafica las relaciones que dan a lugar con la ocurrencia de una falla. Iri et al [49] fue el primero en usar grafos SDG para la detección y aislamiento de fallas. Umeda et al [50] muestra

cómo construir este grafo a partir de las ecuaciones diferenciales del proceso. Shiozaki et al. [51] añadió sentido de condicionalidad a los arcos en la representación de los arcos. Kokawa et al [52] combino información estadística sobre la fallas de los componentes y SDG's para representar la propagación de fallas e identificar su localización. Trabajos enfocados al diagnóstico de fallas que usan SDG se pueden encontrar en [53], [54].

Recientemente se le han incorporado a los modelos basados en SDG teoría difusa, con el objetivo de mejorar la capacidad de detección de fallas [55]. Li y Wang [56] han presentan como los dígrafos difusos pueden ser utilizado para la simulación cualitativa y cuantitativa del comportamiento de un sistema.

**Arboles de falla:** el análisis de fallas mediante arboles fue desarrollado originalmente en los laboratorios Bell en 1961. Un 'árbol de falla' es un diagrama lógico donde se pueden propagar eventos primarios (los cuales pueden ser fallas) a eventos de alto nivel o de peligro. El árbol posee generalmente capas de nodos. En cada nodo operaciones lógicas como AND y OR se encargan de llevar a cabo para la propagación. Los arboles de Falla han sido usados en diferentes estudios de evaluación de riesgos y fiabilidad [57] [58]. En [59] [60] se encuentra una descripción detallada del análisis y construcción de arboles de falla.

**Enfoques basados en física cualitativa:** el conocimiento en los modelos basados en física cualitativa para el diagnóstico de fallas se han representado en su mayoría de dos maneras. El primer enfoque consiste en derivar ecuaciones cualitativas de las ecuaciones diferenciales, como resultado se obtiene un representación de las relaciones causa-efecto del sistema, ejemplos de investigaciones en el área incluyen [61], [62] y [63]. El otro enfoque es la derivación del comportamiento cualitativo meramente de la observación del mismo. En cuanto a las aplicaciones de los modelos cualitativos en el diagnóstico de fallas, la simulación cualitativa (QSIM) y la teoría de proceso cualitativo (QPT) han sido los enfoques populares. Ejemplos de trabajos de investigación en QSIM incluyen [64], [65] y [66]. Ejemplos de la utilización de QPT para diagnóstico de fallas: [67], [68] y [69].

### 2.2.2. Métodos Libres de Modelos

Para los métodos basados en los datos se debe contar con una amplia cantidad de datos históricos del proceso. Hay diferentes maneras en que estos datos pueden ser transformados y procesados para obtener información útil que permita caracterizar un comportamiento anormal o falla. Este proceso se conoce como extracción de conocimiento y puede clasificarse de forma general en métodos estadísticos multivariados y métodos automáticos de reconocimiento de patrones.

**Métodos estadísticos multivariados:** Aplicaciones de métodos estadísticos multivariantes para el diagnóstico de fallas, tales como Análisis de Componentes Principales (PCA) y mínimos cuadrados parciales (PLS) han sido ampliamente reportados en la literatura. Una visión general del uso de PCA y PLS aplicado a diagnóstico de fallas se puede encontrar en [70] y [71].



**Métodos automáticos de reconocimiento de patrones:** el reconocimiento de patrones es el estudio de cómo las máquinas pueden observar el ambiente o entorno, aprender a distinguir patrones de interés a partir de la experiencia y tomar decisiones razonables con respecto a las categorías a las que pertenecen dichos patrones. Aunque las investigaciones en este campo llevan más de cincuenta años el diseño de un reconocedor/clasificador de patrones artificial de propósito general permanece como una meta lejana. Un patrón es una entidad a la que se le puede dar un nombre [72] [73] como por ejemplo, una imagen de huella digital, una palabra manuscrita, un rostro, una señal de voz, una falla, etc. Dado un patrón, la tarea de su reconocimiento o clasificación puede ser resuelta en principio de dos maneras: de forma supervisada, en la cual el patrón será identificado como miembro de una clase predefinida, o de forma no supervisada en la cual el patrón será asignado a una clase desconocida previamente y aprendida en base a la similitud entre patrones [74]. Es importante mencionar que el problema de detectar y aislar una falla puede ser visto como un problema de clasificación, que en la mayoría de los casos es supervisado, en los siguientes capítulos se ampliará esta idea.

*El conjunto de técnicas y metodologías que permiten explorar grandes set de datos de manera automática, con el objetivo de identificar patrones, tendencias o reglas, se agrupan en una gran área conocida como la minería de datos. Dentro del contexto de los métodos 'data-driven' la minería de datos brinda las herramientas adecuadas para los procesos de detección y aislamiento de fallas, las cuales en su mayoría ya han sido evaluadas y probadas con éxito en otro tipo de aplicaciones.*

Esta tesis emplea diferentes técnicas de minería de datos para la construcción de sistemas FDI a partir de datos históricos del proceso. En el siguiente capítulo se hace una pequeña introducción a la minería de datos enfocándose en su aplicación a problemas de detección y aislamiento de fallas, adicionalmente se presentan algunas técnicas de clasificación que se emplearan posteriormente.



# Capítulo 3

## Minería de datos y su aplicación a sistemas FDI

El propósito de este capítulo es introducir y explicar los aspectos fundamentales de la minería de datos. Se hace un análisis del tipo de problemas que se pueden abordar y como es posible su aplicación en la construcción de sistemas FDI, posteriormente se presentan las principales aplicaciones realizadas en el área, y por último se hace una breve explicación de las diferentes técnicas de clasificación/regresión que se emplearan en esta tesis.

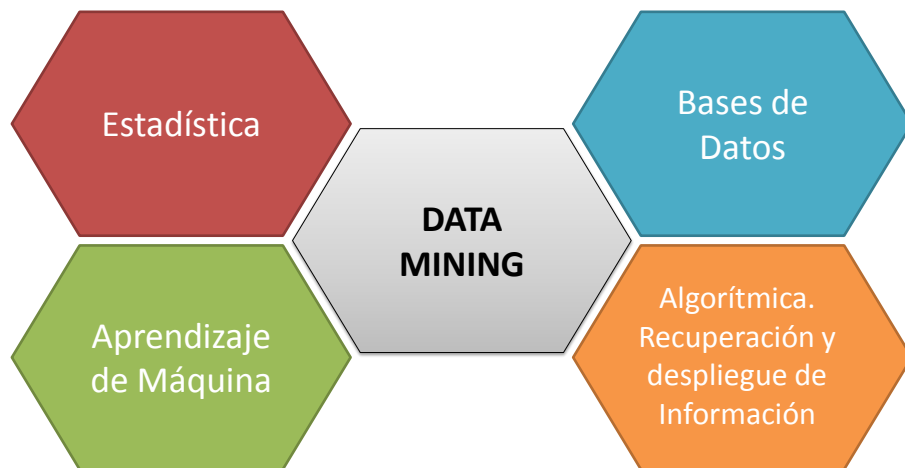
### 3.1. Minería de Datos

Se denomina Minería de Datos al conjunto de técnicas y herramientas aplicadas al proceso no trivial de extraer y presentar conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a partir de grandes conjuntos de datos, con objeto de predecir de forma automatizada tendencias y comportamientos [75] [76].

Con el objeto de descubrir patrones presentes en los datos, hace uso de un gran número de métodos de análisis basados en la estadística y el aprendizaje de máquina. Para el almacenamiento, organización y recuperación de datos hace uso de teoría de bases de datos, datawarehouse y mecanismos lógicos organizacionales.

El proceso de descubrimiento de conocimiento involucra una serie de pasos [77]:

- Entendimiento básico del dominio de aplicación y las metas propuestas.
- Selección de un conjunto de datos en donde realizar el proceso de descubrimiento.
- Limpieza y preprocesamiento de los datos, diseñando una estrategia adecuada para manejar ruido, valores incompletos, valores fuera de rango, valores inconsistentes, etc.

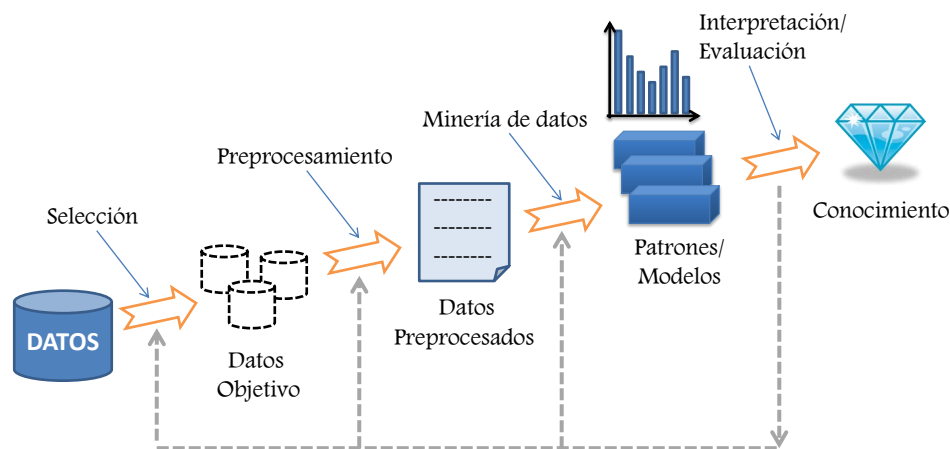


**Figura 3.1** Áreas base de la Minería de Datos

- Selección de la tarea de descubrimiento a realizar, por ejemplo, clasificación, agrupamiento o clustering, reglas de asociación, etc.
- Selección de los algoritmos a utilizar.
- Transformación de los datos al formato requerido por el algoritmo específico de explotación de datos, hallando los atributos útiles, reduciendo las dimensiones de los datos, etc.
- Llevar a cabo el proceso de minería de datos para encontrar patrones interesantes.
- Evaluación de los patrones descubiertos y presentación de los mismos mediante técnicas de visualización. Quizás sea necesario eliminar patrones redundantes o no interesantes, o se necesite repetir algún paso anterior con otros datos, con otros algoritmos, con otras metas o con otras estrategias.
- Utilización del conocimiento descubierto, ya sea incorporándolo dentro de un sistema o simplemente para almacenarlo y reportarlo a las personas interesadas.

Es muy importante la etapa del pre-procesamiento de los datos y su transformación al formato requerido por el algoritmo, ya que dependiendo de cómo se realicen estas tareas, va a depender la calidad final de los patrones descubiertos. Un patrón es interesante si es fácilmente entendible por las personas, potencialmente útil, novedoso o valida alguna hipótesis que el usuario busca confirmar. Un patrón interesante representa conocimiento.

Las principales técnicas de minería de datos se suelen clasificar según su tarea de descubrimiento en:



**Figura 3.2** El proceso de descubrimiento de conocimiento

- Agrupación o *clustering*.
- Clasificación.
- Asociación.

A continuación se realiza una breve descripción de cada una de estas técnicas y los algoritmos más utilizados. Aunque todas las tareas de minería pueden ser empleadas en los procesos de detección y diagnóstico de fallas, las más ampliamente utilizadas son la clasificación y agrupación. Dado el carácter de la tesis una sección de este capítulo está dedicada a la tarea de clasificación.

### 3.1.1. Clustering

La agrupación o clustering consiste en agrupar un conjunto de datos basándose en la similitud de los valores de sus atributos. El clustering identifica regiones densamente pobladas, denominadas clusters, de acuerdo a alguna medida de distancia establecida. De esta manera se busca maximizar la similitud de las instancias en cada cluster y minimizar la similitud entre clusters [78].

Entre los métodos de agrupamiento paramétricos se encuentran los *mixed finite models*, éstas son una poderosa herramienta para modelar los datos como distribuciones de densidades de probabilidad tanto univariadas como multivariadas. En esta tesis se utilizan una mezcla de gaussianas (GMM Gaussian mixed models) para describir de forma probabilística cada una de las fallas, sin embargo dado que se conocen las clases (grupos) de antemano se utilizó la técnica a modo de clasificador y no de agrupamiento. Una introducción a aprendizaje no supervisado usando *mixed finite models* se puede encontrar en [79].

Por otro lado los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad.

Los algoritmos jerárquicos son aquellos en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente, se unen o se dividen dos grupos del nivel anterior. Las estrategias jerárquicas más conocidas son Single Link (SL), Average Link (AL), Complete Link (CL) y el algoritmo Chameleon las cuales se encuentran ampliamente documentadas en la literatura [80] [81].

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo. Estos algoritmos asumen un conocimiento a priori del número de clusters en que debe ser dividido el conjunto de datos, llegan a una división en clases que optimiza un criterio predefinido o función objetivo. Entre los algoritmos que emplean esta técnica podemos mencionar: K-means y sus variantes [82] [83] y CURE [84].

Los algoritmos basados en densidad enfocan el problema de la agrupación teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad. Estos algoritmos usan diversidad de técnicas como grafos, histogramas, kernels, identificación de bordes y cálculo de la entropía- ruido. El principal representante de esta categoría es el algoritmo DBSCAN [85].

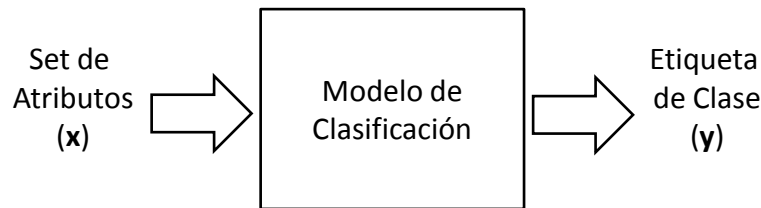
### 3.1.2. Reglas de Asociación

La minería de reglas de asociación consiste en encontrar reglas de la forma  $(A_1, y A_2... y A_m) \Rightarrow (B_1 y B_2...B_n)$  (si entonces), donde  $A_i$  y  $B_j$  son valores de atributos del conjunto de datos. Por ejemplo, se podría encontrar en un gran repositorio de datos de compras en un supermercado, la regla de asociación correspondiente a que si un cliente compra leche, entonces compra pan. Una regla de asociación es una sentencia probabilística acerca de la co-ocurrencia de ciertos eventos en una base de datos, y es particularmente aplicable a grandes conjuntos de datos.

Existen varios algoritmos que realizan el descubrimiento de reglas de asociación, uno de los más utilizados es Apriori [78].

## 3.2. Clasificación

La clasificación encuentra las propiedades comunes entre un conjunto de objetos y los clasifica en diferentes clases, de acuerdo a un modelo de clasificación. Para construir este modelo, se utiliza un conjunto de entrenamiento, en el que cada instancia consiste en un conjunto de atributos y el valor de la clase a la cual pertenece. El objetivo de la clasificación es analizar los datos de



**Figura 3.3** Clasificación: tarea de asignar a conjunto de atributos de entrada ( $x$ ) una etiqueta de clase ( $y$ ).

entrenamiento y, mediante un método supervisado, desarrollar una descripción o un modelo para cada clase utilizando las características disponibles en los datos.

Esta descripción o modelo permite clasificar otras instancias, cuya clase es desconocida (tarea de predicción). El método se conoce como supervisado debido a que, para el conjunto de entrenamiento, se conoce la clase de pertenencia y se le indica al modelo si la clasificación que realiza es correcta o no. La construcción del modelo se realimenta de estas indicaciones del supervisor.

Supongase se tiene un sistema determinado del cual se tiene registro de 4 de sus sensores en diferentes instantes de tiempo, adicionalmente se conoce que en algunos de ellos el sistema presenta algún tipo de fallo. Se contruye el set de datos de la Tabla 3.1, en el cual se tienen los valores de los sensores y el status del sistema. Se desea construir un clasificador que permite a partir de las mediciones de los sensores determinar el estado de operación del sistema (normal/falla).

De forma rápida y haciendo uso de la Tabla 3.1 se definirá el léxico comúnmente usado en minería:

- Los **atributos, variables o características** son las columnas de la tabla.
- Los **objetos, instancias, registros, observaciones o ejemplos** son las filas de la tabla.

De acuerdo a lo anterior una colección de atributos describe a un objeto. Los atributos suelen clasificarse de acuerdo al tipo de valores que pueden tomar, numéricos y continuos como los valores de los sensores o categóricos y discretos como el status de operación. Una descripción detallada de los tipos de atributos se realiza en [78] [75].

Dada la definición del problema el atributo status del set de entrenamiento de la Tabla 3.1 se constituye en nuestro atributo clase. Resumiendo el objetivo es construir un modelo de clasificación que sea capaz de descubrir patrones que le permitan identificar las dos clases, de forma tal que a la llegada de un nuevo conjunto de atributos (valores de los sensores) dicho modelo este en la capacidad de hacer una predicción sobre la clase a la que pertenecen (atributo status).

El problema de detectar y aislar una falla puede ser claramente abordado como un problema de clasificación, sin embargo es indispensable un set de datos de entrenamiento con las siguientes características:

**Tabla 3.1** Ejemplo de un conjunto de datos de entrenamiento

SENSOR 1	ATRIBUTOS				CLASE
	SENSOR 2	SENSOR 3	SENSOR 4	STATUS	
1,38	1,99	3,00	1,01	Normal	
0,74	1,43	1,80	0,51	Normal	
1,25	2,19	2,97	1,34	Normal	
2,22	2,26	2,99	1,80	Normal	
1,88	2,59	3,88	0,60	Normal	
0,22	1,13	1,23	0,11	Falla	
3,77	4,59	6,57	1,73	Falla	
3,01	3,75	5,42	0,96	Falla	
0,05	0,95	0,96	0,17	Falla	
0	0,76	0,76	0,48	Falla	

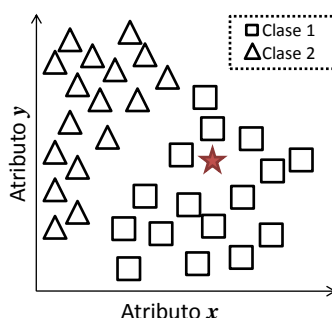
- Debe contener ejemplos de todos los tipos de fallas que se desean clasificar.
- Debe estar correctamente balanceado, esto es el número de registros de cada clase debe ser similar para obtener un modelo equilibrado. Esta puede ser una de las principales desventajas en la construcción de modelos de clasificación para sistemas FDI, dado que los registros de fallas son relativamente bajos en comparación con los registros en un estado de operación normal. Existen diferentes procedimientos para lidiar con esta problemática sin embargo no se abordan en el presente trabajo.
- Los atributos del set de datos de entrenamiento serán las entradas del modelo construido en la fase de predicción, por lo tanto se debe seleccionar solo aquellos atributos que estén disponibles de forma permanente en el sistema.
- No existe un número de ejemplos mínimos para la construcción de modelos, sin embargo, entre mayor sea este número, la fase de entrenamiento será mucho más confiable.

Como se mencionó anteriormente este set de datos pasará adicionalmente por un proceso de pre procesamiento, en el cual se adecuará para la aplicación del algoritmo o técnica para la construcción del modelo.

### 3.3. Algoritmos de Clasificación

En la fase de construcción de un clasificador se debe seleccionar el algoritmo de clasificación a utilizar: por un lado, se puede optar por un clasificador paramétrico, que asume que la distri-





**Figura 3.4** Clasificación de un nuevo ejemplo (estrella) dentro de un conjunto de objetos pertenecientes a dos clases.

bución estadística que siguen los datos es conocida, y trata de estimar los parámetros de dicha distribución. Por el otro lado, se puede elegir un clasificador no paramétrico, en el que no se asume ninguna distribución estadística en particular, y se construye el clasificador utilizando únicamente la información de los datos del conjunto de entrenamiento.

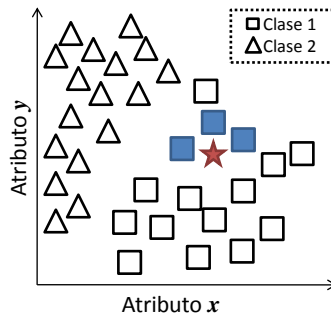
En esta tesis se emplearon 5 diferentes algoritmos: dos paramétricos de tipo Bayesiano (Naive Bayes y Mezcla de Gaussianas) basados en teoría estadística y 4 no paramétricos (KNN, Redes Neuronales y Maquinas de Vectores de Soporte) provenientes del aprendizaje de máquina. A continuación se explica de forma global cada uno de ellos.

### 3.3.1. K-nearest neighbours (KNN)

En este algoritmo, se almacenan los datos de entrenamiento y cuando se quiere clasificar un nuevo set de atributos, se extraen los objetos más parecidos y se usa su clase para clasificar al nuevo set de atributos. Este tipo de aprendizaje se conoce como lazy learning o memorybased learning donde los datos de entrenamiento se procesan solo hasta que se requiere (cuando se requiere constestar alguna pregunta), y la relevancia de los objetos se mide en función de una medida de distancia.

Cuando un nuevo ejemplo se presenta al sistema de aprendizaje, este se clasifica según el comportamiento de los objetos más cercanos (vecinos). A manera de ejemplo en la Figura 3.4 se grafican dos atributos  $x$  y  $y$  y se diferencian dos clases representadas por cuadrados y triangulos. Se desea determinar la clase de un ejemplo (graficado como una estrella).

Lo primero es buscar los  $k$  objetos mas cercanos al nuevo ejemplo que deseamos clasificar, supongamos para este caso que el valor de  $k$  es 3. Para evaluar el concepto de cercanía se hacen uso de diferentes metricas de distancia y similitud, en el caso de atributos continuos la mas utilizada es la distancia euclidiana que para dos datos  $P_1$  y  $P_2$  en un espacio bidimensional se define como:



**Figura 3.5** Los 3 vecinos mas cercanos al nuevo ejemplo en base a la distancia euclidiana.

$$D_{euclidiana}(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.1)$$

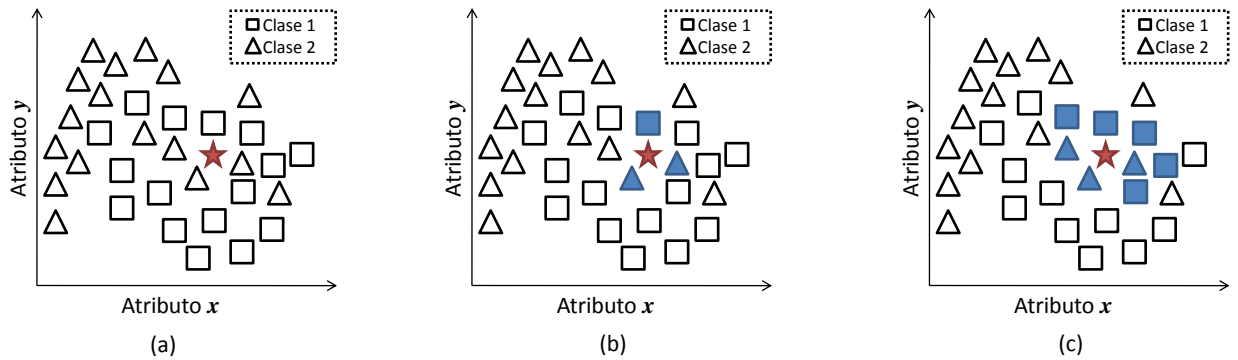
Los tres objetos con la menor distancia euclidiana se muestran en la Figura 3.5. Dado que todos los vecinos cercanos pertenecen a la clase cuadrado el nuevo ejemplo sera asignado a esta clase.

En la Figura 3.6 se muestra un caso mas realista donde los objetos no son linealmente separables. Para este caso la escogencia de un valor adecuado de  $k$  juega un papel fundamental [82]. Para  $k = 3$  (b) el algoritmo asignará el nuevo ejemplo a la clase triangulo mientras que para  $k = 8$  lo asignará a la clase cuadrado (c).

La principal dificultad de este método consiste en determinar el valor de  $k$ , ya que si toma un valor grande se corre el riesgo de hacer la clasificación de acuerdo a la mayoría (y no al parecido), y si el valor es pequeño puede haber imprecisión en la clasificación a causa de los pocos objetos seleccionados como instancias de comparación. Para enfrentar este problema se plantearon diferentes variaciones del método: en cuanto a la forma de determinar el valor de  $k$ , por ejemplo 1-nn, que no es otra cosa más que usar como instancia de comparación al primer vecino más cercano encontrado. También el valor de  $k$  puede hallarse tomando un radio de comparación o mediante el uso de diagramas de Voronoi.

Una característica importante e interesante de  $k$ -nn es que el método puede cambiar radicalmente sus resultados de clasificación sin modificar su estructura, solamente cambiando la métrica utilizada para hallar la distancia. Por lo tanto, los resultados pueden variar tantas veces como métodos de hallar distancia entre puntos haya. La métrica debe seleccionarse de acuerdo al problema que se desee solucionar. La gran ventaja de poder variar métricas es que para obtener diferentes resultados el algoritmo general del método no cambia, únicamente el procedimiento de medida de distancias.

En general las métricas mas ampliamente utilizadas son (en su expresión matemática generalizada) [82]:



**Figura 3.6** (a) Predicción de la clase de un ejemplo nuevo con respecto al vecino más cercano. (b) Vecinos Cercanos para  $K=3$ . (c) Vecinos Cercanos para  $K=8$ .

*Nomenclatura:*  $n$  es la dimensión de los datos.  $P_{1,k}$  y  $P_{2,k}$  son los  $k$ -ésimos atributos de los datos  $P_1$  y  $P_2$  respectivamente:

$$D_{Euclidiana}(P_1, P_2) = \sqrt{\sum_{k=1}^n (P_{1,k} - P_{2,k})^2} \quad \text{Distancia Euclidiana} \quad (3.2)$$

$$D_{Manhattan}(P_1, P_2) = \sum_{k=1}^n |P_{1,k} - P_{2,k}| \quad \text{Distancia de Manhattan} \quad (3.3)$$

$$D_{cos}(P_1, P_2) = \cos^{-1} \frac{\langle P_1, P_2 \rangle}{\|P_1\| \|P_2\|} \quad \text{Distancia Coseno} \quad (3.4)$$

$$D_{Minkowski}(P_1, P_2) = \left( \sum_{k=1}^n |P_{1,k} - P_{2,k}|^r \right)^{\frac{1}{r}} \quad \text{Distancia Minkowski} \quad (3.5)$$

El algoritmo k-nn ha sido ampliamente usado en diferentes aplicaciones dado su sencillez y rápida implementación, sin embargo no es recomendable en problemas no lineales y en aquellos donde las magnitudes de los atributos estén altamente desequilibrados. Un modelo de clasificación construido con k-nn del cual no se obtenga buenos resultados puede ser un indicativo de:

- Existencia de una cantidad considerable de ruido en el set de entrenamiento.
- Las clases sufren de un solapamiento natural en el espacio en que se encuentran.
- El problema que se está resolviendo es altamente no lineal.

### 3.3.2. Clasificadores Bayesianos (CB)

Los clasificadores Bayesianos están basados en teoría de la probabilidad y se basa en la elección de la opción con la mayor probabilidad o el menor riesgo [82].

Asumase que existe un problema para clasificar vectores de atributos (ejemplos) de  $k$  clases diferentes. Un vector de atributos se denota como  $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$  donde  $D$  es la dimensión del vector. La probabilidad que el vector  $x$  pertenezca a la clase  $w_k$  es  $P(w_k|\mathbf{x})$ , y es conocida como la probabilidad a posteriori. La clasificación del vector es realizada acorde a esta probabilidad y pueden ser calculada de acuerdo a la fórmula de Bayes [82]:

$$P(w_k|\mathbf{x}) = \frac{p(\mathbf{x}|w_k)P(w_k)}{p(\mathbf{x})} \quad (3.6)$$

donde  $p(\mathbf{x}|w_k)$  es la función de densidad de probabilidad de la clase  $w_k$  en el espacio de características y  $P(w_k)$  es la probabilidad a priori, la cual nos indica la probabilidad de ocurrencia de la clase. Si las probabilidades a priori no son actualmente conocidas, ellas pueden ser estimadas a partir de las proporciones de la clase en el set de entrenamiento. El divisor:

$$p(\mathbf{x}) = \sum_{i=1}^K p(\mathbf{x}|w_i)P(w_i) \quad (3.7)$$

es meramente un factor de escala que aparece en todas las probabilidades a posterioris y puede ser ignorado. Un ejemplo  $\mathbf{x}$ , se clasifica en la clase  $w_k$  si:

$$P(w_k|\mathbf{x}) > P(w_j|\mathbf{x}) \forall k \neq j \quad (3.8)$$

El mayor problema en los clasificadores Bayesianos es estimar la función de densidad de probabilidad  $p(\mathbf{x}|w_k)$ , la cual nos indica la distribución de los vectores de atributos en el espacio de una clase particular. En la práctica siempre es desconocida por lo tanto debe ser calculada a partir de la información que se tiene 'los datos de entrenamiento'. Se presentan a continuación dos implementaciones para estimarla: el naive bayes y la mezcla de gaussianas.

#### Naive Bayes

El Naive Bayes o clasificador ingenuo estima  $p(\mathbf{x}|w_k)$  asumiendo que los atributos son condicionalmente independientes dada la clase  $k$ . La independencia condicional se define formalmente como:

$$p(\mathbf{x}|w_k) = \prod_{i=1}^D p(x_i|w_k) \quad (3.9)$$

donde  $\mathbf{x} = [x_1, x_2, \dots, x_D]$  es el vector de atributos de tamaño  $D$ .

Por lo tanto para clasificar un nuevo ejemplo el Naive Bayes computa la probabilidad a posteriori para cada clase  $w_k$  y selecciona la máxima:

$$P(w_k|\mathbf{x}) = P(w_k) \prod_{i=1}^D p(x_i|w_k) \quad (3.10)$$

De la ecuación anterior faltan estimar las probabilidades condicionales  $p(x_i|w_k)$ . Al igual que con las probabilidades a priori estas se pueden estimar a partir de los datos de entrenamiento asumiendo además cierta distribución de probabilidad. La distribución gaussiana es la usualmente escogida para representar probabilidades condicionales en atributos continuos [82]. La distribución es caracterizada por dos parámetros, su media  $\mu$ , y su varianza  $\sigma^2$ . Para cada clase  $w_k$  la probabilidad condicional para el atributo  $x_i$  es:

$$p(x_i|w_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}} \quad (3.11)$$

Donde el parámetro  $\mu_{ik}$  puede ser estimado como la media muestral ( $\overline{x_{ik}}$ ) del atributo  $x_i$  de los ejemplos de entrenamiento cuya clase sea  $w_k$ . De forma similar  $\sigma_{ik}^2$  se puede estimar como la varianza muestral. Con los parámetros de las gaussianas se completa la construcción del clasificador, un ejemplo detallado aplicado a un problema de préstamo crediticio se puede encontrar en [86].

### Gaussian Mixture Models

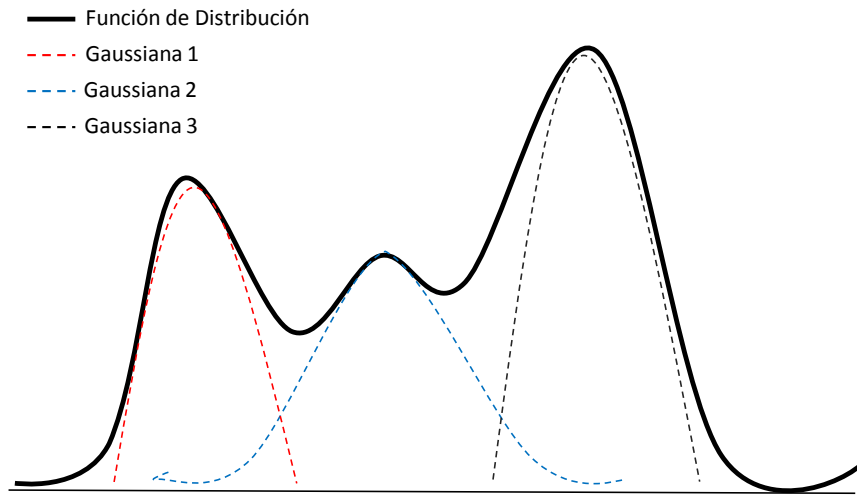
En muchos casos una función de densidad de probabilidad puede ser compleja y no seguir ninguna función de distribución en particular. Para estos casos comúnmente se usan las GMM (Gaussian Mixture Models) las cuales representa funciones complejas como combinaciones lineales de funciones de distribuciones Gaussianas Figura 3.7.

