CORPORACIÓN MEXICANA DE INVESTIGACIÓN EN MATERIALES S.A. DE C.V.

DETECCIÓN DE FALLAS Y SELECCIÓN DE VARIABLES EN UN PROCESO MULTIVARIADO USANDO ANÁLISIS DE SEÑALES Y SISTEMAS INTELIGENTES

DOCTORADO EN CIENCIA Y TECNOLOGÍA

DIRECTOR DE TESIS: Dr. DAVID SALVADOR GONZÁLEZ GONZÁLEZ

CODIRECTOR DE TESIS: Dr.rer.nat. EDGAR O. RESÉNDIZ FLORES

ALUMNO: JESÚS ALEJANDRO NAVARRO ACOSTA

Saltillo, Coahuila, México 11 de enero de 2018

A mis padres,
a mi esposa,
hermanos y amigos.

Agradecimientos

Agradezco a Dios por permitirme conocer a tantas personas valiosas, de las cuales he recibido amor, cariño y amistad incondicionales.

Mi profundo agradecimiento a mis padres María Magdalena Acosta Montes y Enrique Navarro Cedillo, a mis hermanos Héctor, Juan, Mayela y Gabriela por todo su amor y cuidados, por el apoyo y motivación que de alguna u otra forma me brindaron.

A mi esposa Alejandra Castañeda Gonzales por darme todo su amor y comprensión, por apoyarme en los momentos complicados de mi vida, y sobre todo por creer en mí y en mi capacidad como estudiante.

Agradezco a la familia Castañeda Gonzáles por abrirme las puertas de su casa y permitirme compartir preciados momentos haciéndome sentir siempre parte de su familia.

A mis amigos Galo, Marco, David, Eduardo y Raul por la verdadera amistad brindad hacia mí. A los doctores Pedro Pérez y David González, por ser partícipes de mi formación y por dirigir mi trabajo de investigación.

Mi completa gratitud al Dr. Edgar Omar Reséndiz Flores por co dirigir con total dedicación y profesionalismo mi trabajo de tesis, pero sobre todo por motivarme y confiar en mi capacidad, por ampliar mi perspectiva académica y despertar mi interés por la ciencia. Le expreso de la manera más sincera que lo considero un maestro pero sobre todo un amigo.

Al personal administrativo del posgrado de COMIMSA por su apoyo y comprensión para con mi manera de trabajo. Agradezco los apoyos recibidos durante mi formación de doctorado a dos importantes Instituciones; sin ellas mi progreso y desarrollo académico, actividades de movilidad e investigación y producción científica no hubieran sido posibles: El Consejo Nacional en Ciencia y Tecnología (CONACYT) y a la Corporación Mexicana de Investigación en Materiales (COMIMSA).

Resumen

Uno de los grandes retos que enfrenta el sector industrial es alcanzar un nivel óptimo en la calidad de sus productos. Con el fin de lograrlo, se implementa una gran variedad de metodologías y técnicas para llevar a cabo procedimientos de detección de fallas, y así asegurar el buen funcionamiento de los procesos industriales.

Una falla se define como una desviación no autorizada de al menos una característica o variable del sistema. Los tipos de fallas que ocurren en sistemas industriales incluyen cambios de parámetros de proceso, perturbaciones, problemas de actuadores, problemas con sensores, entre otros. Sin embargo debido al gran avance de la tecnología, los procesos se han vuelto complejos presentando características lineales y no lineales, presencia de ruido y correlación entre otras.

Por tal motivo, diversas técnicas de machine learning han sido aplicadas eficientemente para la detección de fallas en diversos procesos debido a su alta capacidad de llevar a cabo tareas de reconocimiento de patrones. Sin embargo en sistemas de alta dimensionalidad, es decir conjuntos de datos con gran cantidad de variables, puede existir información redundante la cual aumenta de manera innecesaria la complejidad e inexactitud de los modelos matemáticos para la detección de fallas. Por lo tanto, es necesario llevar a cabo la selección de características relevantes ya que nos proporciona una forma de reducir el tiempo computacional, mejorar el desempeño de la predicción y una mejor comprensión de los datos en aplicaciones de machine learning.

Por otro lado, en aplicaciones de clasificación en la vida real, los conjuntos de datos empleados para el entrenamiento de los algoritmos sufren el problema de desequilibrio de clases. Es decir, cuando un conjunto de datos está dominado por una clase principal o por clases que tienen significativamente más instancias que las otras clases minoritarias en los datos. Esto ocasiona que los clasificadores tradicionales presenten deficiencias al

momento de predecir a que clase pertenece un nuevo conjunto de datos.

Como se mostró, existe una serie de aspectos a considerar al momento de llevar a cabo la detección de fallas en procesos complejos. En este trabajo de investigación se presenta una metodología para la detección de fallas en procesos multivariados, la cual lleva a cabo un procedimiento para limpieza de datos y selección de características. Dicha metodología es implementada en un proceso industrial de moldeo por inyección de plástico. Así como en un proceso de maquinado de cabeza de motor.