Matemáticamente una GMM para la clase  $k$  se define como:

$$p(\mathbf{x}|w_k) = \sum_{j=1}^{J_k} \alpha_{ij} \mathcal{N}(\mu_{ij}, \Sigma_{ij}) \quad (3.12)$$

Donde  $\mu_{ij}$  y  $\Sigma_{ij}$  son el vector de medias de tamaño  $D$  y  $D \times D$  respectivamente de la  $j$ -th gaussiana de la GMM de la clase  $k$ .  $J_k$  es el número de gaussianas de la GMM de la clase  $k$  y  $\alpha_{ij}$  son los coeficientes o pesos de cada una de ellas.

A diferencia del Naive Bayes en este caso no estamos asumiendo una independencia condicional de los atributos. Una consecuencia directa es el manejo de la distribución gaussiana en un espacio  $D$ -dimensional:



**Figura 3.7** Aproximación de una función de distribución de probabilidad mediante una suma ponderada de Gaussianas

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \mu)^t \Sigma^{-1} (\mathbf{x} - \mu) \right] \quad (3.13)$$

La estimación de los parámetros de las  $k$  GMM se realiza utilizando el algoritmo EM (Expectation-Maximization). El proceso no es trivial y su costo computacional aumenta considerablemente de acuerdo a los valores de  $J_k$ . Una descripción general del algoritmo EM y su aplicación a los GMM puede ser encontrada en [87] y [82].

### 3.3.3. Máquinas de Vectores de Soporte (SVM)

La teoría de las SVMs fue desarrollada inicialmente por V. Vapnik [18] a principios de los años 80 y se centra en lo que se conoce como Teoría del Aprendizaje Estadístico. A continuación se desarrollará brevemente el planteamiento general de las SVMs para posteriormente centrarnos en los problemas de clasificación desde la perspectiva de un aprendizaje supervisado, es decir, el conocimiento de las salidas de un conjunto de entradas nos permite cuantificar (supervisar) la precisión de los resultados del modelo.

El objetivo fundamental de este tipo de estudios es aprender a partir de los datos y para ello busca la existencia de alguna dependencia funcional entre un conjunto de vectores de entrada (atributos):

$$\{x_i, i = 1, \dots, n\} \subseteq X \subseteq \mathbb{R}^d \quad (3.14)$$

y valores de salida (clases):

$$\{y_i, i = 1, \dots, n\} \subseteq Y \subseteq \mathbb{R} \quad (3.15)$$

Por lo tanto un set de entrenamiento se construye de la forma:

$$Z = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq X \times Y \quad (3.16)$$

### Modelo lineal de vectores soporte

Se denota las dos posibles clases como  $Y = \{-1, 1\}$ . La idea básica detrás de las SVM es encontrar un hiperplano óptimo que maximice la separación de las clases 'margen'. Encontrar dicho hiperplano es equivalente a solucionar un problema de optimización con restricciones [88], cuya solución es una combinación lineal de ejemplos de entrenamiento que yacen sobre el margen los cuales se conocen como 'support vectors'.

Un conjunto de vectores  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  donde  $x_i \in \mathbb{R}^d$  e  $y_i \in \{-1, 1\}$  para  $i = 1, \dots, n$  se dice separable si existe algún hiperplano en  $\mathbb{R}^d$  que separa los vectores  $X = \{x_1, \dots, x_n\}$  con etiqueta  $y_i = 1$  de aquellos con etiqueta  $y_i = -1$ .

Dado un conjunto separable existe (al menos) un hiperplano:

$$\pi : w \cdot x + b = 0 \quad (3.17)$$

que separa los vectores  $x_i, i = 1, \dots, n$  (ver Figura 3.8). Las SVMs buscan entre todos los hiperplanos separadores aquel que maximice la distancia de separación entre los conjuntos  $\{x_i, 1\}$  y  $\{x_i, -1\}$  (las dos clases posibles).

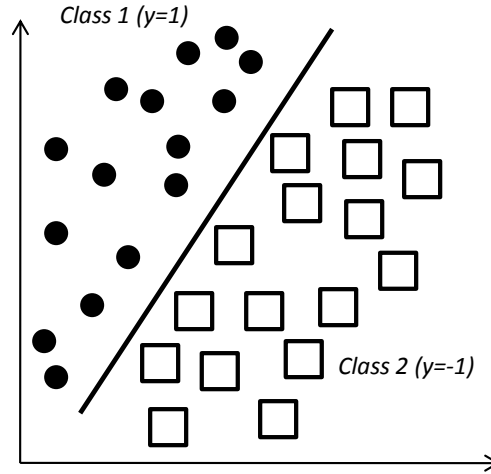
Fijado un hiperplano separador siempre es posible reescalar los parámetros  $w$  y  $b$  de tal forma que:

$$x_i \cdot w + b \geq +1 \quad \text{para } y_i = +1 \quad (\text{region A}) \quad (3.18)$$

$$x_i \cdot w + b \leq -1 \quad \text{para } y_i = -1 \quad (\text{region B}) \quad (3.19)$$

De esta forma la mínima separación entre los vectores y el hiperplano separador es la unidad y las dos desigualdades se pueden expresar en una sola de la forma [88]:

$$y_i(x_i \cdot w + b) - 1 \geq 0, \quad i = 1, \dots, n \quad (3.20)$$



**Figura 3.8** Hiperplano separador en  $\mathbb{R}^2$ , para dos clases

Sean los vectores de etiqueta 1 para los cuales se cumple la ecuación anterior. Estos puntos pertenecen al hiperplano  $\pi_1 : x_i \cdot w + b = 1$  con vector normal  $w$  y distancia perpendicular hasta el origen igual a  $|1 - b| / \|w\|$  donde  $\|w\|$  es la norma euclídea de  $w$ . Análogamente, los vectores de etiqueta  $-1$  que cumplen la ecuación pertenecen al hiperplano  $\pi_2 : x_i \cdot w + b = -1$  con vector normal  $w$  y distancia perpendicular al origen de coordenadas igual a  $|-1 - b| / \|w\|$ . Así se tiene que los hiperplanos  $\pi_1$  y  $\pi_2$  son paralelos, la separación entre ellos es  $2 / \|w\|$  y ningún vector del conjunto de entrenamiento se encuentra entre ellos.

De entre las posibles elecciones de los hiperplanos  $\pi_1$  y  $\pi_2$ , parece natural elegir aquella que proporcione una mayor separación entre ellos, ya que de esta forma permitiría distinguir de forma más clara las regiones donde caen los puntos con distintas etiquetas. Así se plantea:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 \quad (3.21)$$

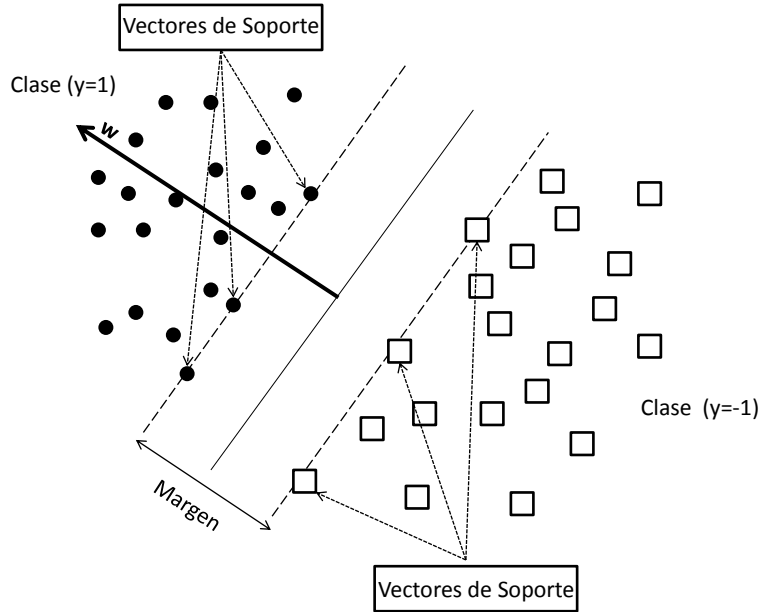
$$\text{s.a. } y_i(x_i \cdot w + b) - 1 \geq 0, i = 1, \dots, n \quad (3.22)$$

La solución para el caso linealmente separable se puede interpretar gráficamente a partir de la Figura 3.9.

Para resolver el problema de optimización con restricciones (3.21-3.22) se utiliza los multiplicadores de Lagrange. Así la función objetivo se construye como:

$$L_P(w, b, \alpha_i) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i(x_i \cdot w + b) - 1) \quad (3.23)$$





**Figura 3.9** Ejemplo de una SVM para dos clases linealmente separables

Siendo este último un problema de programación cuadrática de tipo convexo. Esto significa que se puede resolver el problema dual asociado al problema primal: maximizar la función  $L_P(w, b, \alpha_i)$  respecto a las variables duales  $\alpha_i$  sujeta a las restricciones impuestas para que los gradientes de  $L_P$  con respecto a  $w$  y  $b$  sean nulos, y sujeta también al conjunto de restricciones  $C_2 = \{\alpha_i \geq 0, i = 1, \dots, n\}$  (para más detalles del planteamiento dual consultar [88]). La solución de este problema se expresa en la forma:

$$w = \sum_{i=1}^n \alpha_i y_i x_i, \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.24)$$

y la función objetivo dual:

$$L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \quad (3.25)$$

Los vectores del conjunto de entrenamiento que proporcionan un multiplicador  $\alpha_i > 0$  son denominados vectores soporte y se encuentran en uno de los hiperplanos  $\pi_1$  o  $\pi_2$  (Figura 3.9). Una combinación lineal de vectores de soporte construyen los hiperplanos (dados por el vector normal  $w$ ) y por tanto la solución del problema, de forma tal que el resto de puntos del conjunto de entrenamiento juegan un papel secundario y no son necesarios para tareas de predicción. Si se

trabaja con una gran cantidad de entradas, es útil trabajar con las máquinas de vectores de soporte ya que estos forman un esquema de comprensión que permite reconstruir la solución del problema en terminos de un volumen reducido de información.

Del problema de optimización se puede establecer la condición ([88]):

$$\alpha_i \cdot (y_i \cdot (x_i \cdot w + b) - 1) = 0 \quad (3.26)$$

denominada condición (complementaria) de Karush-Kuhn-Tucker (KKT). Estas restricciones indican que el producto de las restricciones del problema primal ( $y_i \cdot (x_i \cdot w + b) - 1 \geq 0$ ) y las restricciones del problema dual ( $\alpha_i \geq 0$ ) se anulan en todos los vectores de entrenamiento. De esta forma se sigue que las condiciones KKT, para el problema primal definido a partir de la función objetivo son las siguientes:

$$w_j - \sum_{i=1}^n \alpha_i y_i = 0 \quad j = 1, \dots, d \quad (3.27)$$

$$\frac{\partial}{\partial b} L_P = - \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.28)$$

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall_i = 1, \dots, n \quad (3.29)$$

$$\alpha_i \geq 0 \quad \forall_i = 1, \dots, n \quad (3.30)$$

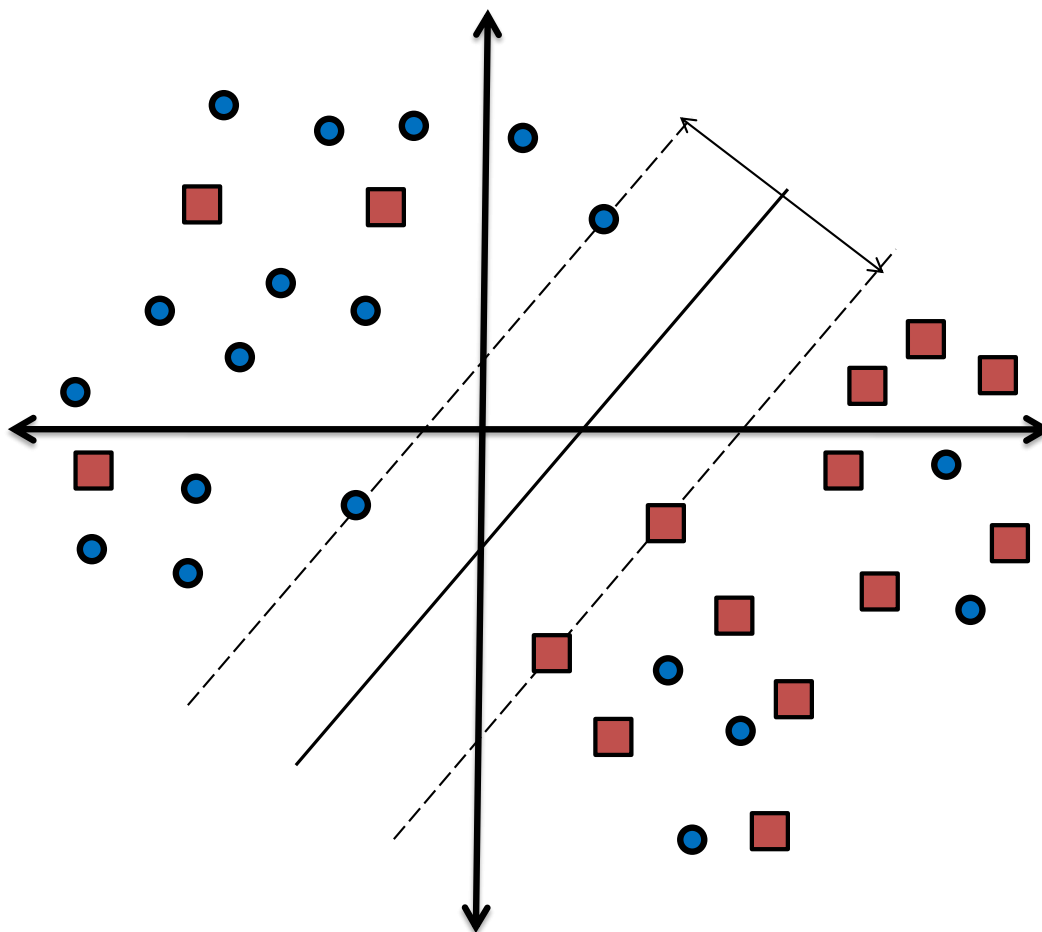
$$\alpha_i(y_i(x_i \cdot w + b) - 1) = 0 \quad \forall_i = 1, \dots, n. \quad (3.31)$$

De los desarrollos iniciales no se sigue una forma explícita para determinar el valor  $b$ , sin embargo, la condición KKT complementaria nos permite determinarlo. Para ello, basta elegir un  $\alpha_i > 0$  y despejar el valor de  $b$  obteniendo  $b = y_i - x_i \cdot w$ . Aunque se ha determinado  $b$ , es más adecuado realizar los cálculos con todos los  $\alpha_i > 0$  y elegir como valor de  $b$  un valor promedio de los resultados obtenidos, con objeto de redondear los errores intrínsecos asociados a todo método de cálculo numérico:

$$b = \frac{1}{\#\{\alpha_i > 0\}} \sum_{\alpha_i > 0} (y_i - x_i \cdot w) \quad (3.32)$$

Una vez obtenidos el vector  $w$  y la constante  $b$  la solución al problema de optimización se interpreta a partir de una función  $\Theta$  de la siguiente forma:

$$\Theta(x) = \begin{cases} 1, & \text{si } \pi(x) > 0, \\ -1, & \text{si } \pi(x) < 0. \end{cases} \quad (3.33)$$



**Figura 3.10** SVM, caso no linealmente separable

### SVMs para datos no linealmente separables

En la práctica no es habitual trabajar con conjuntos separables. En casos como los de la Figura 3.10 donde se encuentran ejemplos de una clase dentro de la región correspondiente a otra, el modelo lineal explicado anteriormente falla.

Situaciones como esta se conocen como no linealmente separables y es necesario reestructurar el problema de optimización introduciendo una variable  $\xi$  de holgura en las restricciones y plantear un nuevo conjunto de restricciones.

$$x_i \cdot w + b \geq +1 - \xi_i \quad \text{para } y_i = +1 \quad (3.34)$$

$$x_i \cdot w + b \leq -1 + \xi_i \quad \text{para } y_i = -1 \quad (3.35)$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n \quad (3.36)$$

Se tiene ahora que para que se produzca un error en la clasificación (una entrada no es ubicada en la clase correcta) es necesario que el valor correspondiente a  $\xi_i$  sea superior a la unidad. Así, si en el vector  $x_i$  se comete un error entonces  $\xi_i \geq 1$  y por tanto  $\sum_i \xi_i$  es una cota superior del número de errores que se cometen dentro del conjunto de entrenamiento. Con el objeto de penalizar los errores se asigna a la función objetivo un coste extra  $C$ :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i^k \quad (3.37)$$

con  $k \geq 1$ . Un valor  $C$  grande, significa que el investigador está asignando un peso a los errores muy alto frente a  $\|w\|^2$ , y por el contrario si  $C$  es pequeño asigna un mayor peso a  $\|w\|^2$ . Por otro lado si  $k$  es grande lo que hacemos es dar mucho más peso a los errores cuantos mayores sean éstos. Se llega por tanto a plantear un problema de programación convexa para cualquier valor de  $k$ . Si  $k = 2$  ó  $k = 1$  se tiene un problema de programación convexa cuadrático. En el presente trabajo se considera  $k = 1$  ya que en este caso se tiene la ventaja de que ningún valor  $\xi_i$ , ni ninguno de sus correspondientes multiplicadores de Lagrange, aparecen en el problema dual. Por tanto el problema de optimización que se plantea es:

$$\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (3.38)$$

$$\text{s.a. } y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0, \forall i \quad (3.39)$$

$$\text{s.a. } \xi_i \geq 0, \forall i \quad (3.40)$$

Utilizando la técnica de los multiplicadores de Lagrange se llega a la función objetivo dual:

$$L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \quad (3.41)$$

la cual hay que maximizar respecto  $\alpha_i$  sujeta a:

$$0 \leq \alpha_i \leq C, i = 1, \dots, n \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.42)$$

y cuya solución final viene dada por:

$$w = \sum_{i=1}^{N_{sv}} \alpha_i y_i s_i \quad (3.43)$$

donde  $N_{sv}$  denota el número de vectores soporte y por  $s_i$  los vectores soporte del conjunto  $\{x_1, \dots, x_n\}$ . Claramente  $N_{sv} \leq n$  y una de las características más interesante de estos modelos es que eligiendo adecuadamente los parámetros es posible conseguir que  $N_{sv}$  sea muy inferior a  $n$  con lo que se consigue una representación 'corta' de la solución en función de los vectores de entrada sin perder capacidad de generalización.

Nótese, que la única diferencia en la solución con respecto a la dada en el caso separable es que los multiplicadores de Lagrange  $\alpha_i$ , están acotados superiormente por la constante  $C$ .

Las condiciones de KKT asociada a este problema son las siguientes:

$$\frac{\partial}{\partial w_j} L_P = w_j - \sum_{i=1}^n \alpha_i y_i x_{ij} = 0 \quad (3.44)$$

$$\frac{\partial}{\partial b} L_P = - \sum_{i=1}^n \alpha_i y_i = 0 \quad (3.45)$$

$$\frac{\partial}{\partial \xi_i} L_P = C - \alpha_i - \mu_i = 0 \quad (3.46)$$

$$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0 \quad (3.47)$$

$$\xi_i, \alpha_i, \mu_i \geq 0 \quad (3.48)$$

$$\alpha_i(y_i(x_i \cdot w + b) - 1 + \xi_i) = 0 \quad (3.49)$$

$$\mu_i \xi_i = 0 \quad (3.50)$$

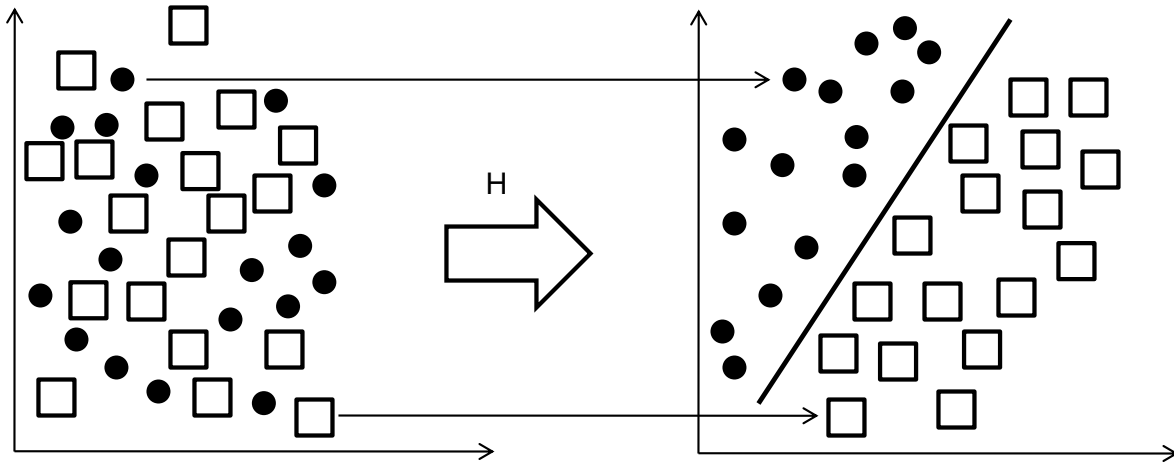
### Máquinas no lineales de vectores soporte

En esta parte se aborda el problema de generalizar los desarrollos anteriores para el caso de clases de funciones no necesariamente lineales en los parámetros. Para ello, obsérvese que los vectores de entrada forman parte de la solución del problema de clasificación, a través de los productos escalares,  $\langle x_i, x \rangle$ . Sea una función  $\Phi$  del conjunto de entradas  $X$ , en un espacio  $H$  (denominado espacio de características) dotado de un producto escalar.

$$\Phi : X \subset \mathbb{R}^d \rightarrow H. \quad (3.51)$$

Ahora, en lugar de considerar el conjunto de vectores  $\{x_1, \dots, x_n\}$  se considera los vectores transformados  $\{\Phi(x_1), \dots, \Phi(x_n)\}$  y si se plantea el problema de optimización original a estos vectores se tiene que los nuevos vectores forman parte de la solución del problema solo a través del producto escalar definido en  $H : \langle \Phi(x_i), \Phi(x) \rangle$ . Así, si se considera una función

$$k : H \times H \rightarrow \mathbb{R} \quad (3.52)$$



**Figura 3.11** SVM, uso de la función de kernel para transformar a un espacio nuevo  $H$

(denominada función kernels), tal que

$$k(x, x') = \Phi(x) \cdot \Phi(x') = \left\langle \Phi(x), \Phi(x') \right\rangle_H \quad (3.53)$$

solo es necesario conocer la función de kernel para resolver el algoritmo y no se necesita tener la forma explícita de la aplicación  $\Phi$ .

Por tanto si se reemplaza  $\langle x_i, x \rangle$  por  $k(x_i, x)$  en el problema de optimización, se consigue una máquina de vectores soporte planteada en un nuevo espacio. Al resolver este problema en  $H$ , donde se trabaja con funciones lineales, la solución que resulta es lineal en él, pero no es necesariamente lineal en el espacio de entradas  $X$ , con lo cual se esta generalizando el problema a conjuntos de funciones no lineales (ver Figura 3.11).

Si se lleva a cabo la transformación de los datos, el vector solución es

$$w = \sum_{i=1}^{N_{SV}} \alpha_i y_i \Phi(s_i) \quad (3.54)$$

donde  $\Phi(s_i)$  denota los vectores soporte del conjunto  $\{\Phi(x_1), \dots, \Phi(x_n)\}$ . Es importante notar que los vectores soporte se encuentran dentro del conjunto  $\{\Phi(x_1), \dots, \Phi(x_n)\}$ , se denotan por  $\Phi(s_i)$ . Aunque parece necesario realizar la transformación de los datos de entrada al espacio  $\Phi$ , no se necesita ya que la solución queda:

$$f(x) = \sum_{i=1}^{N_{SV}} \alpha_i y_i \langle \Phi(s_i), \Phi(x) \rangle + b \quad (3.55)$$

y escrita en términos de la función kernels:

$$f(x) = \sum_{i=1}^{N_{SV}} \alpha_i y_i k(s_i, x) + b \quad (3.56)$$

Una importante consecuencia de la representación dual (en términos del kernel) es que la dimensión del espacio característico no afecta a los cálculos ya que la única información necesaria se encuentra en la matriz de orden  $n \times n$ :  $\{k(x_i, x_j)\}_{i,j=1}^n$  (denominada matriz de Gram).

Al usar  $\Phi : X \rightarrow H$  se trabaja en un nuevo espacio  $H$  por lo cual el vector solución  $w$  se encuentra en este espacio.

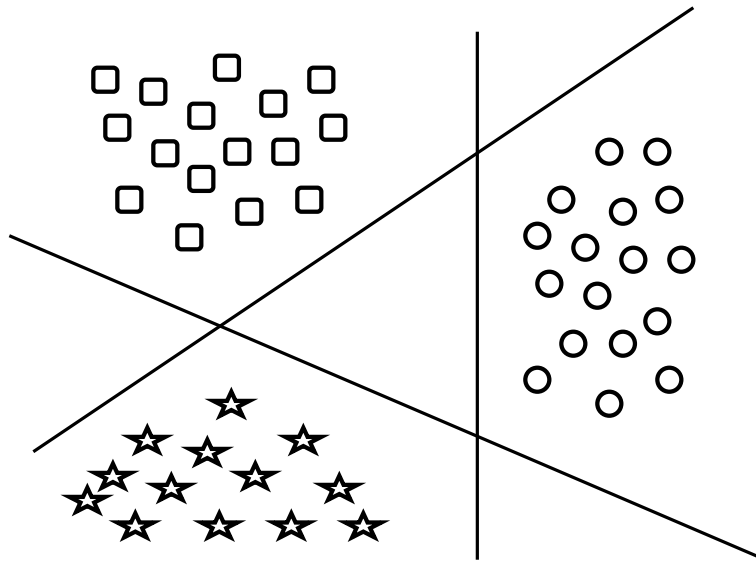
### Clasificación múlticlasa

Para extender las maquina de vectores de soporte a un caso multiclase se empleo la técnica de One-Against-All . Para  $k$  clases esta estrategia construye  $k$  clasificadores binarios uno para cada clase. El  $i_{th}$  clasificador  $f_{C_i}(x)$  es entrenado para identificar la clase  $y_i$  del resto de clases. En el entrenamiento los ejemplos asociados a la clase  $y_i$  son asignados como ejemplos positivos (+1) y el resto de ejemplos como negativos (-1). Los clasificador  $f_{C_i}(x)$  construidos se combinan para dar una función de decisión final  $f_C(x)$ :

$$f_C(x) = \operatorname{argmax}_i \{f_{C_i}(x)\}_{i=1}^M \quad (3.57)$$

#### 3.3.4. Redes Neuronales Artificiales (ANN)

Mediante las redes neuronales artificiales se intenta imitar el proceso de aprendizaje del cerebro humano. El cerebro esta formado por miles de millones de neuronas conectadas entre sí. Utilizan información que es percibida, trasmitida hasta las neuronas, y allí procesada por ellas para dar una respuesta a cada uno de los diferentes estímulos. Cada neurona tiene tres partes: un cuerpo celular, una estructura de entrada (Dendrita) y una de salida (Axon). La mayoría de las terminales de los axones se conectan con las dendritas de otras neuronas (Sinápsis). El comportamiento de una neurona es el siguiente: recibe una señal de entrada con una fuerza determinada, dependiendo de ellas la neurona emite una señal de respuesta, las sinápsis pueden variar en fuerza, algunas pueden dar una señal débil y otras una fuerte. A una neurona pueden llegar miles de señales de entrada, cada una con una fuerza o peso diferente. Matemáticamente el comportamiento de la neurona puede representarse por una lista de sus señales de entrada que son multiplicadas por sus pesos respectivos y posteriormente sumados, el resultado es llamado nivel de activación de la neurona del cual depende la señal de salida que es enviada a cada una de las neuronas a las que esta conectada a ella.



**Figura 3.12** SVM, Clasificación múltiple

Una red neuronal artificial (RNA) es un sistema compuesto de muchos elementos procesadores simples, neuronas, conectados en paralelo, cuya función es determinada por la estructura de la red, la fuerza en las conexiones y el procesamiento realizado por los elementos en los nodos. Las RNA, igual que las personas, aprenden de la experiencia. Las neuronas están conectadas por medio de sinapsis, estas conexiones son direccionadas, es decir, la información sólo puede propagarse en un sentido. Las neuronas pueden agruparse en unidades estructurales denominadas capas y el conjunto de una o más capas constituye la red neuronal. Las conexiones pueden ser excitatorias o inhibitorias, según sea el peso sináptico positivo o negativo, respectivamente. Las conexiones también pueden ser intracapa o intercapa, según se conecten neuronas de una misma capa o de capas diferentes. Según el flujo de datos, las redes pueden ser unidireccionales o recurrentes.

Uno de los primeros modelos de redes neuronales fue el Perceptron diseñado por Rosenblatt [89], el cual contaba con tres tipos de neuronas: sensoriales, asociativas y de respuesta. Las sensoriales tomaban entradas de fuera de la red, las unidades de respuesta llevaban señales fuera de la red al mundo externo y las asociativas eran exclusivamente internas. Estos tipos ahora son llamados unidades de entrada, de salida y ocultas. Rosenblatt desarrolló métodos para que la red alterara los niveles sinápticos de forma que la red aprendiera a reconocer niveles de entrada.

Después de los años 80 la inteligencia artificial se hizo más popular y aumentó su campo de aplicación. Se han desarrollado algoritmos para modelar procesos mentales de alto nivel como la asociación de conceptos, deducción, inducción y razonamiento. En 1986 Rumelhart y McClelland [89] demostraron que algunos problemas imposibles para los Perceptrons simples pueden ser resueltos por redes multi-nivel con funciones de activación no lineales, utilizando procesos simples



de entrenamiento (algoritmo Back-propagation).

### Características generales

*Pesos:* Las redes neuronales pueden tener factores de peso adaptable o fijo. Las que tienen pesos adaptables emplean leyes de aprendizaje para ajustar el valor de la fuerza de interconexión con otras neuronas. Si se utilizan pesos fijos, su tarea debe estar previamente definida. Los pesos son determinados a partir de una descripción completa del problema a tratar. Los pesos adaptables son muy importantes, ya que no se conoce de antemano su valor correcto.

*Aprendizaje:* Se utilizan dos tipos de aprendizaje: supervisado y no supervisado. En el primero se le proporciona a la red tanto la salida como la entrada correcta, y la red ajusta sus pesos para disminuir el error en la salida que ella calcula. Este tipo es utilizado principalmente en el conocimiento de patrones. En el aprendizaje no supervisado a la red se le proporcionan únicamente los estímulos, y la red ajusta los pesos de sus interconexiones utilizando solamente los estímulos y la salida calculada por la red. El aprendizaje no supervisado puede describirse como la estimación de la función de densidad de probabilidad  $p(x)$  que describe la distribución de patrones  $x$  pertenecientes a un espacio de entrada  $R_n$ . La forma de aprendizaje que se utiliza depende del tipo de problema que se intenta resolver.

### Tipologías

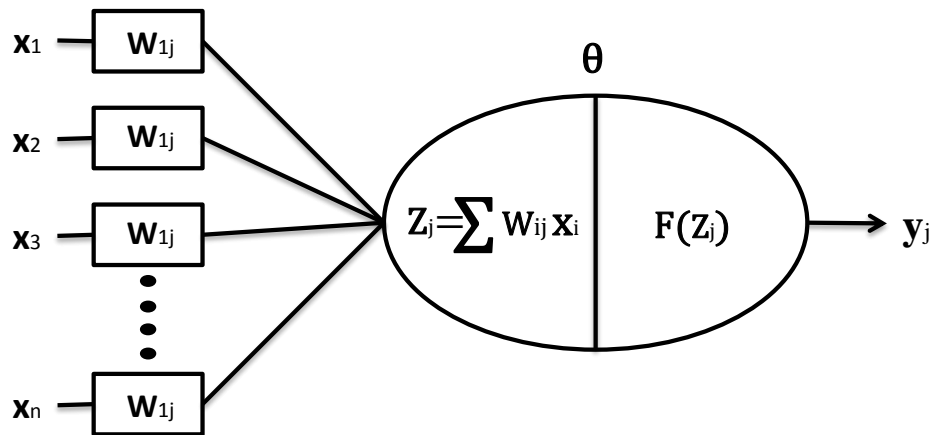
Dependiendo de la estructura de la red neuronal existen varias tipologías, sin embargo las más conocidas son: el Perceptrón Simple y Perceptrón Multicapa.

#### *Perceptrón Simple*

En la Figura 3.13 se observa el modelo de un perceptrón para reconocimiento de patrones. El cuerpo de la neurona es representado como un sumador lineal de estímulos  $z_j$ , seguida por una función no lineal  $f(z_j)$  que es la función de activación, esta utiliza la suma de estímulos para determinar la salida de la neurona. En el perceptrón cada entrada es multiplicada por un factor de peso  $w$  correspondiente, los resultados son sumados y luego evaluados contra el valor de umbral  $\theta$ , si el valor es mayor al máximo el perceptrón se activa.

La primera capa actúa como un grupo de sensores que detectan una señal específica con la importancia correspondiente. La segunda capa toma las salidas de los detectores y clasifica el patrón de entrada dado. El aprendizaje empieza ajustando la fuerza en las conexiones (pesos  $w_i$ ) y el valor de umbral  $\theta$ . La función  $g_i$  es fija y está predeterminada.

El entrenamiento de un perceptrón se hace utilizando una función de aprendizaje. Si la respuesta dada por el perceptrón es incorrecta, se modifican todos los pesos de las conexiones de acuerdo a la siguiente función:



**Figura 3.13** Perceptrón Simple

$$\Delta w_i = n t_i x_i \quad (3.58)$$

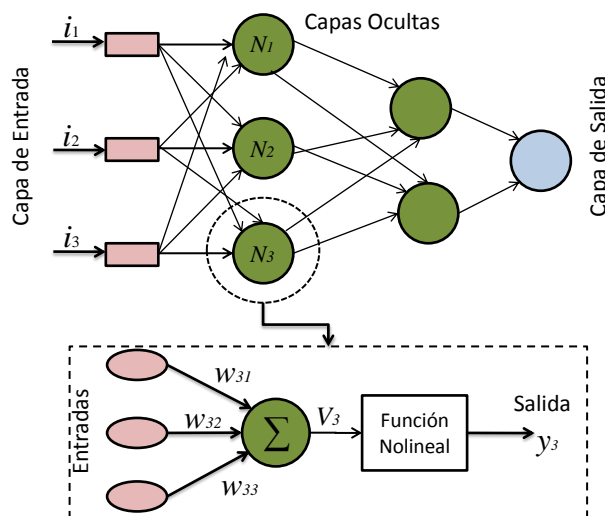
donde  $n$  es la tasa de aprendizaje,  $t_i$  en la salida deseada, y  $x$  es la entrada del perceptron.

La función anterior puede ser utilizada también para la actualización del valor de umbral  $\theta$ . El valor de  $n$  puede ser constante a lo largo del entrenamiento, o puede variar en forma proporcional al error. Un  $n$  proporcional al error lleva a una convergencia rápida, pero causa un aprendizaje variable.

### **Perceptrón Multicapa**

Cuando se combinan varios perceptrones en una capa y los estímulos de entrada después son sumados, se tiene ya una red neuronal multicapa. El perceptron multicapa (Multilayer Perceptron MLP) es la red neuronal artificial mas conocida y con un mayor número de aplicaciones. Está formada por una capa de entrada, al menos una capa oculta y una capa de salida (Figura 3.14). Las conexiones entre las neuronas o pesos sinápticos de cada neurona son optimizados en la fase de aprendizaje. La propagación se realiza de manera que cada neurona hace una combinación lineal de las señales procedentes de las neuronas de la capa anterior cada cual ponderado por su respectivo peso sináptico.

La función de activación cumple con el objetivo de limitar el rango de salida de la neurona y puede ser lineal o no lineal. Se selecciona de acuerdo con el problema y a criterio del investigador, en ocasiones por ensayo y error, también depende de la precisión y velocidad requerida y del algoritmo de aprendizaje escogido. De acuerdo con el rango de las funciones de activación utilizadas es conveniente escalar o transformar los datos de entrada para ajustarlos a dichos rangos. Una buena función de activación debería cumplir: primero, que ella misma y su derivada sean fáciles de computar y segundo, que la función debe tener una amplia parte lineal para lograr velocidad de



**Figura 3.14** Perceptrón Multicapa

entrenamiento y convergencia en pocos ciclos. La función sigmoideal es usada comúnmente porque su derivada es una de las más fáciles de computar.

La falta de métodos de entrenamiento apropiados para los perceptrones multicapa (MLP) hizo que declinara el interés en las redes neuronales en los años 60 y 70. Esto no cambió hasta la reformulación del método de entrenamiento para MLPs Backpropagation a mediados de los años 80 por Rumelhar. En esta red, se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa se conectan entre sí. Cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa. En la Figura 3.14 se observa la arquitectura de este tipo de red.

El término backpropagation se refiere al método para calcular el gradiente de error en una red feed-forward. Básicamente el entrenamiento de este tipo de red consiste en lo siguiente:

Pasada hacia delante (forward pass): las salidas son calculadas y el error en las unidades de salida es calculado. Pasada hacia atrás (backward pass): el error de las salidas es utilizado para alterar los pesos de las unidades de salida. Luego el error en las neuronas de las capas ocultas es calculado mediante propagación hacia atrás del error en las unidades de salida, y los pesos en las capas ocultas son alterados usando esos valores.

### 3.4. Minería de datos aplicada al desarrollo de sistemas FDI

Recientemente se han empleado técnicas de minería de datos al desarrollo de sistemas FDI, en [2] usan modelos ocultos de Markov y filtros de Kalman para realizar detección de fallas en sistemas industriales, en [90] realizan agrupamiento a través de la distancia de Mahalanobis, en [5]

utilizan métodos de kernel sobre series de tiempo y en [14] usan redes Bayesianas en ambientes simulados. Las publicaciones enfocadas en sistemas FDI para turbinas de viento son escasas y la mayoría se especializan solo en algunos tipos de fallas, en [3] usan un sistema clasificador basado en distancias para detectar fallas en el generador a través de un sistema de monitoreo por vibración, en [91] usan el algoritmo de KNN y redes neuronales para detectar fallas en las hojas y rotor de la turbina y en [4] plantean un clasificador basado en distancias ponderadas para la identificación de fallas en el generador y la caja de cambios. En [13] emplearon técnicas de clasificación para la predicción de estados de falla en turbinas de viento.

Aunque estas investigaciones son recientes y en su mayoría no han sido probadas sobre sistemas industriales reales, han demostrado ser un camino prometedor alternativo a las bien conocidas técnicas basadas en modelos. En el siguiente capítulo se expondrá como caso de estudio la turbina de viento así como los modelos simulados sobre los cuales se trabajará, luego en capítulos posteriores se realizará la construcción de diferentes sistemas FDI variando la arquitectura del mismo.

# Capítulo 4

## Caso de estudio: Turbina de viento

El mercado de energía eólica se ha desarrollado muy rápidamente en el mundo. Como lo muestra la Tabla 4.1 se incrementó la capacidad instalada a una tasa promedio de 30.47% durante los años 1998 - 2008 [6]. Muchos países del mundo están invirtiendo cada vez más dinero en la implementación de energías alternativas que les permitan disminuir su dependencia de las energías tradicionales no renovables.

**Tabla 4.1** Energía eólica instalada a nivel mundial (1998-2008)

<b>Año (98-08)</b>	<b>Nueva capacidad instalada (MW)</b>	<b>Tasa de incremento (%)</b>	<b>Capacidad global total (MW)</b>	<b>Tasa de incremento (%)</b>
1998	2187	-	9667	-
1999	4033	84,41	13700	41,72
2000	4339	7,59	18039	31,67
2001	6283	44,8	24322	34,83
2002	6859	9,17	31181	28,2
2003	8114	18,3	39295	26,02
2004	8398	3,5	47693	21,37
2005	11331	34,93	59024	23,76
2006	15127	33,5	74151	25,63
2007	19776	30,73	93927	26,67
2008	27261	37,84	121188	29,02
Promedio	-	<b>30,477</b>	-	<b>28,889</b>

Uno de los mayores inconvenientes en la industria eólica son los costos de mantenimiento necesarios para mantener una turbina en funcionamiento. Estos costos en su mayoría son producto de fallas que no fueron detectadas con suficiente antelación como para tomar medidas correctivas

que evitaran un daño permanente en la turbina. Según estadísticas de la 'World Wind Energy Association' la mayoría de los mantenimientos realizados a turbinas de viento son para reemplazar partes dañadas que juegan un papel crítico en el funcionamiento de la misma [6], en Alemania la IWES 'Fraunhofer-Institut für Windenergie und Energiesystemtechnik' establece que el 63,1 % de las fallas que se presentaron en el 2009 tuvieron como consecuencia el apagado temporal (o en algunos casos permanente) de la turbina [7]. Se estima que los costos totales de mantenimiento a lo largo de la vida útil de una turbina llegan al menos a un 10-15 % de los ingresos totales de la generación [6].

Las turbinas de viento trabajan bajo condiciones ambientales muy duras, la temperatura, la presión y la humedad están en constante cambio, por lo que los tipos y formas de fallas son muchas y variadas, adicionalmente el acceso a la maquinaria de una turbina eólica puede llegar a ser muy complicado y extremadamente costoso en especial para turbinas de alta potencia. Por estas razones es indispensable un sistema de detección automática de fallas lo suficientemente fiable que permita evitar daños catastróficos (se hace referencia a daños que afecten de forma permanente la operación de la turbina) y permita planear de forma eficiente los procesos de mantenimiento. Las turbinas de viento son un excelente escenario para investigar el diseño de sistemas FDI con enfoque data-driven. A continuación se presentan dos modelos de turbina de viento sobre los cuales se desarrollaron y evaluaron los sistemas FDI construidos.

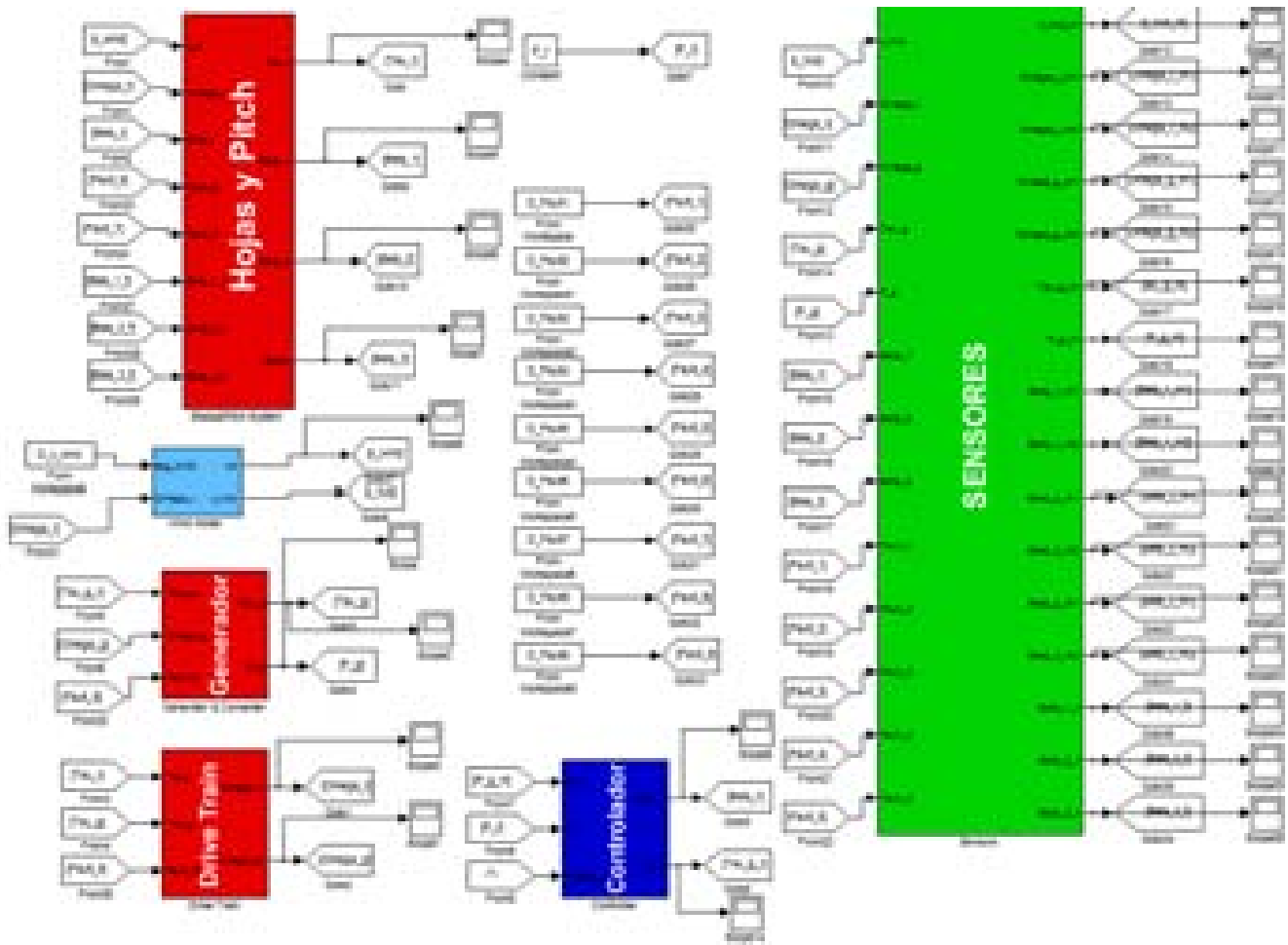
## 4.1. Modelo 1: Benchmark de turbina de viento

El modelo de la turbina de viento fue propuesto en su versión inicial en [1] como inicio de una competencia que buscaba un sistema de detección y diagnóstico de fallas de alto rendimiento. El modelo está construido en simulink (MATLAB) como se muestra en la Figura 4.1.

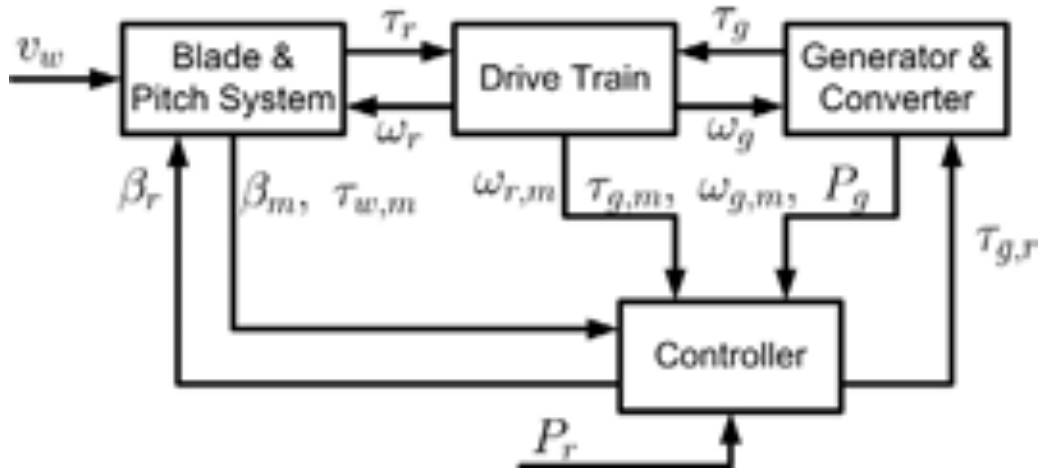
### 4.1.1. Descripción general del sistema

En este modelo se considera una turbina de tres hojas de eje horizontal de velocidad variable con convertidor completo. Funcionalmente el viento hace girar las hojas de la turbina de viento, la conversión de energía cinética en energía mecánica puede ser controlada cambiando la aerodinámica de la turbina girando las hojas mediante el sistema de pitch's o mediante el control de la velocidad de rotación. La energía mecánica se convierte en energía eléctrica mediante un generador totalmente acoplado a un convertidor. Entre el rotor y el generador una transmisión se utiliza para aumentar la velocidad de rotación final que llega al generador. El objetivo del sistema de control es seguir la potencia de referencia, o si no es posible minimizar el error de referencia.

En la Figura 4.2 se puede ver en diagrama de bloques las relaciones entre los componentes del sistema: Hojas-sistema de Pitch, Transmisión, Generador y Controlador.



**Figura 4.1** Modelo de turbina de viento propuesto en [1]. Vista en Simulink.



**Figura 4.2** Interconexión del modelo

$v_w$  es la velocidad del viento que es capturada a través de las hojas. El movimiento se propaga al rotor con velocidad angular  $w_r$  y torque  $T_r$ . A la salida de la transmisión esta la velocidad de entrada al generador  $W_g$  (angular) y torque  $T_g$ . El sistema de control se vale de las señales medidas provenientes del sistema de monitoreo, las cuales son en últimas las disponibles en una turbina real. Estas señales son:

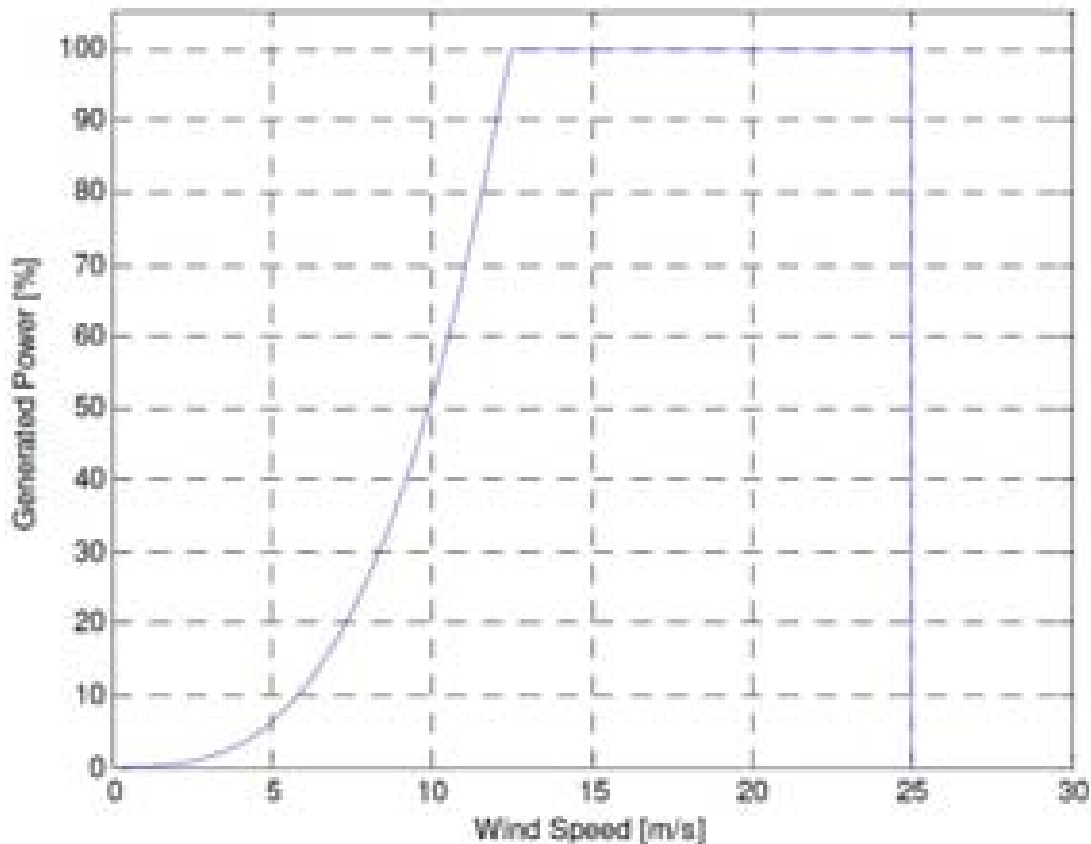
- $\beta_{1,m_1}, \beta_{1,m_2}, \beta_{2,m_1}, \beta_{2,m_2}, \beta_{3,m_1}, \beta_{3,m_2}$  Dos mediciones ( $m_1, m_2$ ) para cada uno de las tres pitch.
- $w_{r,m_1}, w_{r,m_2}$  Dos mediciones para la velocidad del rotor.
- $w_{g,m_1}, w_{g,m_2}$  Dos mediciones para la velocidad del generador.
- $\tau_{g,m}$  Torque en el generador.
- $P_{g,m}$  Potencia en el generador.

Como se puede ver, se cuenta con redundancia física dado que la mayoría de las señales se miden con dos sensores. Las salidas del sistema de control son las referencias de cada uno de los pitch  $\beta_{r_1}, \beta_{r_2}, \beta_{r_3}$  así como el torque en el generador  $\tau_{g,r}$ .

#### 4.1.2. Estrategia de Control

El controlador opera en principio en 4 zonas. La Zona 1 corresponde a la puesta en marcha de la turbina, la Zona 2 corresponde a optimización de potencia, Zona 3 es producción de potencia





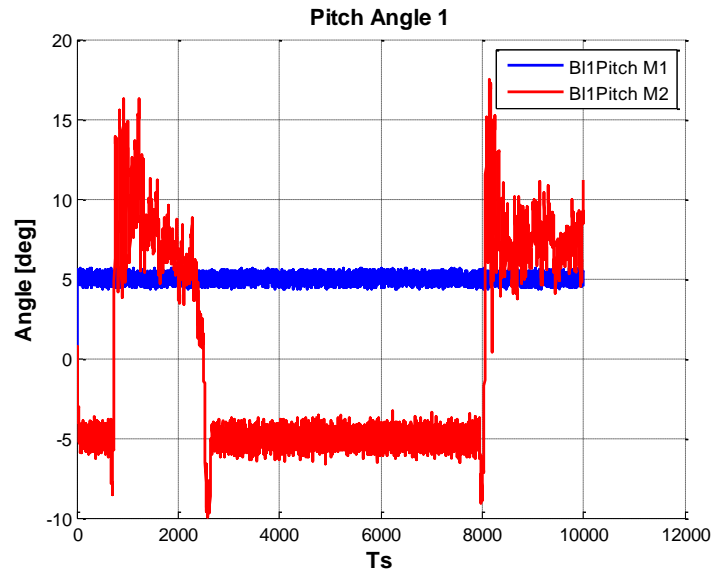
**Figura 4.3** Curva de potencia turbina de viento

constante y la Zona 4 son velocidades de viento altas. La operación normal de una turbina se realiza en las zonas 2 y 3 [92].

En la Figura 4.3 se muestra la curva de potencia de la turbina. De esta figura se puede observar que para vientos entre 0 y 12,5 m/s la turbina es controlada para lograr una producción de potencia óptima (Zona 2). Por encima de esta velocidad la turbina opera en su valor de potencia referencia  $P_r$  (Zona 3) y el objetivo del sistema de control es mantener este valor constante evitando sobrecargas estructurales en el sistema. Esto se logra mediante las señales de control  $\beta_r$ . En ambientes industriales se usan controladores PI para realizar esta tarea [1].

### 4.1.3. Escenarios de falla

En este modelo se consideran diferentes tipos de fallas enfocándose en las de mayor ocurrencia en las turbinas modernas. Adicionalmente establece un conjunto de objetivos (tiempos de detección, número de falsas alarmas y tiempo de recuperación) que debe cumplir el sistema de detección y diagnóstico que se diseñe. A continuación se hace una breve introducción a los escenarios de fa-



**Figura 4.4** . Falla en los sensores de valor fijo

llas.

#### **Fallas y requerimientos de detección**

Las Fallas consideradas para sensores se listan en la Tabla 4.2, y se pueden dividir en dos tipos aquellas de valor fijo y las de valor escalado.

El comportamiento de una falla de valor fijo es como la mostrada en la Figura 4.4 donde en este caso el sensor 1 del pitch 1 queda en un valor constante de 5 grados. Cabe resaltar que aunque el sensor se encuentra en un valor constante esta señal está contaminada por niveles de ruido hasta de un 20 % en magnitud de la señal. Estas magnitudes de ruido son típicas bajo este escenario de falla [1].

Los valores que pueden tener los sensores de los pitch cuando presentan este tipo de falla (valor de la constante  $K$  en la Tabla 4.2) varían dentro del rango de 0 y 10 grados [1]. El modelo establece dos valores  $K = 10$  (Falla 3) y  $K = 5$  (Falla 1), sin embargo para el trabajo realizado en esta tesis múltiples valores fijos entre los rangos establecidos fueron evaluados.

Por otro lado están las fallas de valor escalado como las presentadas en la Figura 4.5. La grafica superior muestra el valor escalado en el sensor 1 del generador (Falla 5) por un valor de  $K = 0,9$ . La grafica inferior muestra el mismo tipo de falla pero esta vez para el pitch 2 (Falla 2), en esta se evidencia una zona donde el factor de escala se confunde con ruido. Este tipo de Falla es especialmente difícil de identificar para magnitudes de los ángulos de los pitch muy cercanos a cero, un análisis de esta problemática se realizará en el siguiente capítulo.

Finalmente como requerimiento de rendimiento el modelo estable que las fallas en los sensores sean detectadas en máximo  $10T_s$  (10 periodos de muestreo).

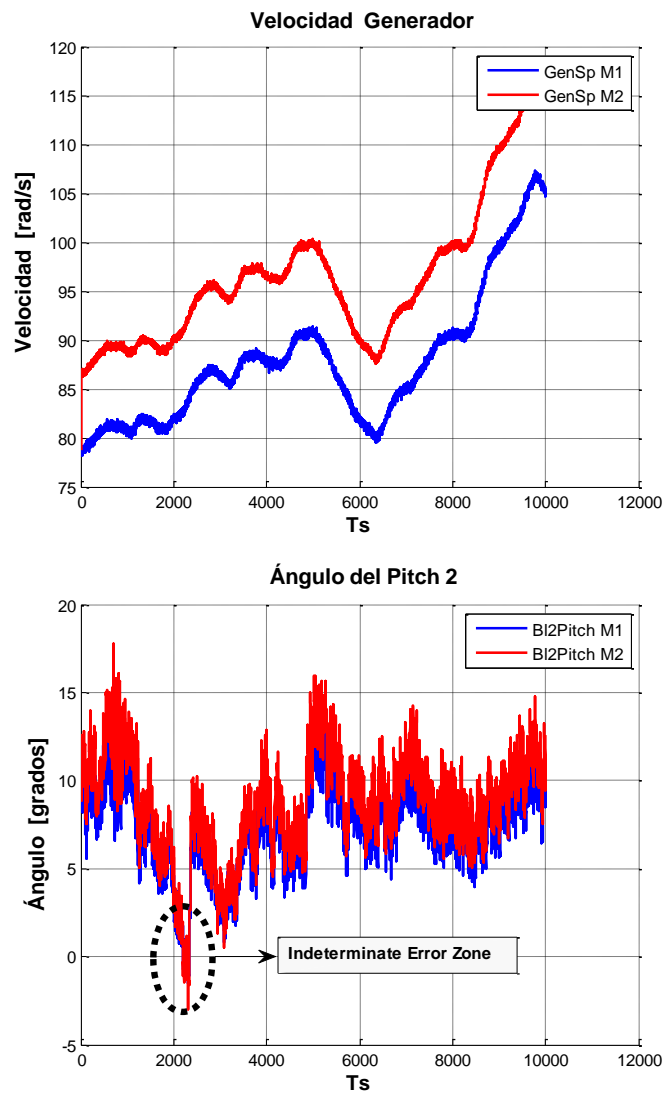


Figura 4.5 . Falla en los sensores de valor escalado

**Tabla 4.2** Fallas en los sensores consideradas

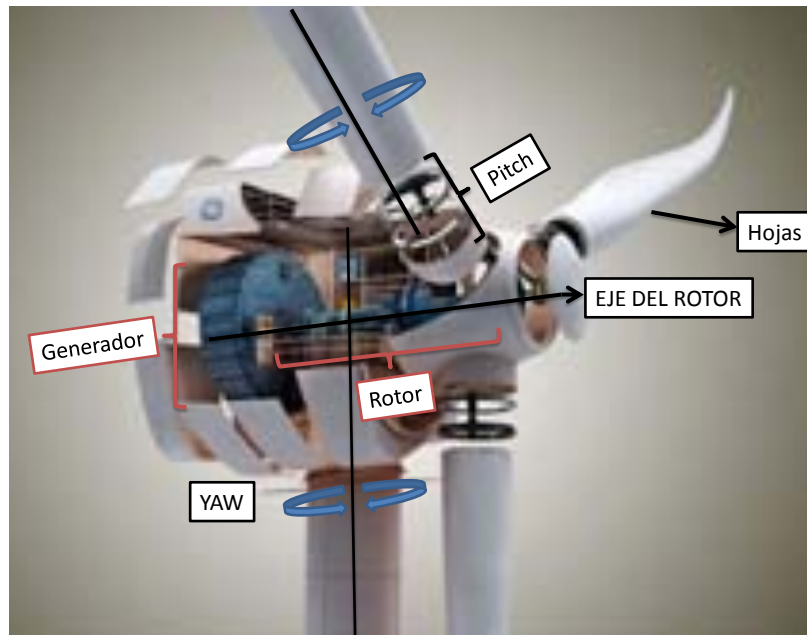
No	Falla	Valores Estandar del Modelo	Descripción
1	$\beta_{1,m_1} = K$	$K = 5$ (grados)	Valor Fijo en el sensor del pitch 1
2	$\beta_{2,m_2} = K\beta_{2,m_2}$	$K = 1,2$	Valor escalado en sensor del pitch 2
3	$\beta_{3,m_1} = K$	$K = 10$ (grados)	Valor Fijo en el sensor del pitch 3
4	$w_{r,m_1} = K$	$K = 1,4$ (rad/s)	Valor Fijo en el sensor de velocidad del rotor
5	$w_{r,m_1} = K_1 w_{r,m_1}$ y $w_{g,m_1} = K_2 w_{g,m_1}$	$K_1 = 1,1$ y $K_2 = 0,9$	Simultaneamente valores escalados los sensores de velocidad del rotor y el generador.
$K, K_1$ y $K_2$ constantes.			

Las fallas en los actuadores se muestran en la Tabla 4.3. La Falla 6 es un fallo hidráulico en el actuador del pitch 2 que involucra un cambio abrupto en los parámetros del modelo. La Falla 7 hace referencia a un incremento en el contenido de aire en el aceite del actuador del pitch 3 lo que genera una respuesta lenta del actuador. Esta falla es de carácter incipiente y se desarrolla de forma lenta en el tiempo. Los tiempos de detección para las Fallas 6 y 7 son de  $8T_s$  y  $600T_s$  respectivamente. Adicionalmente se tiene un falla del sistema (8) que involucra un desplazamiento en el lazo de control interno del generador, como consecuencia se tiene un bias en el torque del generador  $\tau_g = \tau_g + K$  donde  $K$  es una constante y varía entre (0 y 100 Nm). Esta Falla debe ser detectada en  $5T_s$ . Finalmente se tiene la Falla 9 que es un incremento en la fricción de la transmisión. Esta falla normalmente se desarrolla de forma lenta en el tiempo y como consecuencia se evidencia un aumento en las vibraciones del sistema. El modelo deja como desafío la detección de esta falla y no establece ningún tiempo de detección.

**Tabla 4.3** Fallas en los actuadores consideradas

Falla No	Descripción
6	Falla en el actuador del pitch 2. Falla Hidráulica
7	Falla en el actuador del pitch 3. Incremento de aire
8	Falla en el lazo interno de control del generador
9	Incremento en la fricción de la transmisión producto del desgaste

Las Fallas en los actuadores son mucho más desafiantes que las fallas en los sensores, además estas fallas de no ser detectadas a tiempo podrían causar daños severos en la turbina. Como se muestra en [7] las fallas en los actuadores consideradas (en sus diversas tipologías) ocurren en 9 de cada 10 turbinas durante su vida útil y en un 45% de los casos su no detección causo la



**Figura 4.6 .** Componentes del modelo de turbina simulado en FAST

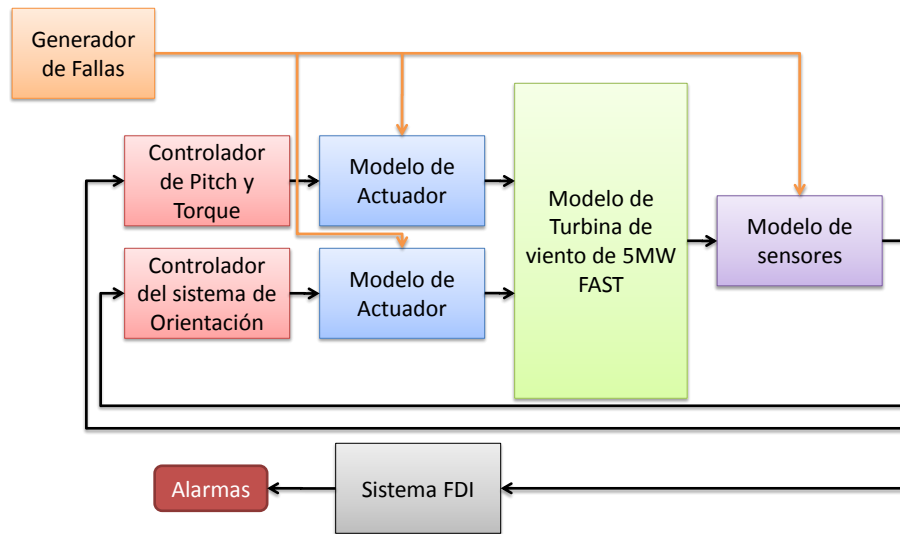
desconexión de la turbina.

## 4.2. Modelo 2: FAST

FAST es un simulador aerolástico diseñado por el NREL (U.S. National Renewable Energy Laboratory's) para el estudio y diseño de sistemas de control. Diferentes modelos de turbinas de tamaños variables están disponibles para uso público, en este trabajo se usó un modelo de 5MW de 3 hojas de velocidad variable con sistema de pitch de giro completo. Para facilitar su implementación se usó su enlace con Simulink tal como se describe en [93], modificaciones adicionales se realizaron para simular las Fallas del benchmark y así realizar una comparación objetiva.

### 4.2.1. Características Generales

La altura de la turbina de eje es 90 m, el radio del rotor es de 63 m y su velocidad nominal es de 12.1 rpm. A diferencia del modelo anterior en FAST el movimiento de los pitch está limitado a 8 deg/s., adicionalmente se poseen modelos de viento son mucho más especializados generados a partir de TurbSIM [94] el cual tiene en cuenta cambios atmosféricos y componentes de velocidad en 3 dimensiones. Se incorpora adicionalmente un sistema de orientación 'YAW System' que le permite a la góndola de la turbina girar tal y como se especifica en la Figura 4.6.



**Figura 4.7** Componentes de la interfaz de simulink construida con FAST

En la Figura 4.7 se muestran los componentes de la interfaz en Simulink construida. Las Fallas se logran corrompiendo las señales de los actuadores y sensores sin modificar el FAST (esto garantiza una respuesta limpia y realista del modelo).

### 4.2.2. Sensores

En este modelo los sensores se modelaron en Simulink añadiendo bloques de Ruido Blando de banda limitada a las variables disponibles en el modelo FAST. Estos bloques de ruido blanco simulan el ruido que se presenta en el sistema de monitoreo, y están caracterizadas por el peso de la PSD (Power Spectral Density) 'Noise Power'. Para hacer los modelos de ruido lo más realistas posibles se siguieron las recomendaciones dadas en [95] y las especificaciones para los sensores del modelo anteriormente explicado.

### 4.2.3. Actuadores

En FAST se consideran 3 subsistemas de actuadores: Fpitch, generador-convertidor y el que controla el sistema de orientación (YAW system). La descripción técnica detallada de cada uno de estos sistemas se puede encontrar en [96]. La estrategia de control empleada es la misma explicada en el modelo anterior, el sistema de orientación funciona solo en zona de optimización de potencia.

**Tabla 4.4** Señales medidas (sensores) seleccionadas en el modelo de turbina FAST

Sensor	Simbolo	Unidades	Potencia de Ruido
<i>Velocidad del Viento (Anemómetro)</i>	$v_{w,m}$	m/s	0,0071
<i>Velocidad de Rotor</i>	$w_{r,m}$	rad/s	$10^{-4}$
<i>Velocidad del Generador</i>	$w_{g,m}$	rad/s	$2 \times 10^{-4}$
<i>Torque del Generador</i>	$\tau_{g,m}$	Nm	0,9
<i>Potencia Electrica Generada</i>	$P_{g,m}$	W	10
<i>Angulo de la i-esima hoja</i>	$\beta_{i,m}$	grados	$1,5 \times 10^{-3}$
<i>Error del YAW</i>	$Y_{e,m}$	grados	$5 \times 10^{-2}$

#### 4.2.4. Escenarios de Falla

Al igual que en el modelo anterior se consideran fallas tanto en los sensores como en los actuadores. En la Tabla 4.5 se muestran los tipos de Fallas considerados:

**Tabla 4.5** Fallas consideradas en el modelo FAST

No.	Falla	Valor estandar del modelo	Tipo
1	$\beta_{1,m} = K\beta_{1,m}$	$K = 0,95$	Valor escalado
2	$v_{w,m} = v_{w,m} + K$	$K = -0,5$	Offset
3	$w_{g,m} = Kw_{g,m}$	$K = 0,95$	Valor escalado
4	$\beta_{1,m} = K$	$K = 1$	Valor fijo
5	$P_{g,m} = KP_{g,m}$	$K = 1,1$	Valor escalado
6	Falla hidráulica en el pitch 2	-	Cambio abrupto en la dinámica del pitch
7	Incremento en el contenido de aire del pitch 3	-	Cambio incipiente en la dinamica pitch
8	Falla en el lazo interno de control del generador	-	Offset en el torque generado
9	Sin movimiento en el sistema de desplazamiento	-	Bloqueo del sistema

**Fallas en los sensores:** Existen tres fallas de valor escalado (Falla 1, Falla 3 y Falla5), una de valor fijo (Falla 4) y 1 de desplazamiento (Falla 2). A diferencia de una falla de valor escalado que actúa como una constante multiplicando una señal, la Falla de desplazamiento actúa como una constante sumada a una señal. La Falla de desplazamiento es bien conocida en los acelerómetros y

ha sido objeto de diferentes estudios [97].

**Fallas en los actuadores:** La Falla 6 y 7 son fallas en los pitch, al igual que en el modelo anterior hace referencia a un daño hidráulico de carácter abrupto y a contenido de aire en aceite que es de carácter incipiente. La Falla 8 involucra un desplazamiento en el lazo de control interno del generador, como consecuencia se tiene un desplazamiento en el torque del generador  $t_g = t_g + K$  donde K es una constante y tiene como valor máximo 1000 Nm. Finalmente la Falla 9 corresponde a un valor fijo de 0 rad/s en la señal de control del sistema de orientación.

- Tiempo de detección menor a 10 Ts para todas las fallas de sensores.
- Tiempo de detección menor a 8 Ts para la Falla 6.
- Tiempo de detección menor a 100 Ts para la Falla 7.
- Tiempo de detección menor a 3 Ts para la Falla 8.
- Tiempo de detección menor a 50Ts para la Falla 9.

### 4.3. Retos de los Modelos

Existen dos problemas fundamentales a la hora de diseñar los sistemas de detección y aislamiento en estos modelos. En primer lugar los altos niveles de ruido que presentan las medidas, en especial el viento. En segundo, las no linealidades en el modelamiento de la aerodinámica de la turbina, tal y como se establece en [98] (en especial para el modelo FAST).

Como se verá en el siguiente capítulo algunos de los algoritmos de clasificación empleados lidian por si mismos con problemas de no linealidad y ruido, lo que constituye una ventaja adicional sobre los métodos basados en el modelo.



# Capítulo 5

## Construcción del sistema FDI mediante clasificadores

En este capítulo se presenta cada uno de los pasos realizados en la construcción del sistema FDI, desde el análisis de los datos hasta su construcción y validación. Siguiendo una metodología propia de minería de datos se inicia con un análisis exploratorio del conjunto de datos con el objeto de determinar la condición y características especiales de las señales involucradas.

### 5.1. Modelo 1: Benchmark

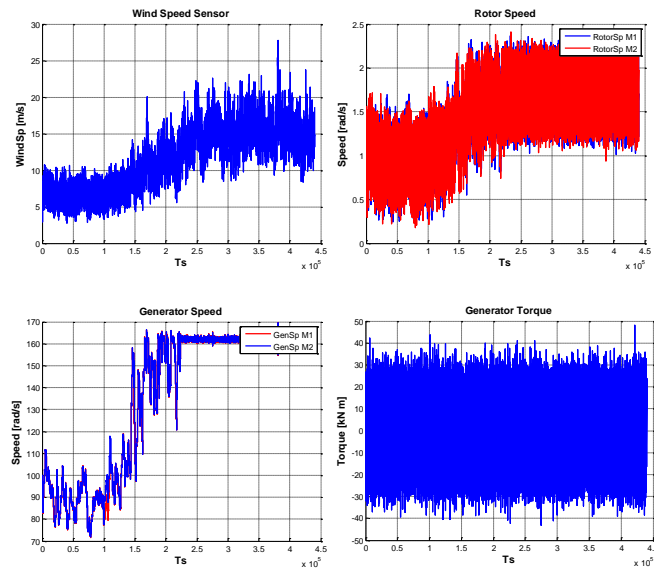
#### 5.1.1. Análisis exploratorio de los datos

Una simulación del modelo descrito en el capítulo anterior con la totalidad de las fallas tiene una duración 4400 seg. Cada 0.01s se guarda registro de las medidas del grupo de sensores de la turbina, estas constituyen nuestro conjunto inicial de características y se listan en la Tabla 5.1 (con la notación que se usará).

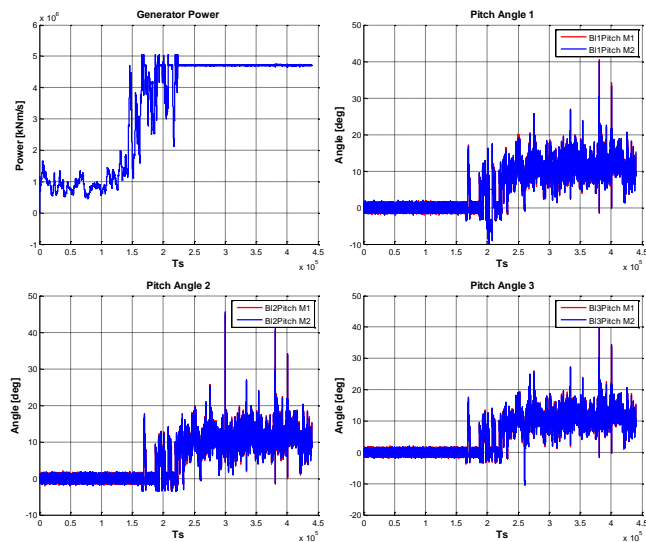
Un factor determinante es el nivel de ruido de cada una de estas señales, que es alto si se tiene en cuenta el ambiente (movimiento, clima, campo magnético, etc.) en el que trabajan estos dispositivos. En las Figuras 5.1 y 5.2 se grafican cada una de las características a lo largo de una simulación, ilustrando el efecto del ruido sobre las medidas.

Como se puede observar la potencia generada es proporcional a la velocidad del generador, su valor nominal de 165 rad/s. Una velocidad mayor no es permitida por el sistema de control. Dado que la mayoría de registros están en el rango de operación nominal (4.8 MW), se deduce que en gran parte de la simulación el controlador opera en zona 3, zona en la cual los pitch entran en movimiento.

En la Tabla 5.2 se presenta la matriz de correlación (la matriz de correlación  $R$  es una matriz cuadrada constituida por los coeficientes de correlación  $r$  de cada pareja de variables) entre los



**Figura 5.1** Grafica de las características: WindSp, RotorSp M1, RotorSp M2, GenSp M1, GenSp M2, y GenTorq.



**Figura 5.2** Grafica de las características: GenPower, B1Pitch M1, B1Pitch M2, B2Pitch M1, B2Pitch M2, B3Pitch M1 y B3Pitch M2.

**Tabla 5.1** Conjunto inicial de atributos del set de entrenamiento

<b>Atributo / Característica</b>	<b>Notación</b>
<i>Medida de velocidad de viento</i>	WindSp [m/s]
<i>Medida 1 de la velocidad del rotor</i>	RotorSp M1 [rad/s]
<i>Medida 2 de la velocidad del rotor</i>	RotorSp M2 [rad/s]
<i>Medida 1 de la velocidad del generador</i>	GenSp M1 [rad/s]
<i>Medida 2 de la velocidad del generador</i>	GenSp M2 [rad/s]
<i>Medida del torque del generador</i>	GenTorq [N m]
<i>Medida de potencia del generador</i>	GenPower [Nm/s]
<i>Ángulo del Pitch 1 Medida 1</i>	B11Pitch M1 [deg]
<i>Ángulo del Pitch 1 Medida 2</i>	B11Pitch M2 [deg]
<i>Ángulo del Pitch 2 Medida 1</i>	B12Pitch M1 [deg]
<i>Ángulo del Pitch 2 Medida 2</i>	B12Pitch M2 [deg]
<i>Ángulo del Pitch 3 Medida 1</i>	B13Pitch M1 [deg]
<i>Ángulo del Pitch 3 Medida 2</i>	B13Pitch M2 [deg]

diferentes atributos. Formalmente la correlación estadística determina la relación o dependencia que existe entre las dos variables que intervienen en una distribución bidimensional su valor varía entre  $-1$  y  $1$ .

- Si el coeficiente de correlación lineal toma valores cercanos a  $-1$  la correlación es fuerte e inversa, y será tanto más fuerte cuanto más se aproxime  $r$  a  $-1$ .
- Si el coeficiente de correlación lineal toma valores cercanos a  $1$  la correlación es fuerte y directa, y será tanto más fuerte cuanto más se aproxime  $r$  a  $1$ .
- Si el coeficiente de correlación lineal toma valores cercanos a  $0$ , la correlación es débil.

Tabla 5.2 Matriz de Correlación

	WindSp	RotorSp MI	RotorSp M2	Gensp MI	Gensp M2	GenTörq	GenPower	BI1Pitch MI	BI1Pitch M2	BI2Pitch MI	BI2Pitch M2	BI3Pitch MI	BI3Pitch M2
WindSp	1	0.7898165	0.79010678	0.87202172	0.87216864	0.00265638	0.87585309	0.92114859	0.9023768	0.91625703	0.91608305	0.91462724	0.91509908
RotorSp MI		1	0.81955933	0.90542486	0.90464261	0.00112599	0.90108326	0.70800803	0.68601127	0.69948256	0.70124181	0.70302413	0.69627725
RotorSp M2			1	0.90521696	0.90594515	0.00117382	0.90203145	0.70880047	0.68648269	0.70020345	0.70181764	0.7037759	0.69683868
Gensp MI				1	0.99917979	0.00139821	0.99623452	0.78183326	0.75737677	0.77236537	0.77421254	0.77632233	0.76870016
Gensp M2					1	0.00133314	0.99573749	0.78222159	0.75786022	0.77280439	0.77464949	0.77676592	0.76913261
GenTörq						1	0.00126347	0.00220559	0.00195295	0.00237386	0.00233857	0.00206337	0.00223197
GenPower							1	0.79249982	0.76900772	0.78354339	0.7854322	0.78762825	0.77981443
BI1Pitch MI								1	0.97097517	0.98687802	0.9868299	0.98461822	0.98514779
BI1Pitch M2									1	0.98743438	0.98743899	0.98539235	0.98560816
BI2Pitch MI										1	0.99308682	0.99004623	0.9904196
BI2Pitch M2											1	0.99005338	0.99040018
BI3Pitch MI												1	0.98458675
BI3Pitch M2													1

Alta Correlación

Media Correlación

Baja Correlación

En general, en todos los casos la correlación es directa (positiva) sin embargo varía en magnitud, en la Tabla 5.2 se clasificó en baja, media y alta correlación. Algunas conclusiones básicas se logran deducir de esta matriz:

- El viento es la señal que controla todo el funcionamiento del sistema, su correlación es alta para casi todas las señales.
- Tal como se esperaba las mediciones de los pitch poseen una correlación directa cercana a 1, dado que la señal de control es igual para todos.
- La medida del torque en el generador presenta bajos niveles de correlación, dado en parte por los valores negativos que presenta (indicando un cambio de dirección) y sus niveles de ruido. En términos estadísticos y para la aplicación de los algoritmos de clasificación esta característica es candidata a ser eliminada del set de datos.

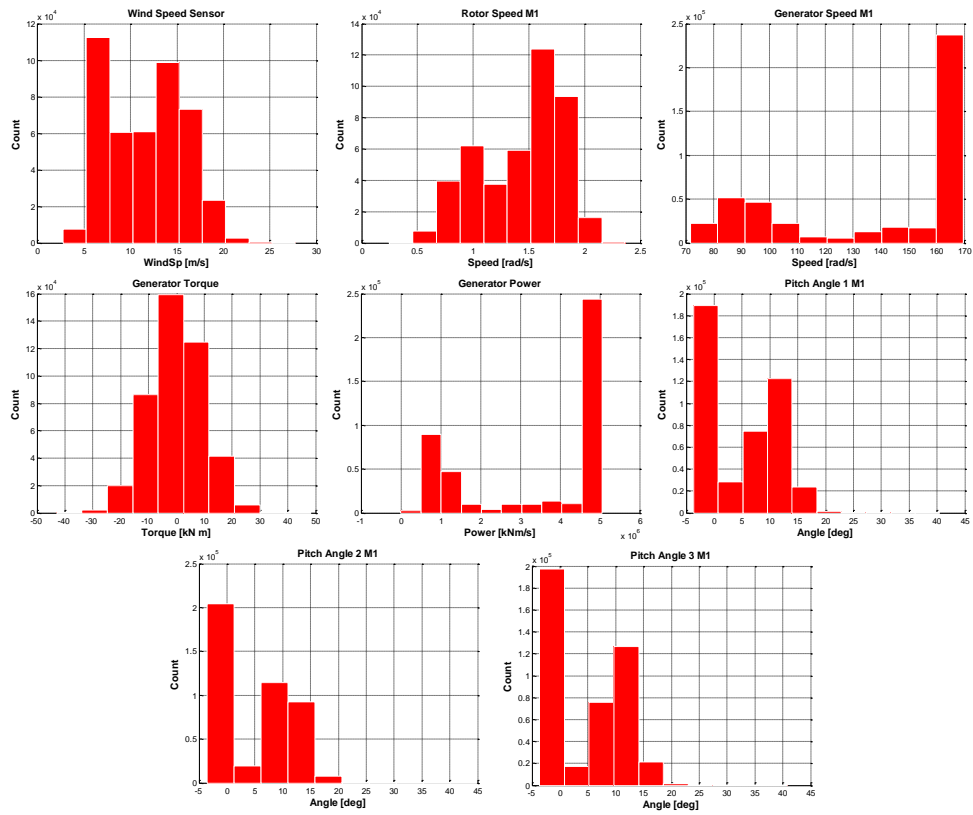
### **Histogramas - Graficas de caja y eliminación de Outliers**

Los histogramas y diagramas de caja nos permiten identificar valores anormales 'outliers' presentes en el set de datos. Formalmente un outlier se define como un dato que está muy lejos de la conducta normal de una variable o de una población [99, 100], estos pueden tener efectos nocivos en los análisis estadísticos, ya que generalmente provocan un aumento en la varianza del error y puede llevar al analista a conclusiones erróneas. Para efectos de tareas de clasificación los outliers influyen de forma negativa en los procesos de entrenamiento afectando las propiedades de separabilidad y generalización del mismo.

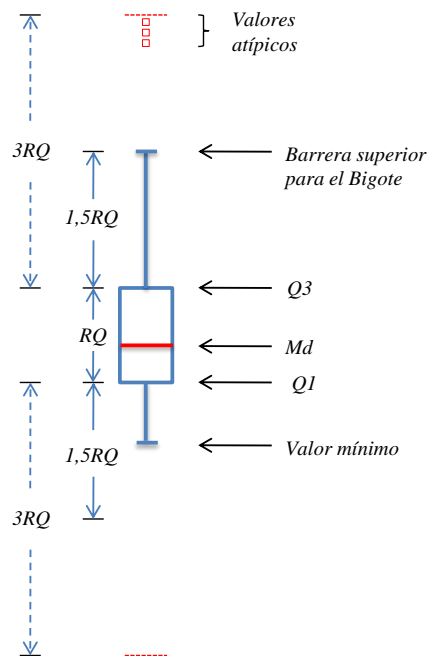
Es importante aclarar que los análisis de eliminación de outliers se realizaron sobre los registros de comportamiento normal, dado que en sí mismos los registros de falla son comportamientos anómalos y no deben ser eliminados del set de datos.

Un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. En el eje vertical se representan las frecuencias, y en el eje horizontal los valores de las variables. En términos matemáticos, puede ser definida como una función inyectiva (o mapeo) que acumula (cuenta) las observaciones que pertenecen a cada subintervalo de una partición. La Figura 5.3 muestra los histogramas para cada características discretizadas a 10 bins (grupos).

Los diagramas de caja por su parte, son un gráfico representativo de las distribuciones de un conjunto de datos en cuya construcción se usan cinco medidas descriptivas de los mismos, a saber: mediana, primer cuartil, tercer cuartil, valor máximo y valor mínimo. Un cuartil, divide la distribución en cuatro partes (corresponden a los cuantiles 0.25, 0.50 y 0.75), por ejemplo el primer cuartil indica que 25 % de las observaciones tiene un valor menor y 75 % un valor mayor. Los percentiles por su parte dividen en cien partes iguales el conjunto de datos ordenados. Ejemplo, el percentil de orden 15 deja por debajo al 15 % de las observaciones, y por encima queda el 85 %.



**Figura 5.3** Histogramas a 10 bins de cada una de las características en el set de datos.



**Figura 5.4** Estructura de un Diagrama de Caja.

La presentación de estas cinco medidas en el diagrama de caja permite al mismo tiempo deducir información sobre la tendencia central, dispersión y simetría de los datos de estudio. Adicionalmente permite identificar con claridad y de forma individual, observaciones que se alejan de manera poco usual del resto de los datos.

La forma de un diagrama de caja es como el presentado en la Figura 5.4:

- El rectángulo delimita el rango intercuartil con la arista izquierda (o inferior) ubicada en el primer cuartil  $Q_1$ , y la arista derecha (o superior) en el tercer cuartil  $Q_3$ .
- Se dibuja una línea a través del rectángulo en la posición que corresponde al segundo cuartil (que es igual al percentil 50 o a la mediana),  $Q_2 = M_d$ .
- De cualquiera de las aristas del rectángulo se extiende una línea, o bigote, que va hacia los valores extremos (valor mínimo y valor máximo). Estas son observaciones que se encuentran entre cero y 1.5 veces el rango intercuartil a partir de las aristas del rectángulo.
- Las observaciones que están entre 1.5 y 3 veces el rango intercuartil a partir de las aristas del rectángulo reciben son valores potencialmente atípicos. Las observaciones que están más allá de tres veces el rango intercuartil a partir de las aristas del rectángulo se conocen como valores atípicos extremos.

Los diagramas de caja también se conocen con el nombre de diagramas de caja y bigotes. Nótese que el rectángulo o caja representa el 50% de los datos que particularmente están ubicados en la zona central de la distribución. La caja representa el cuerpo de la distribución y los bigotes sus colas.

Las gráficas de caja correspondientes al conjunto de atributos del set de datos se muestran en la Figura 5.5.

A través de los análisis de estas gráficas se obtuvieron un conjunto candidato de outliers:

- Valores del Torque en el generador de magnitud superior 34 Knm son producto del ruido. Dado el máximo valor de viento presente (27 m/s) no es posible obtener un torque de magnitud superior a 34 Knm para este modelo turbina.
- Valores de ángulo superiores a 40 grados como las presentadas en el Pitch 2 deben ser consideradas como outliers si se tiene en cuenta que ninguno de los otros pitch las presenta. Adicionalmente valores por encima de 27 grados son muy poco frecuentes (ver histograma) y se pueden considerar como valores atípicos. Tal y como se muestra en la Figura 5.6 estos valores se salen del comportamiento normal de la turbina a lo largo de la simulación.
- Al igual que el punto anterior, ángulos de pitch menores a -3 deg son outliers potenciales (ver Figura 5.6).

### 5.1.2. Pre procesamiento de los datos

**ELIMINACIÓN DE OUTLIERS:** Los registros que se identificaron en la sección anterior fueron retirados del set de datos. Indirectamente la eliminación de outliers conlleva una ventaja adicional y es la reducción del tamaño del conjunto de datos.

**MUESTREO:** El número de registros del comportamiento normal de la turbina, supera en un factor importante el número de registros de los diferentes tipos de Fallas (100.000), esto crea un desbalance en el momento de generar modelos de clasificación. Sumado a esto y dado el interés de esta investigación son mucho más importantes los registros pertenecientes a fallas.

Una forma de reducir el número de registros de comportamiento normal es a través del muestreo. En estadística se conoce como muestreo a la técnica para la selección de una muestra a partir de una población. Al elegir una muestra se espera conseguir que sus propiedades sean extrapolables a la población. Este proceso permite ahorrar recursos, y a la vez obtener resultados parecidos a los que se alcanzarían si se realizase un estudio de toda la población. La forma más simple de muestreo se conoce como muestreo aleatorio simple, en la cual cada elemento de la población tiene la misma probabilidad de ser seleccionado para integrar la muestra. A través de un proceso aleatorio sin reemplazo se realizó una reducción del 60% de los registros de comportamiento normal.



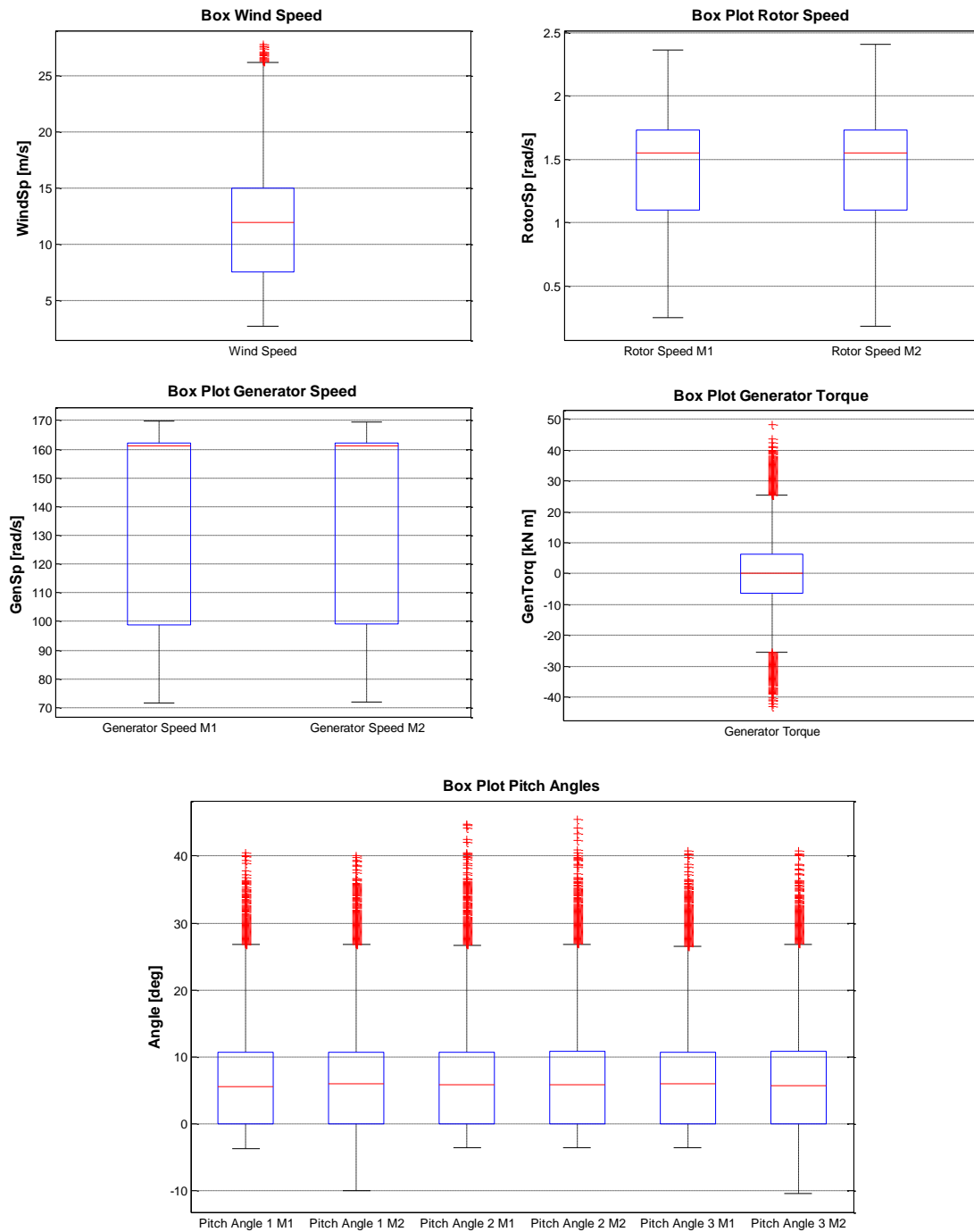


Figura 5.5 Graficas de Caja (Identificación de Outliers).

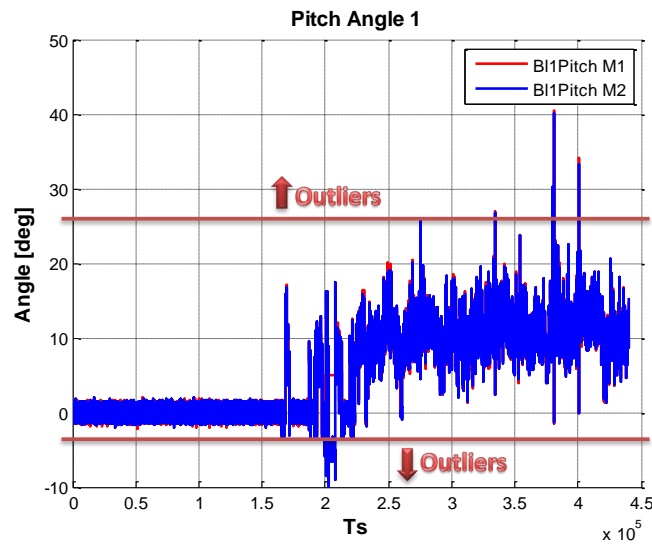


Figura 5.6 Outliers en las medidas de los pitch.

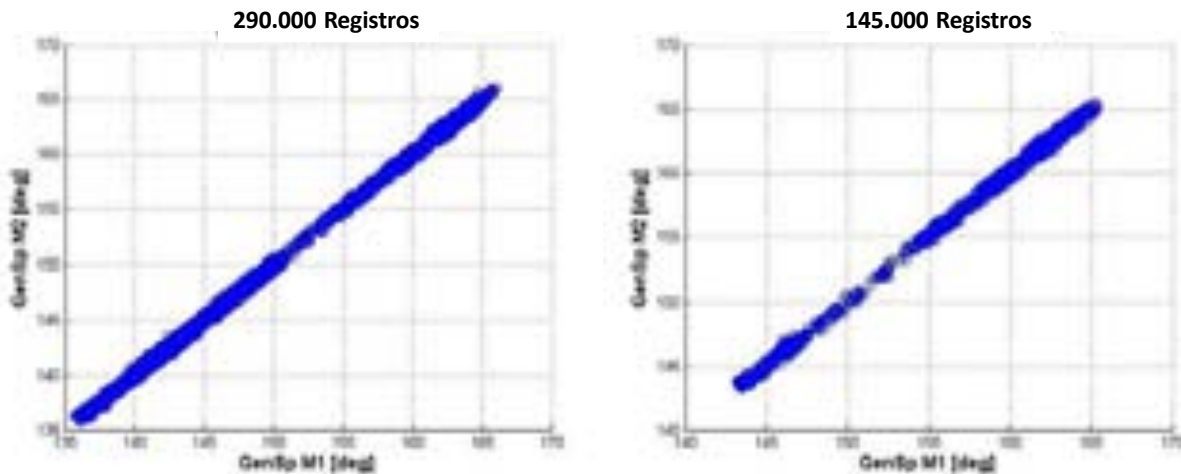


Figura 5.7 Efecto del Muestreo.

La Figura 5.7 muestra el efecto de esta técnica sobre el atributo velocidad del generador (GenSp), la grafica de la izquierda contienen 290.000 registros, la grafica de la derecha posterior al muestreo contiene 145.000 registros.

Para las técnicas de Redes Neuronales y SVM fue necesario realizar una reducción mayor para hacer más sensible el clasificador a las fallas.

**REDUCCIÓN DE CARACTERÍSTICAS:** Dado el conjunto de entrenamiento es recomendable reducir el conjunto de atributos para evitar la 'maldición de la dimensionalidad'. En minería

de datos y estadística se refieren a la maldición de la dimensión (también conocida como efecto Hughes) como el problema causado por el crecimiento exponencial del volumen asociado al incremento del número de dimensiones en un espacio matemático [75]. Como se vio en el capítulo anterior el sistema posee redundancia física, esto es una herramienta útil para la generación de residuos de los métodos basados en modelo [34], en los sistemas basados en datos y en especial en aquellos que usan sistemas clasificadores puede no ser necesario el empleo de medidas de redundancia como se demostrará en resultados posteriores. Esto no quiere decir que el empleo de medidas redundantes no mejore la calidad de los sistemas de clasificación, solo que en procura de obtener conclusiones generales independientes de las características particulares del sistema industrial sobre el cual se esté diseñando un sistema FDI, *se ha eliminado en esta investigación todo tipo de redundancia física.*

**Tabla 5.3** Conjunto de características posterior a la eliminación de redundancia física

<b>Atributo / Características</b>	<b>Notación</b>
<i>Medida de velocidad de viento</i>	WindSp [m/s]
<i>Velocidad del rotor medida 1</i>	RotorSp M1 [rad/s]
<i>Velocidad del generador medida 1</i>	GenSp M1 [rad/s]
<i>Medida del torque del generador</i>	GenTorq [N m]
<i>Medida de potencia del generador</i>	GenPower [Nm/s]
<i>Ángulo del Pitch 1 Medida 1</i>	B11Pitch M1 [deg]
<i>Ángulo del Pitch 2 Medida 1</i>	B12Pitch M1 [deg]
<i>Ángulo del Pitch 3 Medida 1</i>	B13Pitch M1 [deg]

Eliminar la redundancia física implicó además integrar las fallas a un solo sensor dentro del set de datos. Por ejemplo la Falla No 2 (Tabla 4.2), afectaba el sensor redundante ( $m_2$ ), lo registros en los cuales ocurrían esta falla se integraron al set de datos como medidas del sensor primario ( $m_1$ ). En la Tabla 5.3 se listan el set de características reducido.

**CREACION DE CARACTERISTICAS:** Desde un punto de vista de la minería de datos nuevos atributos pueden ser creados a partir de los atributos originales para mejorar la capacidad del clasificador y en algunas ocasiones para hacerlo más sensible a una clase en particular [76]. Por ejemplo, las fallas de valor fijo en los sensores podrían ser difíciles de detectar si estos se encuentran cerca de un valor en estado estacionario. Este problema puede ser resuelto creando un conjunto de atributos que tengan en cuenta la 'velocidad' de las medidas y no solo su valor. Se define la tasa de cambio del atributo  $x$  como:  $\Delta(x(k)) = |x(k) - x(k-1)|$ .

Donde  $x(k)$  es la señal en tiempo de muestreo  $k$ . Para señales lentas y niveles de ruido considerables la tasa de cambio definida podría no ser adecuada, sin embargo en la práctica esta medida fue suficiente. Tomar diferencias de tiempo mayores (Ej:  $k-10$ ) implica un costo adicional en alma-

cenamiento de información y procesamiento, sin embargo pueden ser mucho más adecuadas para los niveles de ruido presentes.

Se pueden agregar atributos más robustos que hagan evaluación de media o desviación sobre un conjunto más amplio de muestras, sin embargo se quiere evitar en lo posible realizar un procesamiento previo de los datos a la entrada del clasificador, pensando en mantener la simplicidad del sistema y su fácil implementación.

### 5.1.3. Aplicación de los algoritmos de clasificación

Resultado del análisis exploratorio y el preprocesamiento de los datos se procede a construir los clasificadores. En esta etapa del proceso ya se cuenta con set de entrenamiento robusto y adecuado para la aplicación de los algoritmos descritos en el Capítulo 3.

La tarea de los clasificadores es buscar un conjunto de patrones que le permitan identificar una clase en función de un conjunto de atributos. Cada registro del conjunto de datos de entrenamiento contiene un atributo nominal (label) que determina si existe un falla (Fault1,...,Fault9) o no (NoFault).

Como análisis preliminar cuatro técnicas de clasificación son empleadas KNN, redes bayesianas (BN), redes neuronales (NN) y máquinas de vectores de soporte (SVM), cada una de cuales fueron implementadas usando Matlab con el apoyo de Prtools para la creación de datasets. Para evaluar los modelos construidos desde el punto de vista de minería (predicción) se definen las siguientes métricas:

$$Precision = \frac{\text{Registros correctamente clasificados}}{\text{Numero de registros}} \times 100\% \quad (5.1)$$

La sensibilidad se puede definir para cada clase  $c_i$  como:

$$Sensibilidad_{c_i} = \frac{\text{Registros de la clase } c_i \text{ correctamente clasificados}}{\text{Numero de registros de la clase } c_i} \times 100\% \quad (5.2)$$

La sensibilidad global a Fallas se define como:

$$Sensibilidad = \frac{\text{Registros de falla correctamente clasificados}}{\text{Numero de registros de falla}} \times 100\% \quad (5.3)$$

Porcentaje de Falsas Alarmas:

$$Falsas Alarmas = \frac{\text{Registros de comportamiento normal clasificados como falla}}{\text{Numero de registros de comportamiento normal}} \times 100\% \quad (5.4)$$

## Diseño de los clasificadores

Todos los set de datos trabajados en el diseño y entrenamiento de los clasificadores recibieron el tratamiento explicado en secciones anteriores, cabe destacar adicionalmente que en todos los casos el número de registros de cada clase es igual.

## Arquitectura de la red neuronal

El desarrollo de la arquitectura involucra determinar cuántas neuronas va tener la red neuronal, cuantas capas y cómo van a ser interconectadas. Por su parte el entrenamiento se refiere al cálculo de los valores de los pesos de las conexiones [101]. Determinar la arquitectura de la red neuronal depende del problema a ser resuelto, el proceso de diseño generalmente es un proceso manual e iterativo, es decir, un experto tiene que experimentar sobre diferentes arquitecturas hasta encontrar una que retorne buenos resultados después de un proceso de entrenamiento.

La arquitectura de las redes neuronales utilizadas, así como el tiempo de entrenamiento empleado es presentada en la Tabla 5.4. Los resultados permiten concluir que las redes de base radial tiene una precisión más alta, sin embargo el consumo de recursos computacionales es más elevado.

**Tabla 5.4** Resultados de cada arquitectura de red neuronal empleada (CO capas ocultas, NxC Número de Neuronas por Capa, FA Función de Activación, TE Tiempo de Entrenamiento)

TIPO	CO	NxC	FA	TE	Precisión
<b>Multilayer</b>	2	10-5	Sigmoidal	3h 01m	70.09 %
<b>Multilayer</b>	3	10-5-5	Sigmoidal	3h 33m	70.63 %
<b>RBF (Función de base radial)</b>	1	10	Función Gaussiana	5h 28m	78.63 %

## Clasificadores Bayesianos

Para los clasificadores basados en teoría de inferencia bayesiana se trabajaron todas las clases como equiprobables, esto es cada clase tiene la misma probabilidad de ocurrencia. Hay varias implicaciones al realizar esta asunción:

- Un comportamiento normal es igual de probable que una falla. Si bien no es acorde a la realidad esta asunción permite darle una alta importancia a la ocurrencia de una falla.
- Se tiene un mayor número de falsas alarmas a costa de una mayor capacidad de detección (lo que se traduce en una mayor sensibilidad para las clases de falla y una menor precisión global del clasificador).

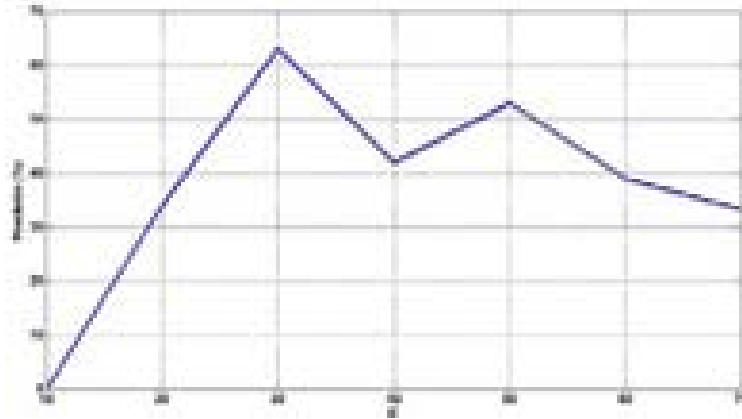


Figura 5.8 KNN encontrando el valor de  $k$ .

### KNN - K vecinos cercanos

Para encontrar el valor óptimo de  $k$  se hizo un barrido de valores calculando para cada caso la precisión del clasificador (Figura 5.8). Con valores entre  $k = 27$  y  $k = 37$  se logra maximizar a alrededor del 60 %.

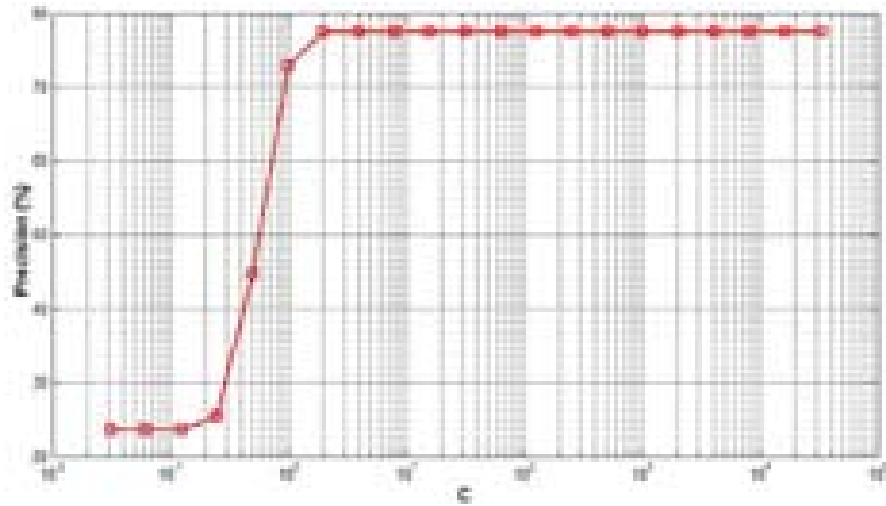
### SVM - Máquinas de Vectores de Soporte

Para las máquinas de vectores de soporte se usó un Kernel Gaussiano, los valores óptimos de  $C$  y  $\sigma$  se encontraron al igual que para el caso de KNN a través de la experimentación. Bajo las recomendaciones establecidas en [78] se fija el valor de  $\sigma$  en 10 y se hace un barrido de valores para  $C$  evaluando en cada iteración la precisión del clasificador (Figura 5.9).

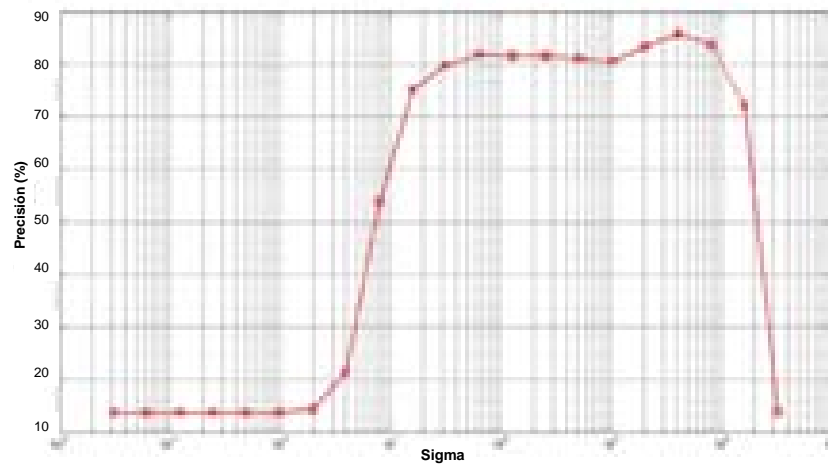
El valor de  $C$  que maximiza la precisión se encuentra con valores superiores a 10. Con el valor de  $C$  fijado se procede a realizar el barrido de  $\sigma$  (Figura 5.10) y encontrar su óptimo en 100 (no se toman valores cercanos a los codos de la gráfica para evitar posibles inestabilidades).

### Evaluación de los clasificadores

Una vez construidos los clasificadores se procede a evaluarlos sobre 10 conjuntos de entrenamiento construidos como una simulación completa del modelo de la turbina, esto implica que los registros de falla están en una cantidad muy inferior en comparación con los registros de comportamiento normal, tal y como se esperaría en un sistema industrial real, en estos casos las medidas de sensibilidad de cada clase juegan un papel muy importante [72]. La precisión global podría ser una medida viciada en el sentido que un clasificador que detecte todo como normal tendrá una precisión mayor que otros clasificadores, incluso si no detecta correctamente ninguna falla. Los



**Figura 5.9** SVM encontrando el valor de  $C$ .



**Figura 5.10** SVM encontrando el valor de  $\sigma$ .

resultados globales comparativos de los diferentes algoritmos empleados se muestran en la Tabla 5.5.



**Tabla 5.5** Evaluación de los clasificadores

	SENSIBILIDAD POR CLASE											Falsas Alarmas
	F1	F2	F3	F4	F5	F6	F7	F8	F9	Normal	Precisión	
KNN	93.20 %	70.12 %	85.23 %	68.21 %	95.12 %	41.12 %	57.12 %	0,00 %	47,78 %	71,75 %	65,42 %	28,25 %
NAIVE BAYES	91,70 %	85,80 %	84,20 %	82,60 %	93,20 %	65,90 %	62,80 %	11,30 %	60,90 %	83,90 %	80,72 %	16,1 %
NN RBF	82,60 %	84,00 %	82,70 %	76,10 %	87,80 %	67,20 %	62,50 %	24,60 %	57,80 %	78,60 %	77,43 %	21,4 %
SVM	86,70 %	88,80 %	80,20 %	85,60 %	98,20 %	68,90 %	67,80 %	1,30 %	61,90 %	76,90 %	74,89 %	23,1 %

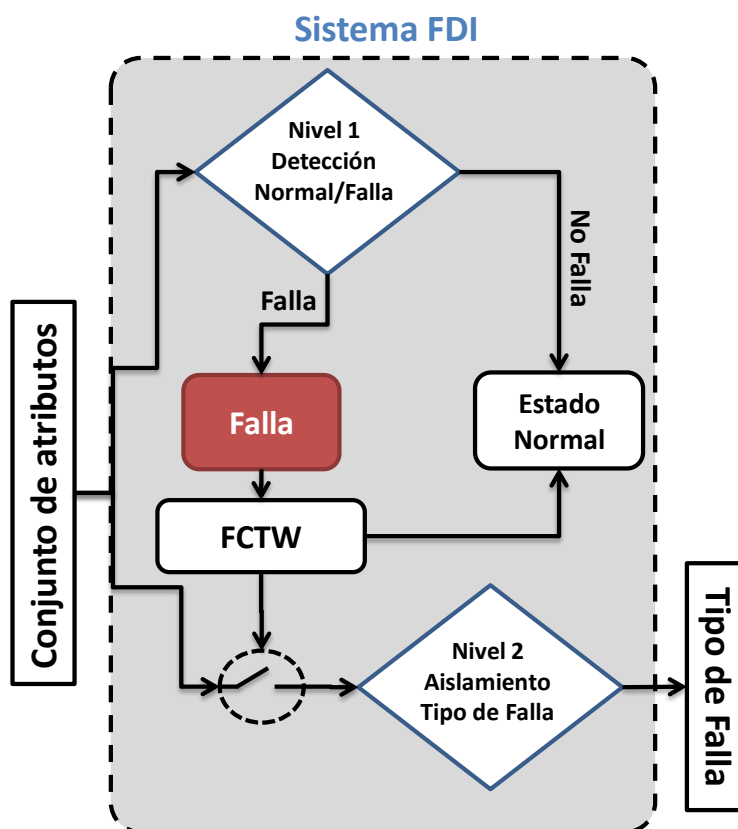
F1, F2 , F3 , F4, F5, F6 , F8 y F9 Abreviatura de Falla 1-9

Al aplicar los clasificadores directamente al conjunto de datos se está realizando dos procesos simultáneamente, la detección y el aislamiento, como consecuencia cuando es incorrecta la detección de la clase 'normal' el registro se clasifica en uno de las diferentes clases de fallas generando una falsa alarma. Esta es una consecuencia directa del esquema actual y se hace inviable para aplicarlo a un sistema industrial real (incumpliendo los requisitos establecidos en el capítulo 2). A pesar de esto un conjunto de conclusiones útiles se obtuvieron:

- El comportamiento abrupto de las fallas 1-5 hace que sean mucho más fáciles de detectar, los porcentajes de sensibilidad oscilan entre 70 %-94 %. Por su parte las fallas 6-9 presentan un comportamiento incipiente y son de difícil detección, la falla 8 en especial es un reto de detección para todos los clasificadores.
- Las peores sensibilidades se obtuvieron con KNN. Como se menciona en el capítulo 2 este resultado es un indicativo de no linealidad.
- NN y SVM tienen altas sensibilidades a las clases fallas a costa de una más alta tasa de falsas alarmas. La red neuronal de base radial tuvo la mayor sensibilidad para la Falla 8.
- En términos generales el clasificador probabilístico naive bayes presentó un mejor desempeño y alta sensibilidad a fallas. Este resultado invita a probar clasificadores probabilistas más elaborados como las GMM.

#### **5.1.4. Construcción del sistema FDI (Arquitectura a dos niveles)**

Se propone un sistema FDI a dos niveles, en cada uno de los cuales se tendrá un modelo de clasificación. En el primer Nivel el clasificador se encarga de detectar la falla diferenciándola de un comportamiento normal, en la construcción del set de datos para su entrenamiento solo se tendrán dos clases 'Falla' y 'Normal'. En este nivel reconocer un estado de operación de 'Falla' es más importante que reconocer uno 'Normal' por lo que se procura entrenar clasificador con una alta sensibilidad a esta clase a costa de la degradación parcial de la precisión general del mismo. Como consecuencia en este nivel se tienen un alto número de falsas alarmas, con el fin de reducirlas a la salida se construyó un contador de fallas sobre ventanas de tiempo (FCTW 'Fault Counter on Time Windows') propuesto en [102]. El objetivo del FCTW es verificar la persistencia de una falla, es decir para que se identifique un estado de 'Falla' a la salida del FCTW, el clasificador debe identificar el estado 'Falla' un número mínimo de veces durante un periodo de tiempo. El tamaño de la ventana de tiempo se puede determinar experimentalmente buscando reducir los falsos positivos pero se debe tener en cuenta que su tamaño influye directamente en el tiempo de detección [102]. Cabe resaltar que este enfoque no es adecuado para fallas intermitentes con periodos largos de ocurrencia.



**Figura 5.11** Arquitectura FDI a dos niveles.

En caso de detectarse el estado de 'Falla' en el Nivel 2 es el encargado de clasificar el tipo de falla del sistema. En este nivel el set de entrenamiento es construido con datos en estado de 'Falla' y las clases son todos los posibles tipos de Fallas. En resumen la estructura propuesta se describe en la Fig 5.11.

Para cada uno de los niveles se entrenaron 4 clasificadores de acuerdo a los algoritmos estudiados.

### Construcción Nivel 1

La construcción de los modelos de clasificación se realizó de la misma forma explicada en la sección anterior. Con el objeto de evaluar su desempeño se construyeron 50 set de pruebas a partir de simulaciones sobre el modelo de turbina. El desempeño promedio obtenido por cada algoritmo se resume en la Tabla 5.6.

La precisión está en el rango del [81% 87%] con sensibilidades superiores al 90% para SVM, NN y GMM. En este nivel estamos interesados en un modelo de clasificación con alta sensibilidad

**Tabla 5.6** Evaluación de los clasificadores Nivel 1

Algoritmo	Precisión (%)	Sensibilidad a Fallas (%)	Falsas Alarmas (%)
<b>SVM</b>	<b>87.45</b>	<b>92.17</b>	<b>10.01 %</b>
<b>NN</b>	<b>85.36</b>	<b>93.41</b>	<b>11.21 %</b>
KNN	84.35	80.14	13.21 %
GMM	81.35	90.54	15.16 %

los cuales generalmente también tienen un mayor número de falsas alarmas.

Es importante aclarar que pese a no obtener una precisión del 100% 'no' se presentaron fallas que no hayan sido detectadas. Una falla se evidencia a lo largo de un conjunto de registros secuenciales, la mala clasificación de algunos de estos registros no es un indicativo de la no detección global de la falla. Se define como criterio:

'Se establece la no detección de una falla si el 70% de los registros donde se evidencie su ocurrencia fueron clasificados de forma errónea como comportamiento normal'.

### FCTW 'Contador de Fallas sobre ventanas de tiempo'

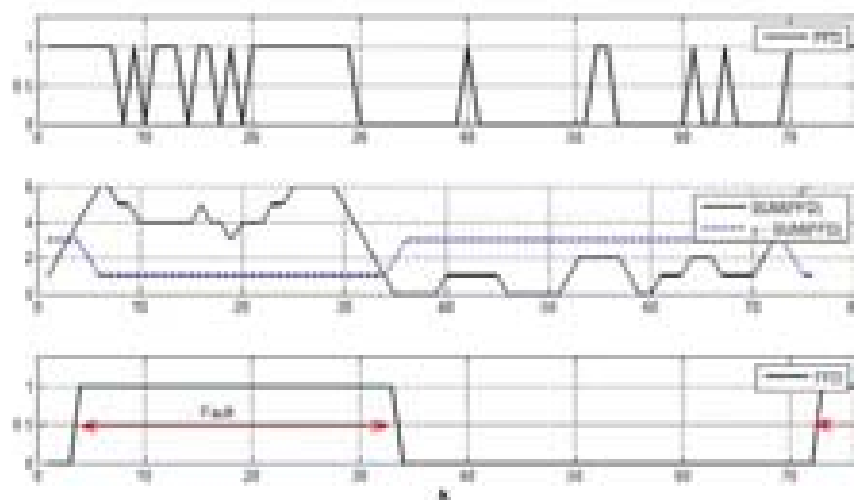
La estructura básica del contador se basa en dos ventanas de tiempo móviles. La señal de salida del clasificador se convierte a señal binaria (1 para estado de falla y 0 para estado normal) y se nombra como  $PF D_k$ . El  $FF D_k$  computa la ocurrencia de la señal binaria  $PF D_k$  dentro de un rango de tiempo determinado. La señal de decisión final  $FF D_k$  se produce cuando el número de  $PF D_k$  sobrepasa un valor constante  $\gamma$ .

$$FF D_k = \begin{cases} 1 & \text{Falla, si } \sum_{i=1}^{T_{w1}} PF D_{k-i} > \gamma - \sum_{j=1}^{T_{w2}} FF D_{k-j} \\ 0 & \text{Normal, en otro caso} \end{cases} \quad (5.5)$$

Donde  $k$  es el índice de tiempo,  $T_{w1}$  es el tamaño de la ventana de tiempo para la señal  $PF D_k$ ,  $\gamma$  es una constante y  $T_{w2}$  es el tamaño de la ventana para la señal  $FF D_k$ . El cómputo del FCTW se reduce a encontrar estos valores de forma tal que se eliminen el número de falsas alarmas y se incremente la precisión general del clasificador del Nivel 1.

Estos parámetros se encontraron experimentalmente teniendo en cuenta las recomendaciones realizadas en [102]:

- $T_{w1} > T_{w2}$
- Para el cálculo de  $T_{w2}$  fijar  $\gamma$  y asumir  $T_{w1} = 10\gamma$ , variar  $T_{w2}$  hasta maximizar precisión.



**Figura 5.12** Ejemplo del funcionamiento del FCTW.

Finalmente si para el tiempo  $k$   $FFD_k > 0$  se deduce que una falla se ha presentado. Como ejemplo del funcionamiento general del FCTW se plantea como señal PDF la mostrada en al Figura 5.12. Los parámetros fueron fijados como:

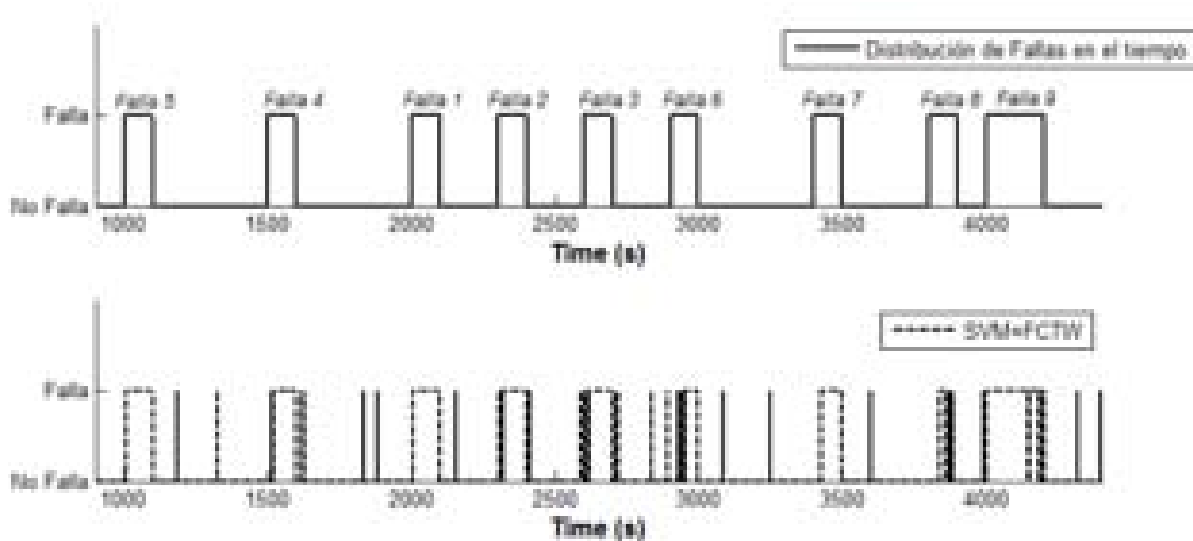
- $T_{w1} = 7$
- $T_{w2} = 2$
- $\gamma = 3$

En el sistema se presentan de forma real fallas en los intervalos de tiempo 0-30 y 69-77. Como se ve la señal de salida del clasificador (PDF) presenta intermitencias en el intervalo de tiempo 10-20 que se eliminan en la señal de salida del FCTW (FFD). Las Falsas Alarmas en el intervalo 30-69 se eliminan dado que no alcanzan a superar el umbral ( $\gamma$ ).

Una consecuencia directa del uso de ventanas de tiempo es el retraso en la detección, el inicio de la Falla es detectado 2 tiempos de muestreo después. La detección en el mejor de los casos se retrasa el valor de  $T_{w2}$  y puede llegar a ser tan alta como  $T_{w1}$ .

En la Tabla 5.7 se muestran los resultados promedios obtenidos a la salida del FCTW en combinación con los diferentes modelos de clasificación. Los set construidos para las pruebas en este nivel siguen la distribución de Fallas en el tiempo de la grafica superior de la Figura 5.13, que son las originalmente propuestas en [1]. La figura inferior corresponde a la predicción realizada por SVM+FCTW para una de las corridas realizadas.

En todos los casos la sensibilidad y la precisión aumentan significativamente. El porcentaje de falsas alarmas se mantiene significativamente bajo, teniendo en cuenta que la evaluación se realiza



**Figura 5.13** Predicción realizada por SVM+FCTW en el Nivel 1.

**Tabla 5.7** Evaluación de los clasificadores Nivel 1 a la salida del FCTW

Algoritmo	Precisión (%)	Sensibilidad a Fallas (%)	Falsas Alarmas
<b>SVM+FCTW</b>	<b>92.290</b>	<b>85.32</b>	<b>1,1 %</b>
<b>NN+FCTW</b>	<b>92.101</b>	<b>84.14</b>	<b>1,1 %</b>
KNN+FCTW	86.180	75.97	5.4 %
GMM+FCTW	87.270	80.74	5,0 %

registro a registro. En la Tabla 5.8 se realiza un análisis detallado del tiempo de recuperación dada la ocurrencia de una falsa alarma (False Alarm Clear Time). Con SVM y NN se logra un tiempo de recuperación promedio de 3.5 Ts.

**Tabla 5.8** Evaluación de los clasificadores Nivel 1 a la salida del FCTW

Algoritmo	Mín FA Clear Time	Max FA Clear Time	Promedio FA Clear Time
<b>SVM+FCTW</b>	<b>2Ts</b>	<b>7Ts</b>	<b>3,5Ts</b>
<b>NN+FCTW</b>	<b>2Ts</b>	<b>7Ts</b>	<b>3,5Ts</b>
KNN+FCTW	3Ts	12Ts	4,3Ts
GMM+FCTW	2Ts	14Ts	4,1Ts

## Construcción Nivel 2

Para el nivel dos todos los clasificadores son multiclase, esto es cada uno de los tipo de falla corresponden a una clase para un total de 9. Los resultados de la prueba de validación sobre 50 set de pruebas se muestran en la Tabla 5.9. La construcción de cada set se realizo exclusivamente con registros de fallas. La precisión más alta consiguió con el uso de redes neuronales y la mezcla de gaussianas.

**Tabla 5.9** Evaluación de los clasificadores Nivel 2

	<b>Precisión</b>
<b>SVM</b>	77,12
<b>NN</b>	<b>79,97</b>
<b>KNN</b>	58,47
<b>GMM</b>	<b>81,98</b>

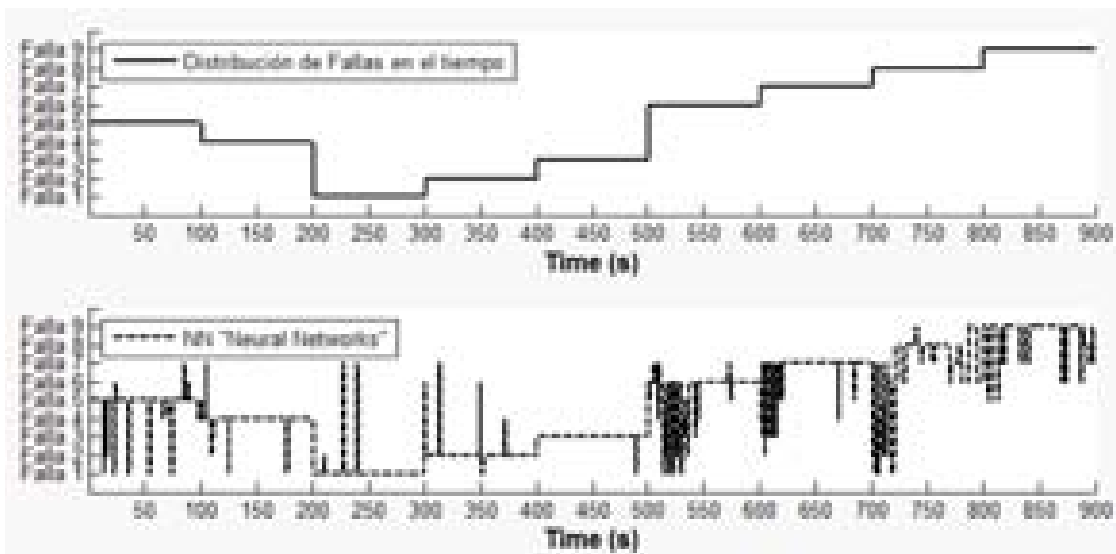
En la Tabla 5.10 se muestra la sensibilidad de cada algoritmo para cada una de las fallas (las celdas rojas son puntos de alta sensibilidad). Las SVM y NN detectan de forma correcta las Fallas 1-5 que corresponden a fallas en los sensores; el algoritmo GMM tiene sensibilidad sobresaliente para las Fallas 6-7 que corresponden a fallas en los actuadores, KNN por otra parte fue el algoritmo con peor comportamiento esto nos indica que la distancia entre los registros de diferentes fallas están muy próximos entre sí, siendo este un problema de clasificación altamente no lineal. En la Figura 5.14 se grafica la predicción realizada por las redes neuronales en uno de los set's de pruebas.

**Tabla 5.10** Sensibilidad a Fallas Nivel 2

	<b>Sensibilidad (%)</b>								
	<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>	<b>F5</b>	<b>F6</b>	<b>F7</b>	<b>F8</b>	<b>F9</b>
<b>SVM</b>	96,02	80,52	96,06	87,06	82,79	60,24	72,2	44,3	74,6
<b>NN</b>	94,08	85,16	98,9	86,02	79,06	65,86	78,72	53,2	78,4
<b>KNN</b>	84,23	65,21	91,2	79,24	45,75	38,27	39,4	17,8	69,2
<b>GMM</b>	95,32	75,89	93,21	86,32	88,65	71,2	88,5	57,2	81,1
F1, F2 , F3, F4, F5, F6 , F8 y F9 Abreviatura de Falla 1-9									

### 5.1.5. Pruebas experimentales

Dados los resultados de desempeño obtenidos, se construyo el sistema FDI usando para el Nivel 1 las maquinas de vectores de soporte y para el Nivel 2 la mezcla de gaussianas. Se generaron 50



**Figura 5.14** Clasificación realizada por las redes Neuronales en el Nivel 2.

set de datos de 440000 registros cada uno a una tasa de muestreo de 0,01s (representan un tiempo de simulación de 4400 segundos) con diferentes secuencias de ruido para realizar las pruebas experimentales, el proceso se realizó de forma secuencial emulando un comportamiento real. La evaluación se realizó en términos de las métricas de establecidas del benchmark:

- Tiempo de detección (promedio -TDP-, máximo -TDMA- y mínimo -TDMI-): Tiempo que le toma al sistema FDI detectar la ocurrencia de una falla.
- Fallas no detectadas (FND): Número de fallas que no fueron detectadas. Tal como se menciona anteriormente se establece como falla no detectada aquella para la cual no se reconocieron al menos el 30% de los registros de ocurrencia de la misma.
- Falsas alarmas (FA): Número promedio de falsas alarmas. Para el sistema FDI construido se determina como falsa alarma aquella que siendo un comportamiento normal es detectada como falla en el Nivel 1, se le 'carga' esta falsa alarma al tipo de falla detectada en el Nivel 2.
- Tiempo de recuperación de una falsa alarma (TR-FA): Tiempo promedio que tarda el sistema FDI en recuperarse posterior a la ocurrencia de una falsa alarma.
- Falsos aislamientos (FAISL): Número promedio de clasificaciones erróneas realizadas en el Nivel 2 para una determinada falla. A diferencia de una falsa alarma un falso aislamiento ocurre cuando una falla correctamente detectada en el Nivel 1 es mal clasificada en el Nivel 2.



- Tiempo de recuperación de un falso aislamiento (TR-FAISL): Tiempo promedio que tarda el sistema FDI en recuperarse posterior a la ocurrencia de un falso aislamiento.

**Tabla 5.11** Resultados de la Evaluación al sistema FDI

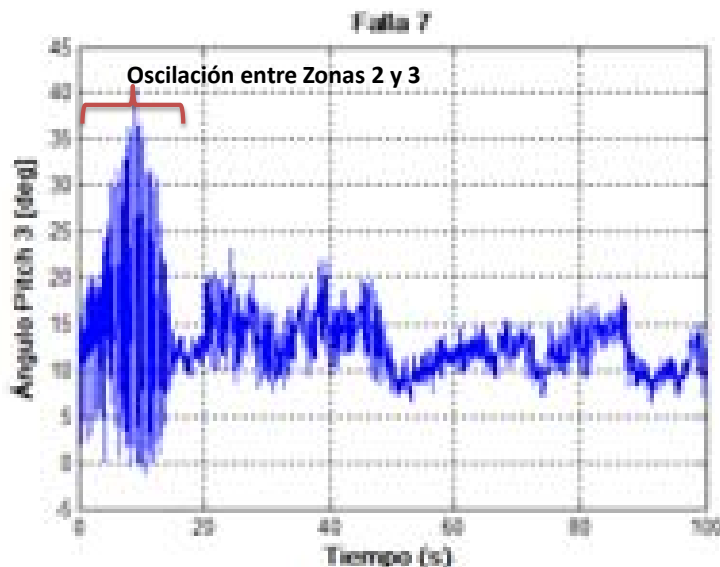
Falla No.	TDP	TDMA	TDMI	FA	TR-FA	FND	FAISL	TR-FAISL
1	4.2Ts	7 Ts	3 Ts	2	3,5 Ts	0	4	1 Ts
2	4.3 Ts	8 Ts	3 Ts	4	3,4 Ts	0	10	1 Ts
3	4.2 Ts	7 Ts	3 Ts	1	3 Ts	0	6	1 Ts
4	17.2 Ts	24 Ts	8 Ts	3	4.5 Ts	0	11	1 Ts
5	5.2 Ts	9 Ts	3 Ts	3	4.5 Ts	0	11	1 Ts
6	5125 Ts	5231 Ts	5055 Ts	3	3.4 Ts	0	18	1 Ts
7	2325.4 Ts	2725 Ts	1627 Ts	5	4.2 Ts	0	12	1 Ts
8	3450 Ts	5450 Ts	2450 Ts	12	14.2 Ts	0	35	1 Ts
9	1230 Ts	1420 Ts	980 Ts	8	7.1 Ts	0	7	1 Ts

En la Tabla 5.11 se resumen los resultados de la evaluación. Los requerimientos establecidos en el capítulo 4 se cumplen exclusivamente para las Fallas 1-4 (TDP < 10 Ts). La Falla 5 se retrasa en la detección hasta 14 Ts cuando el ruido en magnitud supera los 0.3 rad/s en los 30 primeros tiempos de muestreo iniciada la falla en el sensor de velocidad del rotor, pasado este tiempo independiente del ruido la Falla es detectada. En general las fallas en los sensores presentaron un bajo número de falsas alarmas y falsos aislamientos.

La falla hidráulica en el segundo pitch (Falla 6) es la que tarda más en identificarse, esto se debe a que en el benchmark esta falla se simula en las zonas 2 y 3 de operación de la turbina y es solo en la zona 3 cuando la acción de control da movimiento a los pitch (en la zona 2 la estrategia de control los posiciona en valores cercanos a cero grados). Durante los 50 segundos iniciales está en zona de operación 2, siendo imposible evidenciar la falla a partir de los datos. Un fenómeno similar ocurre con la Falla 7 para la cual el sistema de control se encuentra oscilando en zona de operación 2 y 3 en los primeros 18 segundos de su inicio. Después de este tiempo el movimiento de los pitch se hace evidente en los datos la degradación de este actuador. En la Figura 5.15 se muestra de forma grafica el comportamiento de la Falla 7.

La Falla 6 presento el segundo mayor índice de falsos aislamientos al ser mal clasificada 18 veces. Cabe resaltar que las 18 veces el sistema la confundió con la Falla 7.

Para el sistema FDI diseñado la Falla 8 es todo un desafío dado que el valor del bias que afecta la medida del torque se incrementa desde 0 Nm a 100 Nm de forma gradual en el tiempo. Debido a los altos niveles de ruido en esta medida, solo cuando el offset es igual o superior a 80Nm es detectado por el sistema FDI afectando la velocidad del generador. Esta falla presento los niveles más altos de falsas alarmas y falsos aislamientos.



**Figura 5.15** Comportamiento de la Falla 7.

La Falla 9 es de carácter incipiente y su efecto degrada de forma general la turbina. Esta falla como se verá en el siguiente apartado es un desafío para los sistemas FDI que hacen uso de residuos.

### Comparación de resultados con métodos basados en modelo

Dado que este modelo se propuso para una competición IFAC se cuenta con un número importante de referencias de sistemas FDI construidos basado en modelos. De ellos se seleccionó uno que contiene la mayoría de principios de generación de residuos explicados en el Capítulo 2 y fue publicado por un grupo importante de investigadores en el área de la Universidad de Minnesota [103]. El objetivo es realizar una comparación en términos de las métricas establecidas en el benchmark y determinar las ventajas de de cada una de las técnicas.

La publicación realizada en [103] usa lo que se conoce industrialmente en el área como 'técnicas de umbralización con contadores'. La idea básica es contar residuos (de forma muy similar al FCTW) hasta que superen un determinado umbral, por lo tanto se debe construir para cada falla un sistema de conteo y se determina experimentalmente un umbral. En esta publicación generan residuos mediante 3 técnicas:

- **Ecuaciones de Paridad:** La generación de residuos se realiza comparando la respuesta 'ideal' de un modelo matemático construido, con las medidas reales. Cabe mencionar que se usaron las señales de control dentro del modelamiento matemático.
- **Redundancia Física:** Se comparan las señales redundantes propias del modelo.

- **Filtros de Kalman:** Se construyeron filtros para hacer estimaciones de las velocidades del rotor y generador, los residuos se generan al compararlas con las medidas reales.

A la salida de todos los residuos implementaron filtros digitales para reducir el ruido. Algunos de ellos empleaban hasta 40 muestra de tiempo anteriores para su cálculo lo cual implica un almacenamiento y procesamiento mayor.

La evaluación de resultados realizados se resume en la Tabla 5.12. La tasa de falsas alarmas (FA Rate) se define como el número de falsas alarmas por tiempo de muestreo  $T_s=0.01s$ , a partir de esta y para efectos de comparación se calcula el número de Falsas Alarmas por cada 4400s (que corresponde al tiempo de simulación trabajado en esta tesis).

**Tabla 5.12** Resultados de la Evaluación al sistema FDI

No.	TDP	FA Rate	FA-Simulación	TR-FA	FND
1	3 Ts	$0,091 \times 10^{-5}$	0,4004	2,9 Ts	0
2	819,4 Ts	$0,046 \times 10^{-5}$	0,2024	2,3 Ts	0
3	3 Ts	$0,046 \times 10^{-5}$	0,2024	2,4 Ts	0
4	12,4 Ts	$0,068 \times 10^{-5}$	0,2992	18,8 Ts	0
5	187,4 Ts	$0,159 \times 10^{-5}$	0,6996	17,5 Ts	0
6	5050 Ts	$0,182 \times 10^{-5}$	0,8008	3,9 Ts	0
7	1573 Ts	$0,022 \times 10^{-5}$	0,0968	3,2 Ts	0
8	1 Ts	$0 \times 10^{-5}$	0	0 Ts	0
9	N/A	N/A	N/A	N/A	100

Un conjunto de conclusiones se derivan de la comparación de estos resultados:

- Los tiempos de detección no favorecieron al sistema FDI data-driven construido, solo para las Fallas 2 y 5 se supero al sistema basado en modelo. Un punto clave en los tiempos de detección es que en el sistema a dos niveles basado en clasificadores no es posible una detección en menos de 3 Ts debido al contador de fallas FTCW que se implemento.
- El sistema basado en modelos no es capaz de detectar la Falla 9, tal y como lo explican en [103] el cambio generado es demasiado pequeño como para ser detectado mediante técnicas basadas en modelos o generación de residuos. Por el contrario el sistema data-driven construido si fue capaz de detectar esta pequeña degradación en la eficiencia general de la turbina con un tiempo promedio de detección de 12,30s con un máximo de 8 falsas alarmas en una simulación.
- Para la Falla 6 y 7 las técnicas basadas en modelos presentan los mismos inconvenientes de detección tardía en región de operación 2 explicados anteriormente.

- La Falla 8 es totalmente detectada mediante residuos basados en ecuaciones de paridad y filtros, todo lo contrario pasa con el sistema data-driven construido donde la variación de 100 Nm en el torque es difícilmente perceptible para los niveles de ruido de esta medida. La diferencia en los resultados se justifica en el uso de señales del sistema de control para determinar el aumento de eficiencia en el torque del generador.
- Las Falsas Alarmas son significativamente bajas para el sistema basado en modelos y tal como lo especifican en [103] pueden reducirse aun mas cambiando los limites de umbralización a costa quizás de mayores tiempos de detección. Es natural que un sistema data-driven construido mediante clasificadores posea un mayor número de falsas alarmas dado que la detección se hace registro a registro, sin embargo se destaca la implementación del contador FCTW para su disminución.

En términos generales si se tiene en cuenta que no se uso redundancia física, señales de control y/o filtros digitales, se obtuvieron muy buenos resultados con el sistema data-driven construido. Es importante señalar además, que el conocimiento específico sobre la turbina necesario para construir un clasificador es considerablemente menor que el necesario para construir un modelo que permita una generación adecuada de residuos, se concluye por lo tanto que el tiempo y esfuerzo de construcción de un sistemas FDI basado en modelo es mayor.

## 5.2. Modelo 2: FAST

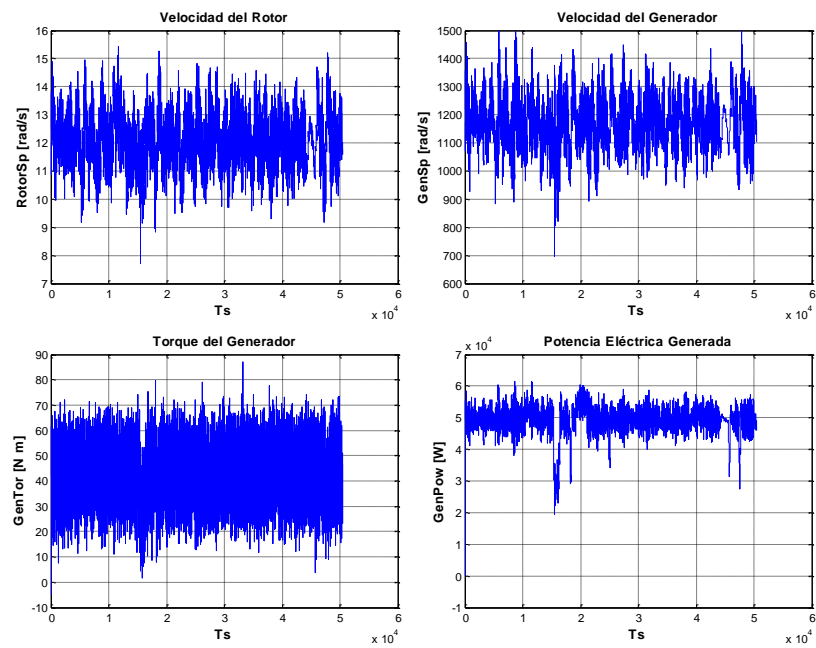
### 5.2.1. Análisis exploratorio de los datos

Como se mencionó en el capítulo 4 este modelo no cuenta con redundancia física por lo que no fue necesaria la reducción de características, en la Tabla 5.13 se lista el conjunto de atributos manejados.

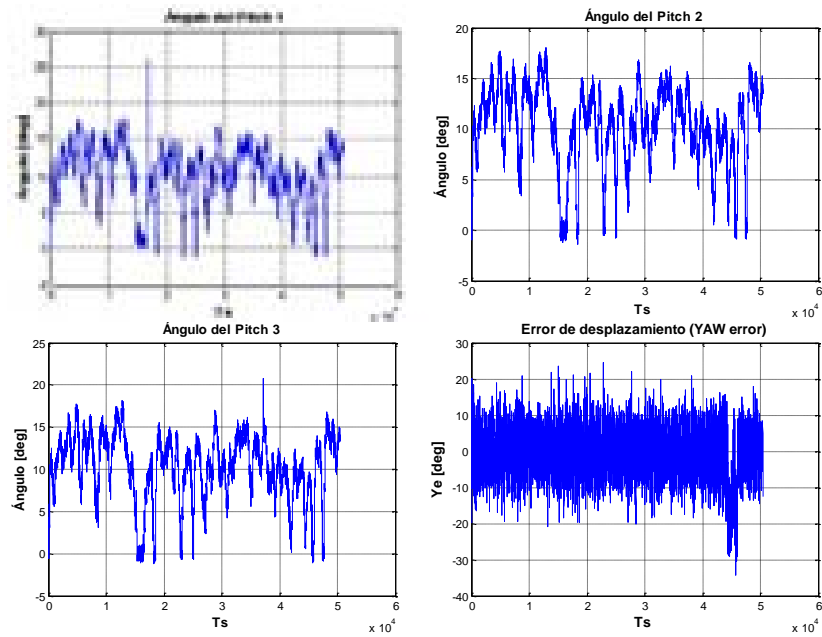
En las Figuras 5.16 y 5.17 se grafican cada una de estas características para una simulación de 500 s a una tasa de muestreo  $T_s = 0,01s$ . Para esta simulación se trabajo con el modelo de viento especificado en la Figura 5.18, se observa casi de forma permanente valores de vientos altos (zona de operación 3 de la turbina).

#### Eliminación de outliers

A los registros de comportamiento normal que se usaron como parte del conjunto de datos de entrenamiento se les realizo un análisis de histogramas y diagramas de caja para la identificación de outliers.



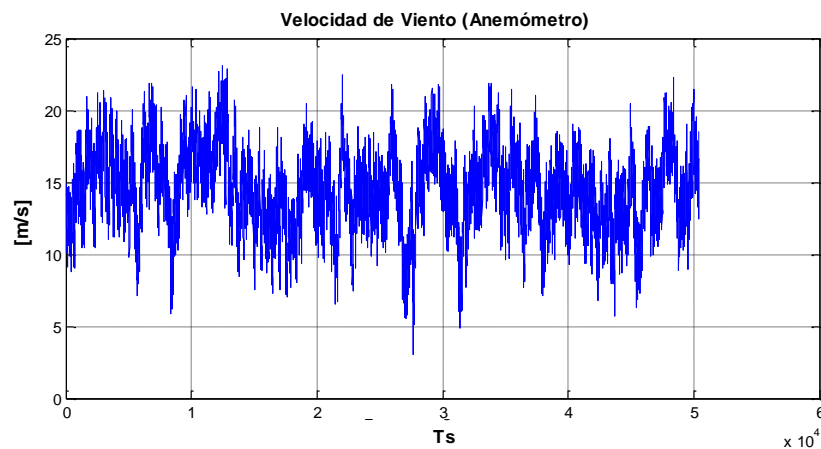
**Figura 5.16** Grafica de las características: Velocidad del Rotor, Velocidad-Torque del Generador y Potencia Generada (Modelo Fast).



**Figura 5.17** Grafica de las características: Angulos del Pitch y Error de desplazamiento.

**Tabla 5.13** Conjunto de atributos del set de entrenamiento, Modelo FAST

<b>Atributo/Característica</b>	<b>Notación</b>
<i>Medida de la velocidad del Viento</i>	WindSp [m/s]
<i>Medida de velocidad del Rotor</i>	RotorSp [rad/s]
<i>Medida de velocidad del Generador</i>	GenSp [rad/s]
<i>Medida del torque del Generador</i>	GenTorq [N m]
<i>Medida de potencia eléctrica generada</i>	GenPower [W]
<i>Medida del Ángulo del Pitch 1</i>	B11 Pitch [deg]
<i>Medida del Ángulo del Pitch 2</i>	B12 Pitch [deg]
<i>Medida del Ángulo del Pitch 3</i>	B12 Pitch [deg]
<i>Error de desplazamiento (YAW Error)</i>	Ye [deg]

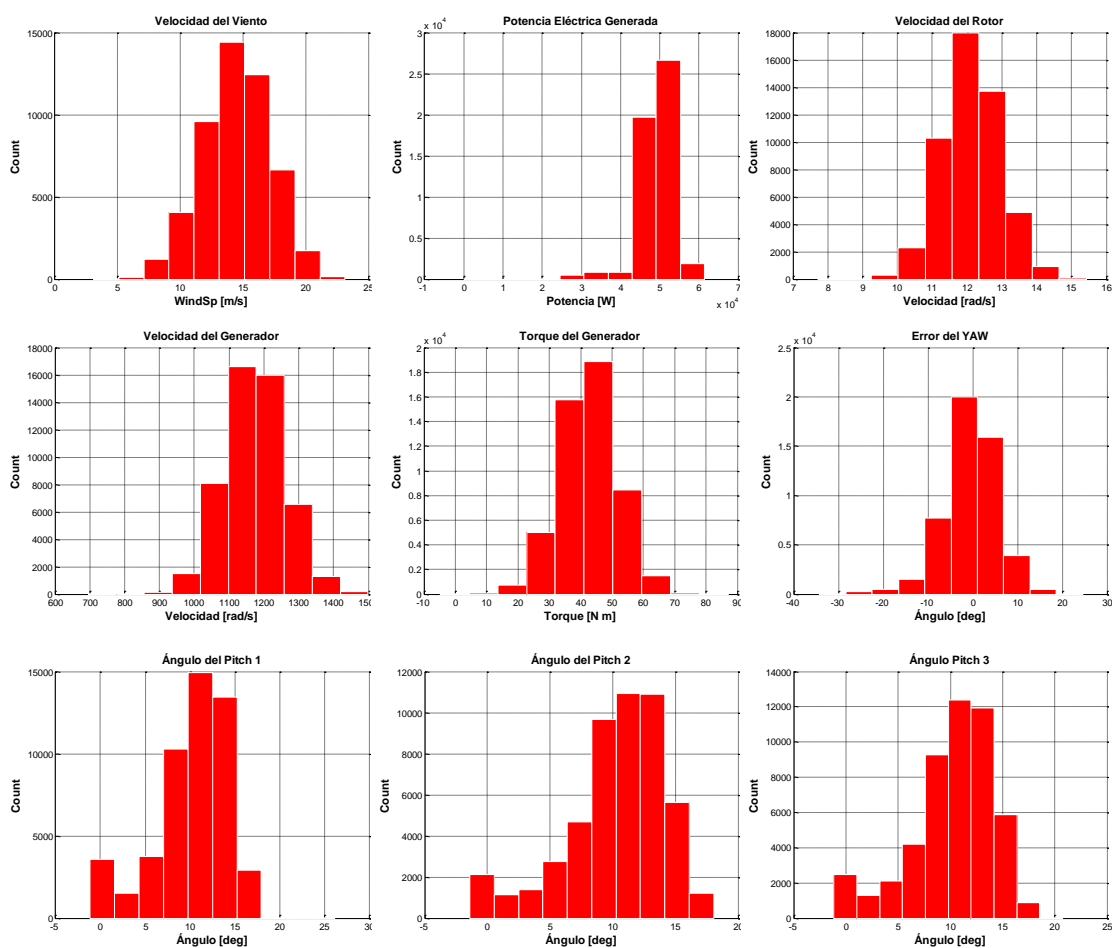
**Figura 5.18** Modelo de Viento usado en FAST.

La metodología de construcción de cada grafica fue la misma utilizada en el modelo anterior en las Figura 5.19 se muestran los histogramas a 10 bins de cada característica y en las Figura 5.20 los diagramas de caja.

En los histogramas se puede ver claramente la correcta operación del sistema de control, la mayor parte de la simulación la potencia generada es máxima (alrededor de 5MW) y el error de desplazamiento (YAW error) es mínimo dado un modelo de viento cuyos valores más frecuentes oscilan entre 13m/s y 17 m/s.

A través de los diagramas de caja se eliminaron un conjunto de registros con características atípicas:

- Registros del Pitch 1 por encima a 22 grados, dado que ningún otro pitch las presenta. Tén-



**Figura 5.19** Histogramas a 10 bins de cada una de las características en el set de datos (Modelo Fast).

gase en cuenta que la señal de control es la misma para los tres pitch y su respuesta debería ser similar en magnitud. Estos valores se presentan por valores atípicamente altos de ruido en el modelo del sensor.

- Valores de torque por encima de 70 N m son producidas por vientos superiores a 25 m/s [96] estas velocidades no se presentan en el modelo de viento trabajado.

Como último paso en el preprocesamiento se añadió un conjunto de características referentes a la tasa de cambio de las señales involucradas. Se define la tasa de cambio del atributo  $x$  como:  $\Delta(x(k)) = |x(k) - x(k-1)|$  y cumple con el mismo propósito explicado en el modelo anterior.

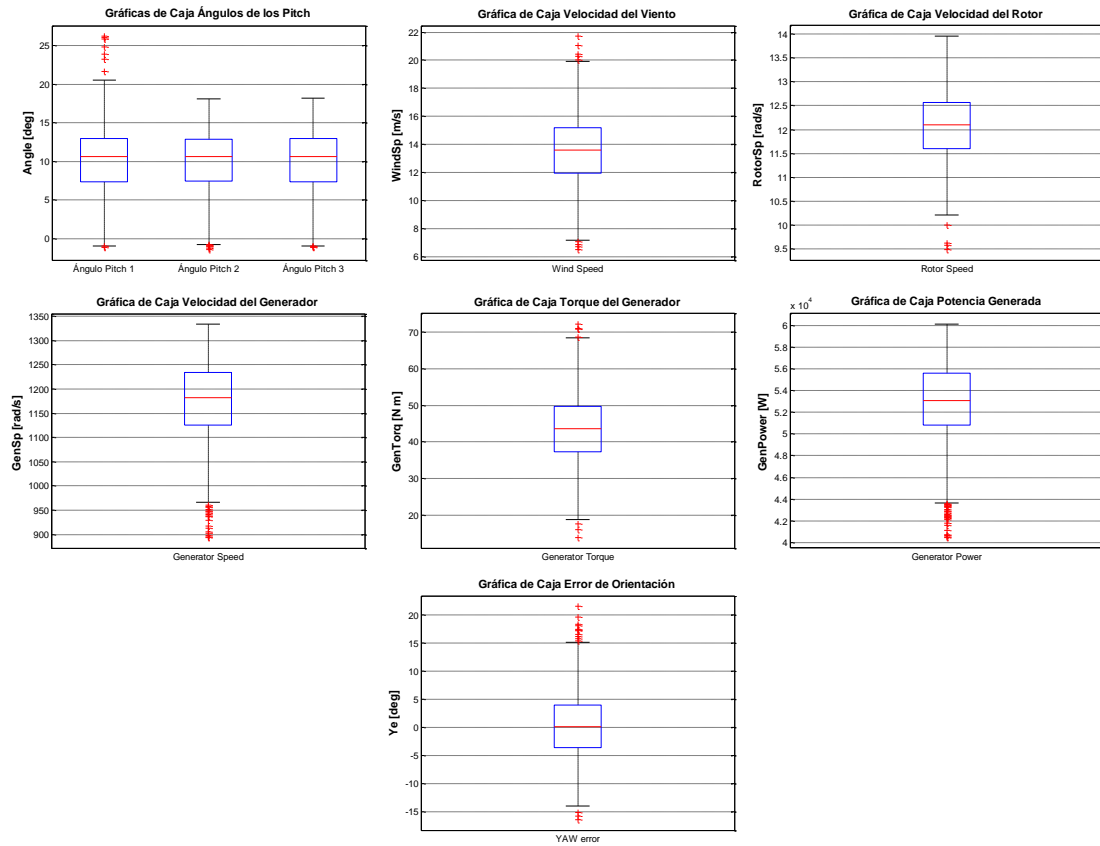


Figura 5.20 Gráficas de Caja, identificación de outliers en modelo FAST.



### 5.2.2. Construcción del sistema FDI (Arquitectura a dos niveles)

Se propone un sistema FDI a dos niveles, en cada uno de los cuales se tendrá un modelo de clasificación. Siguiendo la misma línea de trabajo se inicia con la construcción del Nivel 1 aplicando las 4 técnicas de clasificación estudiadas.

#### Nivel 1

Se construyeron 50 set de pruebas a partir de simulaciones sobre el modelo de turbina. El desempeño promedio obtenido por cada algoritmo se resume en la Tabla 5.14.

**Tabla 5.14** Evaluación de los clasificadores Nivel 1

Algoritmo	Precisión (%)	Sensibilidad a Fallas (%)	Falsas Alarmas
<b>SVM+FCTW</b>	<b>95,68</b>	<b>85,32</b>	<b>1,70 %</b>
<b>NN+FCTW</b>	<b>95,65</b>	<b>84,17</b>	<b>1,44 %</b>
KNN+FCTW	83,15	77,29	15,79 %
GMM+FCTW	90,13	80,74	7,67 %

Para el Nivel 1 los resultados favorecen a las Maquinas de Vectores de Soporte en precisión global y sensibilidad a Fallas, sin embargo las redes neuronales tienen un número de falsas alarmas mucho menor. En general este comportamiento nos indica que el clasificador SVM detecta mejor las fallas a costa de un número mayor número de falsas alarmas. Se exploraron diferentes configuraciones de redes neuronales en busca de un menor porcentaje de falsas alarmas, se logro un mínimo de 1,44 % con redes de base radial y 5 capas ocultas.

#### Nivel 2

Para el Nivel 2 los resultados obtenidos se muestran en las Tablas 5.15 (sensibilidad) y 5.16 (Precisión).

**Tabla 5.15** Evaluación de los clasificadores Nivel 2 - Sensibilidad a Fallas

	Sensitivity (%)								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
<b>SVM</b>	94,62	80,22	94,26	84,86	81,39	65,44	83,6	72,92	92,76
<b>NN</b>	92,68	82,86	97,1	83,82	77,66	64,06	80,12	71,84	96,61
<b>KNN</b>	82,83	62,91	89,4	77,04	44,35	36,47	40,8	76,4	90,35
<b>GMM</b>	93,92	83,59	92,41	84,12	87,25	79,4	89,9	77,82	91,33

**Tabla 5.16** Evaluación de los clasificadores Nivel 2 - Precisión Global

Algoritmo	Precisión (%)
<b>SVM</b>	<b>83,34</b>
NN	82,97
KNN	66,73
<b>GMM</b>	<b>86,64</b>

En el Nivel 2, con las GMM se obtiene la precisión global más alta y la mayor sensibilidad para Fallas de actuadores, este resultado es interesante y ratifica las cualidades expresadas por esta técnica en el modelo anterior. Se plantea como hipótesis que las técnicas probabilísticas como las GMM presentan un mejor desempeño en Fallas de carácter incipiente.

Los resultados referentes a la sensibilidad muestran un comportamiento muy similar al modelo benchmark, solo para la Falla 9 no se puede hacer una comparación directa dado que difiere en ambos modelos. La Falla 8 muestra un incremento en su detección como resultado del aumento a 1000N en el offset del torque del generador, aun así no se supero el 80 % de sensibilidad para esta Falla.

### 5.2.3. Pruebas experimentales

La construcción del sistema FDI se realizo en el Nivel 1 con redes neuronales y el Nivel 2 con GMM. Se generaron 50 set de datos de 500000 registros cada uno a una tasa de muestreo de 0,01s (representan un tiempo de simulación de 5000 segundos) con diferentes secuencias de ruido para realizar las pruebas experimentales, el proceso se realizo de forma secuencial emulando un comportamiento real. En cada set de datos se simularon las 9 Fallas en diferentes instantes de tiempo. La evaluación se realizo en términos de las métricas anteriormente establecidas. Los resultados se muestran en la Tabla 5.17.

Dado el modelo de viento no se tienen transiciones entre zonas de operación de la turbina permitiendo detecciones mucho más rápidas para las Fallas 6, 7 y 8. Todas las Fallas de sensores se detectan en menos de 6 Ts con un número de falsas alarmas de cómo máximo tres (3) por simulación y un tiempo de recuperación de 3 Ts. En el modelo FAST a diferencia del benchmark los falsos aislamientos que se presentaron son menores, es solo para la Falla 8 donde existe una cantidad considerable. Esta Falla se considera un desafío para los métodos basados en datos que no usan las señales de control para entrenamiento.

**Tabla 5.17** Resultados de la Evaluación al sistema FDI modelo FAST

No.	TDP	TDMA	TDMI	FA	TR-FA	FND	FAISL	TR-FAISL
1	4.6 Ts	8 Ts	3 Ts	2	3 Ts	0	4	1 Ts
2	4.8 Ts	10 Ts	3 Ts	1	3 Ts	0	3	1 Ts
3	5.2 Ts	8 Ts	3 Ts	1	3 Ts	0	4	1 Ts
4	3.4 Ts	4 Ts	3 Ts	0	0	0	3	1 Ts
5	5.2 Ts	9 Ts	3 Ts	1	3 Ts	0	3	1 Ts
6	125.6 Ts	321 Ts	50 Ts	6	4,2 Ts	0	7	1 Ts
7	257.4 Ts	425 Ts	170 Ts	5	5 Ts	0	8	1 Ts
8	150.8 Ts	550 Ts	60 Ts	21	5.2 Ts	0	13	1 Ts
9	5.2 Ts	7 Ts	3 Ts	3	3 Ts	0	7	1 Ts



# Capítulo 6

## Construcción de un sistema FDI híbrido

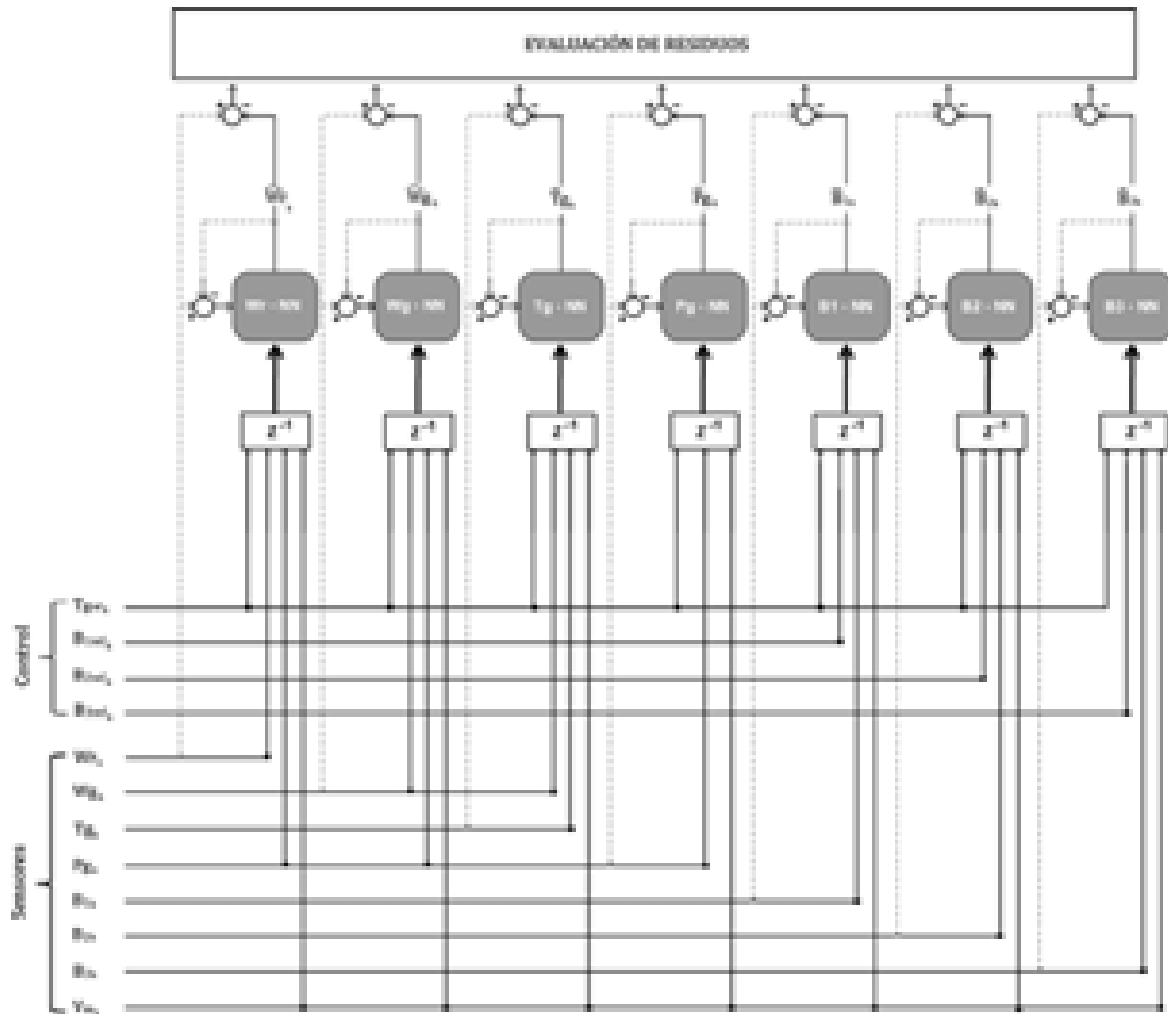
Los resultados obtenidos con la arquitectura a dos niveles expuesta en el capítulo anterior permitieron identificar dos debilidades de los sistemas FDI basados en clasificadores:

- Presentan un número alto de falsas alarmas (aun con la implementación del FCTW).
- Se presentan falsos aislamientos que no son más que errores en la clasificación de una falla en el nivel 2. Tal y como se especifica en [18] no aislar correctamente una falla puede llegar a ser tan grave como su no detección.

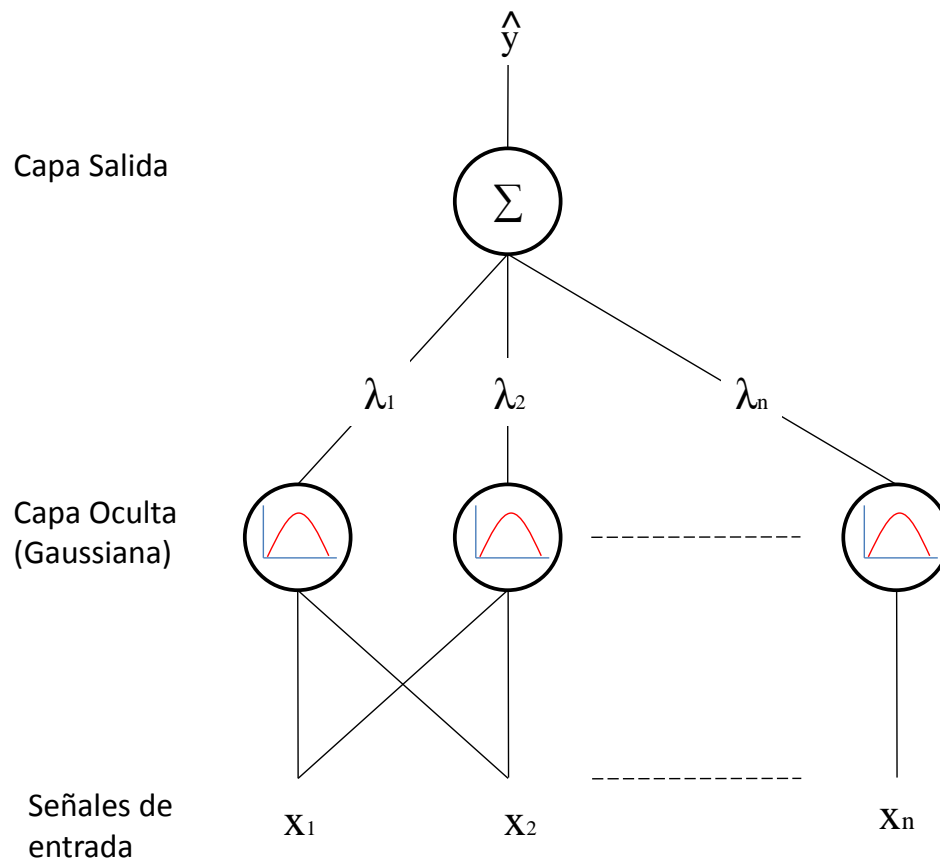
Con el objeto de superar estas limitaciones se propone una nueva arquitectura que aprovecha las fortalezas conocidas de los clasificadores y las combina con modelo de regresión para la generación de residuos. Tal y como se evidencio en los resultados de [103] las técnicas basadas en modelos para la generación de residuos presentan menores tasas de falsas alarmas y una menor sensibilidad al ruido en las señales, sin embargo no son muy adecuadas para fallas incipientes que degradan lentamente la eficiencia global del sistema (i.e Falla 9).

El esquema híbrido propuesto puede ser dividido en tres etapas (Figura 6.1). La etapa 1 involucra el modelamiento de cada una de las señales del sistema de monitoreo (sensores) con una red neuronal, cuya función va ser la de predecir el valor de dicha señal en el instante de tiempo  $k$  a partir de un conjunto de señales de entrada (redundancia analítica). La etapa 2 consiste en la generación de residuos a partir de la comparación de la salida real y la estimación de la red neuronal y finalmente la etapa 3 evalúa el conjunto de residuos generados con un sistema de clasificación. En la etapa 3 se hace tanto la detección como es aislamiento de la falla.

La diferencia principal con los sistemas FDI diseñados en el capítulo anterior son los atributos de entrada al sistema de clasificación, en vez de trabajar directamente con las señales del sistema de monitoreo se trabaja con un conjunto de residuos generados por la diferencia entre las salidas reales y estimaciones/predicciones de los modelos de regresión construidos con redes neuronales.



**Figura 6.1** Arquitectura híbrida propuesta.



**Figura 6.2** Red de base radial (NN-RBF) con conexión completa.  $x, \lambda, \hat{y}$  son las entradas, los pesos de las conexiones y la salida respectivamente.

## 6.1. Construcción de los modelos de regresión con redes neuronales

Tal y como se muestra en la Figura 6.1 un total de 7 modelos de regresión se construyeron, la estructura empleada en todas las redes neuronales se conoce en la literatura como EMRAN RBF, su selección se realizó debido a su excelente capacidad de generalización y su habilidad de reducir neuronas ocultas que no contribuyen significativamente a la salida de la red neuronal. Esta propiedad permite reducir el tamaño total de la red y su tiempo de ejecución lo cual es beneficioso para aplicaciones en línea donde existen limitaciones de potencia computacional.

La EMRAN RBF consiste en un grupo de señales de entrada, una capa oculta y una capa de salida (Figura 6.2). Empieza inicialmente con cero neuronas ocultas y se van añadiendo si y solo si los siguientes tres criterios se cumplen:

$$e_k = |y_k - \hat{y}_k| > E1 \quad (6.1)$$

$$e_{RMS_k} = \sqrt{\sum_{j=k-(n-1)}^k \frac{e_j^2}{n}} > E2 \quad (6.2)$$

$$d_k = \|\mathbf{x}_k - \boldsymbol{\mu}_{rk}\| > E3 \quad (6.3)$$

Donde  $y_k$  y  $\hat{y}_k$  son la salida real del sensor y la estimación de la red neuronal en el tiempo de muestra  $k$ , y  $\boldsymbol{\mu}_{rk}$  es el centro (media) de la neurona oculta más cercana a la al vector de entrada actual  $x_k$ .  $E1$  asegura que el error de estimación sea menor que un umbral,  $E2$  chequea si la RMS (root mean square) de los pasados  $n$  errores de estimación son suficientemente pequeñas y  $E3$  verifica si la distancia mínima entre el vector actual de entrada y los centros de las neuronas ocultas es significativamente pequeña.  $E3$  es generalmente expresada como  $E3 = \max \left\{ \varepsilon_{max} \gamma_{df}^i, \varepsilon_{min} \right\}$ , donde  $\varepsilon_{max}$  es el valor máximo inicial que decae exponencialmente según  $\gamma_{df}^i$  hasta un valor mínimo  $\varepsilon_{min}$  [104].

Si al menos uno de estos tres criterios se cumple entonces el algoritmo de entrenamiento actualiza los parámetros (peso, centro y ancho) de la neurona 'mas' activa (o también llamada la 'ganadora'). Esto reduce el número de parámetros a ser actualizados lo que agiliza el proceso de entrenamiento. Una descripción detallada de su implementación y evaluación puede ser encontrada en [104].

### 6.1.1. Selección de entradas/salidas de las redes neuronales

En principio se construyeron redes que tenían como entradas todas las posibles señales del sistema de monitoreo, como resultado se obtuvieron procesos de entrenamiento lentos, redes de grandes tamaños y modelos resultantes con poca capacidad de generalización. Fue necesario realizar pequeñas pruebas para eliminar aquellas señales de entrada que tienen una contribución menor en la estimación. Esto se logro calculando la diferencia entre error de estimación RMS antes y después que un atributo es incluido en el set de entrenamiento. Si la diferencia es pequeña y se mantiene en diferentes modelos entrenados, entonces el atributo puede ser removido de forma segura sin que esto afecte de forma significativa la estimación. El resultado de las 7 redes se describe por las siguientes relaciones de entrada/salida, donde  $k$  es el tiempo de muestreo:

$$Wr_k = NN(Tg_{ref_{k-1}}, Wr_{k-1}, Pg_{k-1}, Vm_{k-1}) \quad (6.4)$$



$$Wg_k = NN(Tg_{ref_{k-1}}, Wg_{k-1}, Pg_{k-1}, Vm_{k-1}) \quad (6.5)$$

$$Tg_k = NN(Tg_{ref_{k-1}}, Wg_{k-1}, Tg_{k-1}, Vm_{k-1}) \quad (6.6)$$

$$Pg_k = NN(Tg_{ref_{k-1}}, Pg_{k-1}, Vm_{k-1}) \quad (6.7)$$

$$B1_k = NN(Tg_{ref_{k-1}}, B1_{ref_{k-1}}, B1_{k-1}, Vm_{k-1}) \quad (6.8)$$

$$B2_k = NN(Tg_{ref_{k-1}}, B2_{ref_{k-1}}, B2_{k-1}, Vm_{k-1}) \quad (6.9)$$

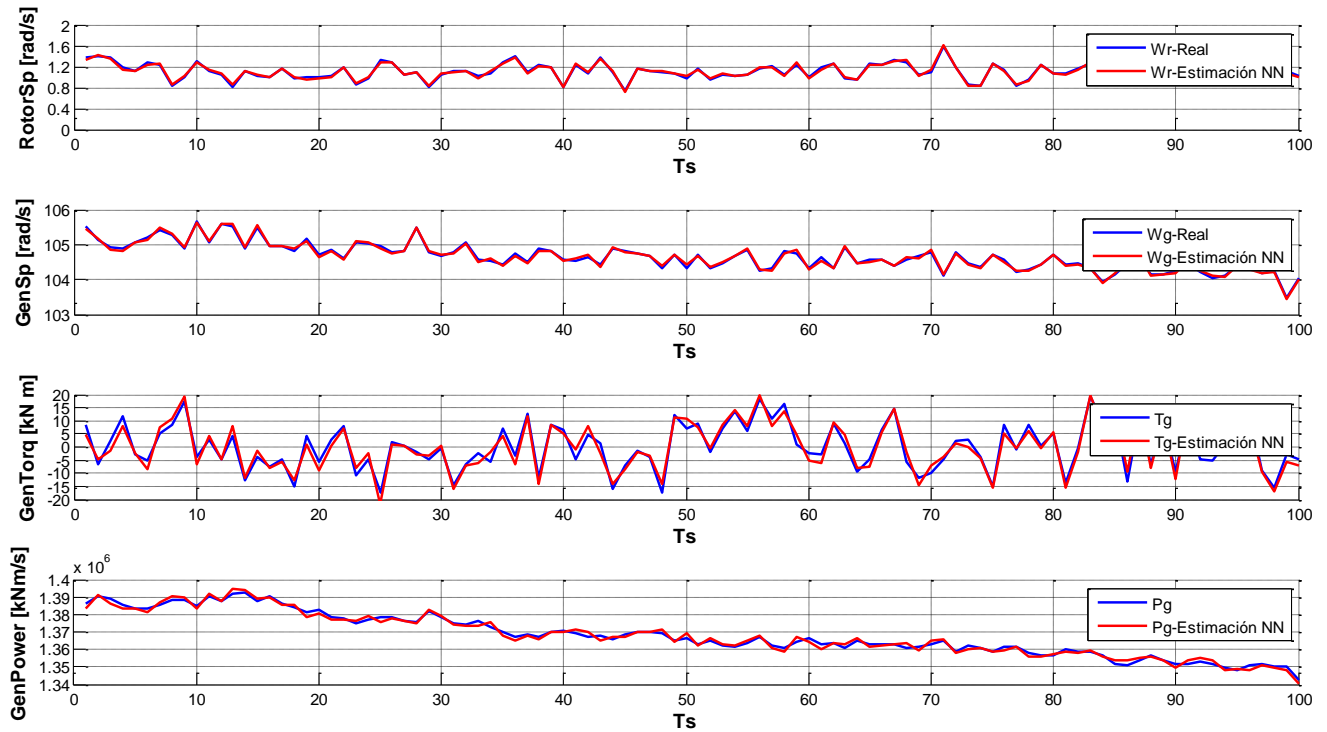
$$B3_k = NN(Tg_{ref_{k-1}}, B3_{ref_{k-1}}, B3_{k-1}, Vm_{k-1}) \quad (6.10)$$

Como se observa en la Figura 6.2 y ecuaciones 6.4-6.10 se añadieron las señales de referencia del sistema de control como entrada a los modelos de regresión, esto con el objetivo de mejorar los resultados de detección de las Fallas 8 y 9. Si bien se estableció en el capítulo anterior que una característica deseable de un sistema FDI es la independencia del controlador, con el objeto de lograr una detección a niveles industriales se incluyó las señales de referencia del controlador.

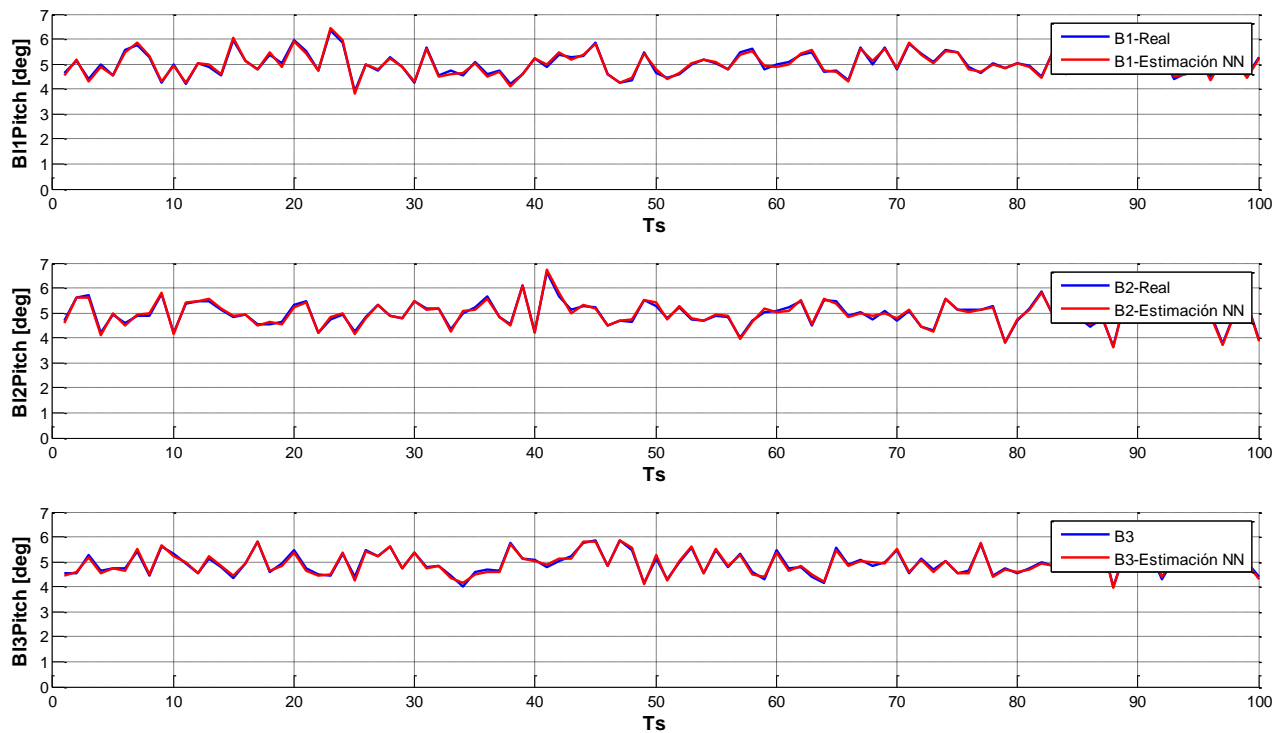
### 6.1.2. Entrenamiento de las redes neuronales.

Para el entrenamiento de las redes neuronales se construyó un set de entrenamiento de 50.000 registros de comportamiento normal obtenidos a partir de simulaciones sobre el modelo benchmark de turbina de viento explicado en capítulos anteriores. Un problema frecuente que se presentó en estos modelos de regresión fue el sobre entrenamiento (memorización de los datos), para evitarlo en el proceso de entrenamiento se verifica por cada época un set de pruebas de 5.000 registros. El entrenamiento continúa hasta el punto donde el error de estimación RMS en el set de pruebas se incrementa por más de 100 evoluciones (épocas) de la red. Este comportamiento de sobreentrenamiento tan marcado no se presentó en los modelos de clasificación construidos con redes neuronales.

Un resumen de las estructuras de las redes construidas se muestra en la Tabla 6.1. El error de estimación que se presenta, fue el obtenido al evaluar el desempeño individual de cada red aplicado a un set de pruebas de 5.000 registros. En las figuras 6.3 y 6.4 se muestra las estimaciones realizadas por cada red entrenada vs la salida real.



**Figura 6.3** Comparación entre la salida real y la estimación realizada por las redes neuronales (I).



**Figura 6.4** Comparación entre la salida real y la estimación realizada por las redes neuronales (II).

**Tabla 6.1** Resumen de las características de las redes neuronales entrenadas

	$Wr - NN$	$Wg - NN$	$Tg - NN$	$Pg - NN$	$B_1 - NN$	$B_2 - NN$	$B_3 - NN$
Error de estimación	0.0973 <i>rad/s</i>	0.1054 <i>rad/s</i>	1.74 Nm	0.891 <i>KNm/s</i>	0.070 deg	0.091 deg	0.081 deg
Atributos de entrada	$T_{gref},$ $Wr, Pg,$ $Vm$	$T_{gref},$ $Wg, Pg,$ $Vm$	$T_{gref},$ $Wg, Tg,$ $Vm$	$T_{gref},$ $Pg,$ $Vm$	$T_{gref},$ $B_{1ref}, B_1,$ $Vm$	$T_{gref},$ $B_{2ref}, B_2,$ $Vm$	$T_{gref},$ $B_{3ref}, B_3,$ $Vm$
Salida	$\hat{W}r$	$\hat{W}g$	$\hat{T}g$	$\hat{P}g$	$\hat{B}_1$	$\hat{B}_2$	$\hat{B}_3$
Neuronas en capa oculta	15	17	21	11	9	11	10
Normalización de los datos de entrada	0-10	0-10	-1-1	0-10	0-10	0-10	0-10

## 6.2. Generación de Residuos.

En su forma más simple un residuo se forma por la diferencia entre la salida real del sensor y la estimación, sin embargo este enfoque hace al sistema susceptible a falsas alarmas. En este trabajo se uso una forma filtrada propuesta en [18]:

$$r_k = \bar{w} \sum_{j=k-(\Omega-1)}^k \frac{(y_j - \hat{y}_j)^2}{\Omega} \quad (6.11)$$

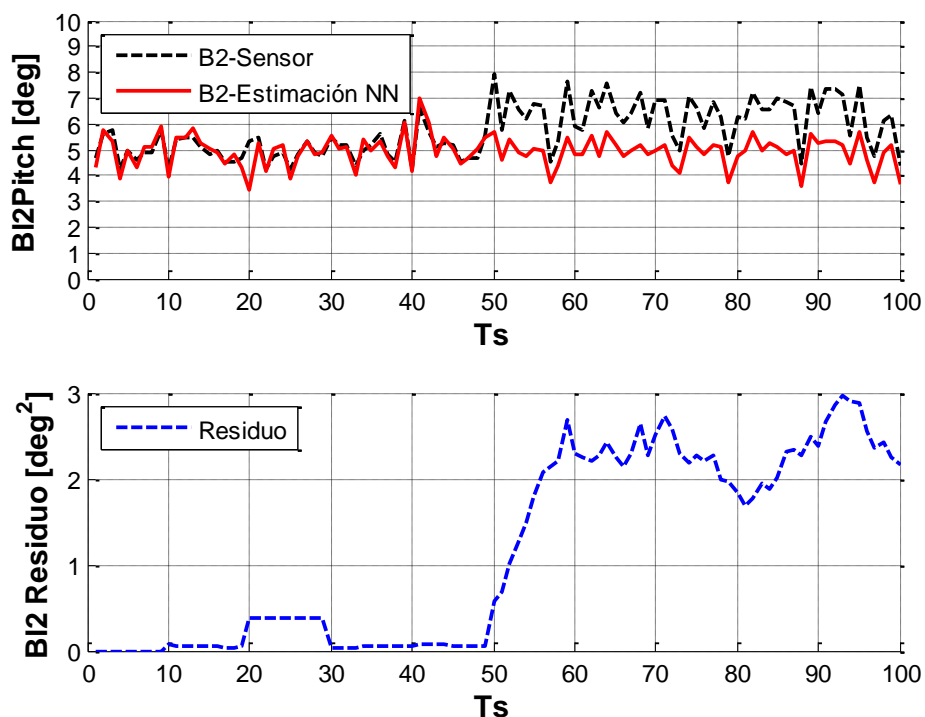
En esta fórmula para el cálculo del valor final del residuo  $r_k$  se emplea un filtro de medias móviles para los  $\Omega$  residuos pasados.  $\bar{w}$  cumple con la función de escalar la salida del valor del residuo final (en caso de que sea necesario).

Los valores de los  $\Omega$  de cada modelo se obtuvieron experimentalmente manteniendo un balance entre la sensibilidad a fallas y el número de falsas alarmas. Por su parte se uso  $\bar{w}$  para mantener los valores de todos los residuos en un rango común para facilitar su evaluación con los clasificadores. Los valores encontrados de estas variables se resumen en la Tabla 6.2:

En la Figura 6.5 se ejemplifica la generación de residuos para el caso de la Falla 2 de valor escalado en el sensor del pitch 2 introducida a los 50 tiempos muestrales de la simulación. En un ambiente ideal se obtendrían residuos diferentes de cero exclusivamente con la ocurrencia de una falla, sin embargo como ya se ha señalado el ruido y el error de estimación propio de los modelos pueden generar perturbaciones en los residuos.

**Tabla 6.2** Valores de  $\Omega$  y  $\bar{w}$  utilizados para la generación de residuos

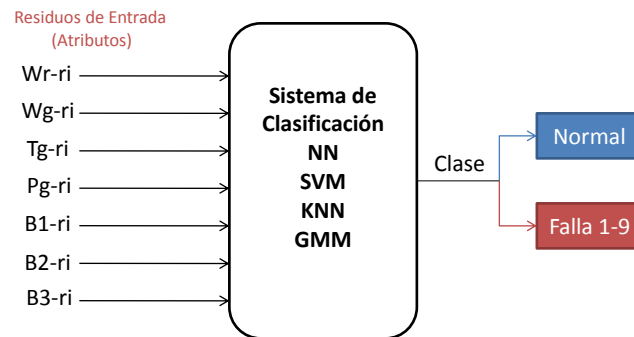
	$Wr - ri$	$Wg - ri$	$Tg - ri$	$Pg - ri$	$B_1 - ri$	$B_2 - ri$	$B_3 - ri$
$\bar{w}$	100	100	0.1	$10^{-4}$	1	1	1
$\Omega$	13	13	15	12	10	10	10

**Figura 6.5** Comportamiento de los residuos  $B_2 - ri$  (grafica inferior) al ser introducida una falla de valor escalado a los 50 Ts de ejecución en la señal  $B_2$  (grafica superior).

Determinar el umbral para el cual un residuo puede ser con certeza categorizado como falla puede ser una tarea ardua y requiere de amplia experimentación sobre el sistema. Como alternativa, en este trabajo de investigación se construirán clasificadores para la evaluación de los residuos, los algoritmos de SVM, NN, KNN y GMM serán empleados y se seleccionará aquel con mejor desempeño.

### 6.3. Evaluación de Residuos: Construcción de Clasificadores

El sistema de clasificación propuesto tiene como entrada todos los residuos generados en la etapa previa y debe identificar 10 diferentes clases (9 de Falla y una de comportamiento Normal),



**Figura 6.6** Evaluación de residuos mediante clasificadores.

es de esperarse mejoras notables en los niveles de detección teniendo en cuenta que se está simplificando la tarea de clasificar a dos escenarios posibles:

- Cuando se está en presencia de un comportamiento normal todos los residuos deben estar en niveles muy próximos a cero.
- Cuando hay algún tipo de falla los valores de uno o más de los atributos de entrada presentan valores mayores 0.

Para la construcción del set de entrenamiento se generaron secuencias de residuos para cada uno de los 9 tipos de Fallas. En un número igual se generaron residuos de comportamiento normal con el objeto de que el clasificador aprenda a diferenciar el ruido natural de las señales evaluadas. Un set total de 100.000 registros de residuos (80.000 de Fallas 1-9 y 20.000 de comportamiento normal) se empleó para el entrenamiento de los clasificadores. Los atributos de entrada al clasificador.

Para su posterior evaluación se construyeron 50 set de pruebas, cada uno correspondiente a una simulación de 1000 segundos (100.000 registros), los resultados promedios obtenidos por los algoritmos evaluados en términos de las métricas explicadas en el capítulo anterior se muestran en la Tabla 6.3.

Bajo el esquema propuesto las fallas 6,7 y 8 mejoran notablemente, la clase asociada al comportamiento normal llega a niveles del 99.8 lo cual indica una reducción notable de las falsas alarmas. En precisión global las GMM superan a las máquinas de vectores de soporte y redes neuronales. La sensibilidad para las fallas en sensores supera el 90% en todos los algoritmos, siendo las SVM las de mejor desempeño. No lograr un porcentaje del 100% de sensibilidad se debe en principio a el tiempo requerido para el levantamiento del residuo y su posterior clasificación, tal y como se ejemplifica en la Figura 6.7 con la ocurrencia de la Falla 2 (introducida a los 50 Ts). En esta figura

**Tabla 6.3** Evaluación de los clasificadores construidos para la evaluación de residuos

	Sensibilidad (%)										Presición
	F1	F2	F3	F4	F5	F6	F7	F8	F9	Normal	Global
SVM	98,4	98,2	96,06	98,7	95,3	75,47	82,5	98,5	87,6	99,8	93,8
NN	99,2	97,2	98,9	97,5	93,2	70,22	84,1	98,7	88,3	99,7	93,48
KNN	92,5	93,5	91,2	89,7	67,4	17,2	43,5	77,8	77,5	94,8	76,76
GMM	99,1	98,3	93,21	98,3	94,5	77,2	90,3	98,2	90,2	99,8	94,57

se ven todos los residuos de entrada al clasificador, sin embargo dada la naturaleza de la falla, el residuo  $B_2 - ri$  es el único que se incrementa. La detección se realiza cuando este residuo supera el umbral imaginario a los 53 TS, es decir 3 Ts después. Estos tiempos de levantamiento van en detrimento de la sensibilidad de cada clase y en últimas van a determinar el tiempo que tarda el sistema FDI en detectar una falla.

Se habla de umbral imaginario y no real porque los sistemas de clasificación empleados definen su salida en función de la relación (lineal o no lineal) de múltiples atributos de entrada y su predicción puede variar según las condiciones de cada uno de ellos. Para el caso expuesto en la Figura XX un clasificador de tipo SVM determino que el vector de entrada [ $B_1 - ri = 0,049$ ,  $B_2 - ri = 1,282$ ,  $B_3 - ri = 0,02$ ,  $Wr - ri = 0,1016$ ,  $Wg - ri = 0,1029$ ,  $Tg - ri = 0,72$ ,  $Pg - ri = 0,19$ ] pertenece a la clase Falla 2. Esta clasificación puede variar si por ejemplo se cambian el valor de  $Pg-ri$  a 1.16, en este caso la SVM determina que dicho vector pertenece a una Falla 6 que corresponde a una falla hidráulica en el pitch 2.

Finalmente dado que presenta la mayor precisión y una alta sensibilidad a cada una de las Fallas se selecciona para las GMM para la construcción final del sistema FDI híbrido, a continuación se entrará a realizar pruebas experimentales en término de las métricas definidas en el benchmark.

## 6.4. Pruebas Experimentales

Se utilizaron 50 set de datos de 4400 segundos de simulación (440.000 registros) previamente construidos para la evaluación del sistema FDI a dos niveles en el capítulo anterior. Cada set de datos contiene diferentes secuencias de Fallas Las métricas de evaluación empleadas fueron:

- Tiempo de detección (promedio -TDP-, máximo -TDMA- y mínimo -TDMI-): Tiempo que le toma al sistema FDI detectar la ocurrencia de una falla.
- Fallas no detectadas (FND): Número de fallas que no fueron detectadas.
- Falsas alarmas (FA): Número promedio de falsas alarmas.

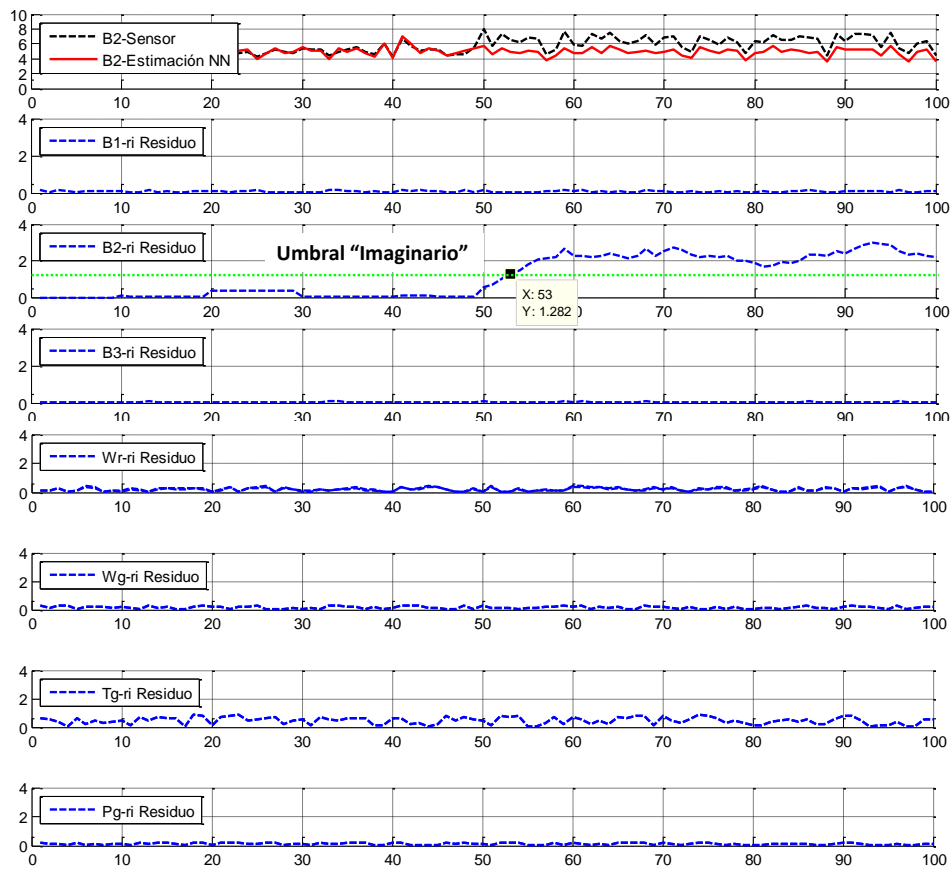


Figura 6.7 Ejemplo de conjunto de residuos de entrada al clasificador.



- Tiempo de recuperación de una falsa alarma (TR-FA): Tiempo promedio que tarda el sistema FDI en recuperarse posterior a la ocurrencia de una falsa alarma.

Cabe aclarar que en comparación con la arquitectura a dos niveles el modelo híbrido propuesto no sufre de falsos aislamientos. Los resultados obtenidos se muestran en la Tabla 6.4. De la comparación de estos con los obtenidos por el sistema FDI a dos Niveles se puede deducir:

- Hay una reducción notoria del número de falsas alarmas presentadas y se pueden reducir aun más con incrementos en el parámetro  $\Omega$  a costa de tiempo de detección. Con un valor de  $\Omega=100$  en todos los modelos de redes, las falsas alarmas se reducen a 0 con tiempos de detección entre 25 Ts y 45 Ts mayores a los presentados en la Tabla 6.4. Esto constituye una ventaja ya que fácilmente se puede variar este parámetro y adecuar la respuesta del sistema a las necesidades del problema.
- En el modelo híbrido se mejoraron los tiempos de detección de todas las fallas, a pesar de esto los problemas de oscilación entre zonas de operación de la turbina siguen afectando la velocidad de detección de las Fallas 6 y 7. Con la inclusión de las señales de control se mejora de forma notoria la detección de la Falla 8, de hecho dado que el offset se presenta directamente en el lazo de control es todo un reto detectar esta falla a partir de las señales del sistema de monitoreo tal y como se realizó en el sistema FDI a dos niveles.
- Un punto en contra del modelo híbrido propuesto es su lenta implementación, la construcción de los modelos de regresión con redes neuronales toma mucho más tiempo que la elaboración de los modelos de clasificación en el sistema FDI a dos niveles. El costo computacional también es superior al tener que emplear en paralelo 7 modelos de regresión, generadores de residuos y un clasificador versus los dos clasificadores y un contador de fallas de la arquitectura a dos niveles.

**Tabla 6.4** Resultados de las pruebas experimentales aplicadas al sistema FDI híbrido

Falla No.	TDP	TDMA	TDMI	FA	TR-FA	FND
1	3.7 Ts	5 Ts	2 Ts	0	–	0
2	3.5 Ts	5 Ts	2 Ts	0	–	0
3	3.5 Ts	5 Ts	2 Ts	0	–	0
4	3.5 Ts	5 Ts	2 Ts	1	2 Ts	0
5	5.2 Ts	6 Ts	2 Ts	1	3 Ts	0
6	5051 Ts	5053 Ts	5050 Ts	1	4 Ts	0
7	1578 Ts	1580 Ts	1575 Ts	2	3 Ts	0
8	5.5 Ts	7 Ts	3 Ts	0	–	0
9	983 Ts	1043 Ts	530 Ts	2	7 Ts	0

# Capítulo 7

## Conclusiones y Trabajo Futuro

En esta tesis, se construyeron diferentes sistemas FDI con enfoque 'data-driven' aplicados al dominio de la energía eólica para la detección y aislamiento de fallas en turbinas de viento. Los resultados obtenidos demuestran que bajo las arquitecturas propuestas y la aplicación de una metodología propia de minería de datos es posible obtener altos niveles de detección y un bajo número de falsas alarmas, por lo tanto de cara a su aplicación sobre sistemas industriales reales se muestra como una gran alternativa a las técnicas tradicionales basadas en modelos. De manera general la proposición anterior responde al objetivo primario de esta tesis, sin embargo en su consecución se abarcaron diferentes áreas cuya exploración y aplicación al problema específico dio como resultado un conjunto de conclusiones que se exponen a continuación.

### 7.1. Referentes a los sistemas FDI diseñados

- Los sistemas basados en clasificadores presentan por 'naturaleza' un gran número de falsas alarmas dado, en gran parte, a que realizan su predicción registro a registro sin tener en cuenta clasificaciones en tiempos muestrales pasados. Para solventar este problema se añadió a la salida del clasificador un contador de fallas en ventanas de tiempo, sin embargo su uso puede no ser adecuado para fallas de corta duración de carácter intermitente.
- Al separar la tarea de aislamiento en un segundo nivel se logro maximizar la sensibilidad global a fallas del sistema. Al dejar solo clases de falla se logro minimizar el error de clasificación y maximizar la precisión global.
- El sistema híbrido propuesto aprovecha las ventajas de detección de un esquema basado en residuos y las combina con la excelente capacidad de aislamiento de un clasificador. Si bien la construcción de modelos de regresión que permitan la correcta generación de residuos es una tarea ardua, se logro reducir significativamente el número de falsas alarmas en el

sistema. De acuerdo a los requerimientos establecidos en [1] el modelo híbrido es apto para su implementación en una turbina real.

- Independiente del sistema FDI diseñado siempre va existir un trade-off entre la sensibilidad a las fallas y el número de falsas alarmas.

## 7.2. Referente a las Fallas tratadas y las turbinas de viento

- Las Fallas 1-5 tienen tiempos de detección aceptables en ambos modelos (Benchmark y FAST) y en ambas arquitecturas (dos niveles e híbrido), pero solo es en el sistema FDI híbrido donde el número de falsas alarmas es adecuado para su implementación en sistemas industriales.
- Las Fallas 6 y 7 en los actuadores presentan tiempos de detección altos, esto se debe a que en el benchmark se simulan en las zonas 2 y 3 de operación de la turbina y es solo en la zona 3 cuando los pitch empiezan a moverse. La Falla 6 por ejemplo, durante los 50 segundos iniciales está en la región de operación 2, siendo imposible evidenciar la falla. En FAST el modelo de viento utilizado permitió simular todas las fallas en región 3 y como se esperaba los tiempos de detección mejoraron notablemente. En general es difícil identificar una falla en los pitch cuando la turbina está en zona de optimización de potencia.
- La Falla 9 por su parte, es detectable mediante clasificadores, de hecho se concluye que fallas incipientes de desarrollo lento en el tiempo son detectados en menor tiempo por los sistemas FDI propuestos en comparación con publicaciones basadas en modelos.

## 7.3. Referente a los algoritmos de clasificación

- Las SVM, NN y GMM presentaron un buen desempeño en los sistemas diseñados. En términos generales la mayor sensibilidad a fallas en sensores se logró con redes neuronales, mientras que las GMM presentaron alta sensibilidad a fallas en los actuadores.
- El algoritmo KNN si bien no presentó resultados sobresalientes sirvió de referente en las pruebas experimentales debido a su fácil implementación. Los resultados obtenidos por este algoritmo son un indicativo de alta no linealidad.

## 7.4. Trabajo Futuro

Como trabajo futuro en el área se plantea:

- 
- Implementación de los sistemas FDI diseñados sobre una turbina real.
  - Implementar estrategias automáticas de evolución de redes neuronales ('neuroevolución') para la obtención de la arquitectura óptima de la red.
  - Proponer nuevas estrategias o arquitecturas de sistemas FDI 'data-driven'.



# Bibliografía

- [1] P. F. Odgaard, J. Stoustrup, and M. Kinnaert, “Fault Tolerant Control of Wind Turbines - a Benchmark Model,” In *Proceedings of Fault Detection, Supervision and Safety of Technical Processes*, (2009).
- [2] N. Bergantino, F. Caponetti, and S. Longhi, “FaultBuster: data driven fault detection and diagnosis for industrial systems,” *Proceedings of 7th Workshop on Advanced Control and Diagnosis*, Zielona Gora, Poland (2009).
- [3] L. Wei, H. Wang, and F. Li, “Fault diagnosis of turbine generator vibration based on wavelet packet and data-driven,” In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, **2**, 29 –32 (2009).
- [4] Z. Wang, N. Wang, and W. Huan, “Fault Diagnosis of Turbine Generator Vibration Based on Supervision of Data-Driven,” In *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, **2**, 529 –532 (2009).
- [5] L. Datong, P. Yu, and P. Xiyuan, “Fault prediction based on time series with online combined kernel SVR methods,” In *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, pp. 1163 –1166 (2009).
- [6] C. A. Walford, “Wind turbine reliability: Understanding and minimizing windturbine operation and maintenance costs,” *Energy* pp. SAND2006–1100 (2006).
- [7] J. Ribrant, Master’s thesis, KTH School of Electrical Engineering, 2006.
- [8] A. S. Willsky, “A survey of design methods for failure detection in dynamic systems,” *AUTOMATICA* **12**, 601–611 (1976).
- [9] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems* (Kluwer Academic Publishers, Norwell, MA, USA, 1999).
- [10] J. Korbicz, C. Cempel, T. U. of Zielona Gora, and L. S. S. in Zielona Gora, *Analytical and Knowledge-based Redundancy in Fault Detection and Diagnosis* (Technical University Press, 1993).

- [11] P. M. Frank, "Analytical and Qualitative Model-based Fault Diagnosis - A Survey and Some New Results," *European Journal of Control* **2**, 6–28 (1996).
- [12] Y. Zhou, Ph.D. thesis, TEXAS A&M UNIVERSITY, 2004.
- [13] A. Kusiak and W. Li, "The prediction and diagnosis of wind turbine faults," *Renewable Energy* **36**, 16–23 (2011).
- [14] S. Verron, T. Tiplica, A. Kobi, and L. Istia, "Fault detection of univariate non-Gaussian data with Bayesian network," *Structure* pp. 56–61 (2010).
- [15] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD 03* **52**, 29 (2003).
- [16] A. Kusiak and S. Shah, "Data-mining-based system for prediction of water chemistry faults," 2006.
- [17] PATTON and CHEN, "Advances in computational intelligence for fault diagnosis systems," *International Journal of Applied Mathematics and Computer Science* **9**, 468–735 (1999).
- [18] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods: An introduction," *Control Engineering Practice* **5**, 639 – 652 (1997).
- [19] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods," *Computers Chemical Engineering* **27**, 293–311 (2003).
- [20] V. Venkatasubramanian, "A review of process fault detection and diagnosis Part II: Qualitative models and search strategies," *Computers Chemical Engineering* **27**, 313–326 (2003).
- [21] V. Venkatasubramanian, "A review of process fault detection and diagnosis Part III: Process history based methods," *Computers & Chemical Engineering* **27**, 327–346 (2003).
- [22] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control Systems Magazine* **8**, 3–11 (1988).
- [23] P. Frank and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control* **7**, 403–424 (1997).
- [24] R. J. Patton, "Robustness in model-based fault diagnosis: the 1995 situation," *Annual Reviews in Control* **21**, 103–123 (1997).
- [25] R. V. Beard, Ph.D. thesis, 1971.



- [26] R. N. C. R. N. Clark, “The dedicated observer approach to instrument failure detection,” 1979.
- [27] P. M. Frank, “ENHANCEMENT OF ROBUSTNESS IN OBSERVER-BASED FAULT-DETECTION,” *International Journal of Control* **59**, 955–981 (1994).
- [28] R. Patton, R. Clark, and P. Frank, in *Fault Diagnosis in Dynamic Systems: theory and applications*, R. J. Patton, P. M. Frank, and R. N. Clarke, eds., (Prentice Hall, 1989), p. 624.
- [29] R. J. Patton and J. Chen, “Observer-based fault detection and isolation: Robustness and applications,” *Control Engineering Practice* **5**, 671–682 (1997).
- [30] R. K. Mehra and J. Peschon, “An innovations approach to fault detection and diagnosis in dynamic systems,” *Automatica* **7**, 637–640 (1971).
- [31] A. S. Willsky, “A survey of design methods for failure detection in dynamic systems,” *Automatica* **12**, 601–611 (1975).
- [32] A. Zolghadri, “An algorithm for real-time failure detection in Kalman filters,” *IEEE Transactions on Automatic Control* **41**, 1537–1539 (1996).
- [33] M. S. N. Said, M. E. H. Benbouzid, and A. Benchaib, “Detection of broken bars in induction motors using an extended Kalman filter for rotor resistance sensorless estimation,” *IEEE Transactions on Energy Conversion* **15**, 66–70 (2000).
- [34] J. O’Reilly, *Observers for linear systems / J. O’Reilly* (Academic Press, London :, 1983), pp. xi, 246 p. :.
- [35] R. Seliger and P. M. Frank, “Fault-diagnosis by disturbance decoupled nonlinear observers,” 30th IEEE Conference on Decision and Control pp. 2248–2253 (1991).
- [36] R. Seliger and P. M. Frank., “Robust residual evaluation by threshold selection and a performance index for nonlinear observer-based fault diagnosis.,” *Int. Conf. on Fault Diagnosis (Tooldiag93)* (1993).
- [37] F. Hamelin and D. Sauter, “Robust fault detection in uncertain dynamic systems,” *Automatica* **36**, 1747 – 1754 (2000).
- [38] G. Bastin and M. R. Gevers, “Stable adaptive observers for nonlinear time-varying systems,” *IEEE Transactions on Automatic Control* **33**, 650–658 (1988).
- [39] J. Gertler and D. Singer, “A new structural framework for parity equation-based failure detection and isolation,” *Automatica* **26**, 381–388 (1990).

- [40] J. J. Gertler, Q. Luo, K. Anderson, and X. W. Fang, "Diagnosis of plant failures using orthogonal parity equations," Proc. of the 11th IFAC World Congress (1990).
- [41] J. Gertler, "Fault detection and isolation using parity relations," Control Engineering Practice **5**, 653–661 (1997).
- [42] J. J. Gertler and R. Monajemy, "Generating directional residuals with dynamic parity relations," Automatica **31**, 627–635 (1995).
- [43] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," IEEE Transactions on Automatic Control **29**, 603–614 (1984).
- [44] R. J. Patton and J. Chen, in *Preprints of IFACIMACS Symposium SAFEPROCESS91*, R. Isermann, ed., (IFAC, 1991), Vol. 1, pp. 239–255.
- [45] X. Ding and P. M. Frank, "Fault detection via factorization approach," Systems Control Letters **14**, 431–436 (1990).
- [46] P. M. Frank and X. Ding, "Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis," Automatica **30**, 789–804 (1994).
- [47] J. Chen and R. J. Patton, in *Robust model-based fault diagnosis for dynamic systems*, K.-Y. Cai and C. Beijing University Of Aeronautics Beijing, eds., (Kluwer Academic Publishers, 1999), No. 14, p. 356.
- [48] M. Schlechtingen and I. Santos, "Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection," Mechanical Systems and Signal Processing **25**, 1849–1875 (2011).
- [49] M. Iri, K. Aoki, E. O'Shima, and H. Matsuyama, "An algorithm for diagnosis of system failures in the chemical process," Computers & Chemical Engineering **3**, 489 – 493 (1979).
- [50] T. Umeda, T. Kuriyama, E. O'Shima, and H. Matsuyama, "A graphical approach to cause and effect analysis of chemical processing systems," Chemical Engineering Science **35**, 2379 – 2388 (1980).
- [51] J. Shiozaki, H. Matsuyama, E. OShima, and M. Iri, "An Improved Algorithm for Diagnosis of System Failures in the Chemical Process," Computers chem Engng **3**, 489–493 (1985).
- [52] M. Kokawa, S. Miyazaki, and S. Shingai, "Fault location using digraph and inverse direction search with application," Automatica **19**, 729 – 735 (1983).

- [53] M. A. Kramer and B. L. Palowitch Jr., "A rule-based approach to fault diagnosis using the signed directed graph," *AICHE Journal* **33**, 1067–1078 (1987).
- [54] O. Oyeleye and M. A. Kramer, "Qualitative Simulation of Chemical Process Systems : Steady-State Analysis," *AICHE Journal* **34**, 1441–1454 (1988).
- [55] C.-c. Han, R.-f. Shih, and L.-s. Lee, "Quantifying Signed Directed Graphs with the Fuzzy Set for Fault Diagnosis Resolution Improvement," *Industrial & Engineering Chemistry Research* **33**, 1943–1954 (1994).
- [56] R. F. Li and X. Z. Wang, "Qualitative / Quantitative Simulation of Process Temporal Behavior Using Clustered Fuzzy Digraphs," *AICHE Journal* **47** (2001).
- [57] A. Hunt, B. E. Kelly, J. S. Mullhi, F. P. Lees, and A. G. Rushton, "The propagation of faults in process plants : 10 , Fault tree synthesis-2," *Science* **39**, 243–250 (1993).
- [58] N. Ulerich and G. Powers, "On-line hazard aversion and fault diagnosis in chemical processes: the digraph+fault-tree method," *Reliability, IEEE Transactions on* **37**, 171 –177 (1988).
- [59] M. W. Averett, "Fault Tree Analysis," *Risk Analysis* **8**, 463–464 (1988).
- [60] S. A. Lapp and G. J. Powers, "Computer-Aided Synthesis of Fault Trees," *IEEE Trans on Reliability* (1977).
- [61] S. Soylemez and W. D. Seider, "A new technique for precedence-ordering chemical process equation sets," *AICHE Journal American Institute of Chemical Engineers* **19**, 934–942 (1973).
- [62] J. De Kleer and J. S. Brown, "A qualitative physics based on confluences," *Artificial Intelligence* **24**, 7–83 (1984).
- [63] Y. Iwasaki and H. A. Simon, "Causality in device behavior," *Artificial Intelligence* **29**, 3–32 (1986).
- [64] O. Raiman, "Order of magnitude reasoning," *Artif. Intell.* **51**, 11–38 (1991).
- [65] M. L. Mavrovouniotls and G. Stephanopoulos, "Reasoning with orders of magnitude and approximate relations," In *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, AAAI'87 pp. 626–630 (AAAI Press, 1987).
- [66] H. Kay and B. Kuipers, "Numerical behavior envelopes for qualitative models," In *Proceedings of the eleventh national conference on Artificial intelligence*, AAAI'93 pp. 606–613 (AAAI Press, 1993).

- [67] S. Grantham and L. Ungar, “A first principles approach to automated troubleshooting of chemical plants,” *Computers & Chemical Engineering* **14**, 783 – 798 (1990).
- [68] S. D. Grantham and L. H. Ungar, “Comparative analysis of qualitative models when the model changes,” *AIChE Journal* **37**, 931–943 (1991).
- [69] B. Falkenhainer and K. D. Forbus, “Compositional modeling: finding the right model for the job,” *Artificial Intelligence* **51**, 95–143 (1991).
- [70] J. V. Kresta, J. F. Macgregor, and T. E. Marlin, “Multivariate statistical monitoring of process operating performance,” *The Canadian Journal of Chemical Engineering* **69**, 35–47 (1991).
- [71] J. MacGregor and T. Kourti, “Statistical process control of multivariate processes,” *Control Engineering Practice* **3**, 403 – 414 (1995).
- [72] S. Watanabe, *Pattern recognition: human and mechanical* (John Wiley & Sons, Inc., New York, NY, USA, 1985).
- [73] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **22**, 4–37 (2000).
- [74] S. M. Weiss and N. Indurkha, *Predictive data mining: a practical guide* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998).
- [75] I. H. Witten and E. Frank, in *Data Mining: Practical Machine Learning Tools and Techniques, The Morgan Kaufmann Series in Data Management Systems*, 2nd ed., J. Gray, ed., (Morgan Kaufmann Publishers, San Francisco, CA, 2005).
- [76] F. Gorunescu, *Data mining concepts, models and techniques* (Springer, Berlin; Heidelberg, 2011).
- [77] R. S. Michalski, I. Bratko, and A. Bratko, in *Machine Learning and Data Mining; Methods and Applications*, R. S. Michalski, I. Bratko, and A. Bratko, eds., (John Wiley & Sons, Inc., New York, NY, USA, 1998).
- [78] J. Han and M. Kamber, *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*, 1st ed. (Morgan Kaufmann, 2000).
- [79] M. A. T. Figueiredo and A. K. Jain, “Unsupervised Learning of Finite Mixture Models,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **24**, 381–396 (2000).

- [80] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*, illustrated edition ed. (SIAM, Society for Industrial and Applied Mathematics, 2007).
- [81] G. Karypis, E.-H. Han, and V. Kumar, “Chameleon: hierarchical clustering using dynamic modeling,” *Computer* **32**, 68–75 (1999).
- [82] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, 2 ed. (Wiley-Interscience, 2001).
- [83] T. N. Tran, R. Wehrens, and L. M. Buydens, “KNN-kernel density-based clustering for high-dimensional multivariate data,” *Computational Statistics & Data Analysis* **51**, 513 – 525 (2006).
- [84] S. Guha, R. Rastogi, and K. Shim, “Cure: an efficient clustering algorithm for large databases,” *Information Systems* **26**, 35 – 58 (2001).
- [85] M. Ester, H.-p. Kriegel, S. Jörg, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *KDD-96 Proceedings* pp. 226–231 (1996).
- [86] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)* (Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005).
- [87] J. A. Bilmes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,” Technical report (1998) .
- [88] V. N. Vapnik, *Statistical learning theory*, 1 ed. (Wiley, 1998).
- [89] S. Haykin, *Neural Networks and Learning Machines*, No. v. 10 in *Neural networks and learning machines* (Prentice Hall, 2009).
- [90] E. Lughofer, H. Efendic, L. Del Re, and E. Klement, “Filtering of dynamic measurements in intelligent sensors for fault detection based on data-driven models,” In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, **1**, 463 – 468 Vol.1 (2003).
- [91] A. Kusiak and A. Verma, “A Data-Driven Approach for Monitoring Blade Pitch Faults in Wind Turbines,” *Sustainable Energy, IEEE Transactions on* **2**, 87 –96 (2011).
- [92] K. E. Johnson, L. Y. Pao, M. J. Balas, and L. J. Fingersh, “STANDARD AND ADAPTIVE TECHNIQUES FOR MAXIMIZING ENERGY CAPTURE,”.
- [93] M. L. B. J. Jason M. Jonkman, “FAST User’s Guide,” Technical report, NREL/EL-500-38230 (2005) .

- [94] B. Jonkman., “TurbSim User’s Guide: Version 1.50.,” Technical report, NREL/EL-500-46198 (2009) .
- [95] N. Kasdin, “Discrete simulation of colored noise and stochastic processes and 1/f alpha; power law noise generation,” *Proceedings of the IEEE* **83**, 802 –827 (1995).
- [96] W. M. J. Jonkman, S. Butterfield and G. Scott, “Definition of a 5MW Reference Wind Turbine for Offshore System Development,” Technical report, NREL/TP-500-38060 (2009) .
- [97] K. Johnson and P. Fleming., “Development, implementation, and testing of fault detection and condition monitoring on the national wind technology center’s controls advanced research turbines.,” *Mechatronics* pp. 21(4):728–736 (2011).
- [98] S. Zhao, Master’s thesis, Wright State University, 2011.
- [99] D. W. Zimmerman, “A note on the influence of outliers on parametric and nonparametric tests,” *Journal of Experimental Education* pp. 391–401 (1994).
- [100] D. Hawkins, *Identification of Outliers, Monographs on Applied Probability and Statistics* (Chapman and Hall, 1980).
- [101] X. Yao, “Evolving Artificial Neural Networks,” 1999.
- [102] A. Lipnickas, “Two Stage Neural Networks Based Classifier System for Fault Diagnosis,” in *Computational Intelligence in Fault Diagnosis, Advanced Information and Knowledge Processing*, V. Palade, L. Jain, and C. D. Bocaniala, eds., (Springer London, 2006), pp. 209–230.
- [103] A. A. Ozdemir, P. Seiler, and G. J. Balas, “Wind Turbine Fault Detection Using Counter-Based Residual Thresholding,” *Proceedings of the 18th IFAC World Congress, 2011* pp. 8289–8294 (2011).
- [104] Y. Li, N. Sundararajan, and P. Saratchandran, “Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems,” *Control Theory and Applications, IEE Proceedings* - **147**, 476 –484 (2000).