Palabras clave: detección de fallas, selección de características, limpieza de señales, clases no balanceadas, procesos industriales multivariados.

Campo específico: inteligencia artificial, análisis de señales.

Índice general

1. Introducción **1**

2. Planteamiento del problema **5**

2.1. Problemas específicos 6

2.2. Estado del arte 7

2.2.1. Clasificación 7

2.2.2. Detección de fallas y selección de características 11

2.3. Preguntas de investigación 17

2.4. Hipótesis 17

2.5. Objetivos 18

2.5.1. Objetivo General 18

2.5.2. Objetivos específicos 18

2.6. Justificación 18

2.7. Alcances y limitaciones 18

3. Marco Téorico **20**

3.1. Transformada Wavelet 20

3.2. Multirresolución 24

3.2.1. Análisis multirresolución 24

3.3. Limpieza de señales usando Wavelet 29
3.3.1. Análisis de componentes principales (PCA) 30
3.3.2. Limpieza de señales multuvariada usando wavelets y PCA 31
3.4. Correlación 32
3.4.1. Coeficiente de correlación lineal de Pearson 32
3.4.2. Correlación no paramétrica: coeficiente de Kendall 33
3.4.3. Significancia del coeficiente de correlación 34
3.5. Inteligencia artificial 35
3.6. Redes neuronales artificiales (ANN) 36
3.7. Teoría del aprendizaje estadístico (SLT) 37
3.7.1. Problema de aprendizaje 37
3.7.2. Minimización del riesgo general 38
3.7.3. Minimización del riesgo empírico 38
3.7.4. Dimensión Vapnik-Chervonenkis (VC) 39
3.7.5. Límites del riesgo general 40
3.7.6. Principio de Inducción de Minimización del Riesgo Estructural . . . 40
3.8. Máquinas de soporte vectorial (SVM) 41
3.8.1. Caso linealmente separable. 41
3.8.2. Caso no linealmente separable. 43
3.9. Support Vector Data Description (SVDD) 45
3.9.1. SVDD con ejemplos negativos 48
3.10. Distancia Mahalanobis en algoritmos de vectores de soporte 49
3.10.1. Distancia Mahalanobis (MD) 50
3.10.2. Distancia Mahalanobis en el espacio de referencia kernel 51

3.10.3. Mahalanobis Data Description Machine (MDDM) 52
3.11. Funciones kernel 53
3.12. Construcción de kernels 54
3.12.1. Kernels básicos 54
3.12.2. Operaciones en kernels 54
3.12.3. Kernels usados 55
3.13. Problema de optimización 56
3.13.1. Programación no lineal 57
3.13.2. Programación cuadrática 59
3.13.3. Métodos de punto interior para optimización 61
3.14. Error en clasificadores 64
3.14.1. Matriz de confusión 64
3.14.2. Error en clasificadores de una clase 64
3.14.3. Validación cruzada de K subconjuntos (k-fold cross validation) . . . 65
3.15. Selección de características 66
3.15.1. Tipo filtro 66
3.15.2. Tipo envoltura 67
3.15.3. Soluciones integradas 67
3.16. Metaheurísticas 67
3.16.1. Optimización por enjambre de partículas (PSO) 68
3.16.2. PSO binario 70
3.16.3. PSO binario Gompertz (GBPSO) 71
3.16.4. Algoritmo genético 73

4. Metodología **74**

4.1. Pre procesamiento 76

4.2. Aprendizaje 77

4.2.1. Entrenamiento de SVDD para detección de fallas 78

4.3. Monitoreo y prueba 80

5. Experimentación **81**

5.1. Bases de datos de referencia 81

5.1.1. Conjunto de datos: iris de Fisher 81

5.1.2. Base de datos cáncer de mama 84

5.2. Aplicaciones en procesos industriales 85

5.2.1. Moldeo por inyección de plástico 86

5.2.2. Maquinado de cabeza de motor 93

6. Conclusiones y trabajo futuro **98**

Referencias 102

Índice de figuras

3.1. Dilatación diádica de la función seno y la wavelet Db2. 26

3.2. Traslación de la wavelet Db2. 27

3.3. Transformación no lineal. 44

3.4. Hiperesfera y margen 45

3.5. Convexidad de funciones. 58

3.6. Posibles situaciones de clasificación para algoritmos OCC. 65

3.7. Función Gompertz 72

4.1. Metodología. 76

4.2. Partícula GBPSO-SVDD. 77

4.3. Funcionamiento de GBPSO-SVDD 79

5.1. Limpieza Wavelet en la señal valor punta contrapresión (x_f^{17}). 88

5.2. Convergencia de GBPSO, BPSO and GA para caso 1). 90

5.3. Convergencia de GBPSO, BPSO and GA para caso 2). 90

5.4. Convergencia de GBPSO, BPSO and GA para caso 3). 91

5.5. Gráfico de detección de fallas (caso 1) 92

5.6. Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 1). 95

5.7. Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 2). 95

5.8. Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 3). 96

5.9. Gráfico de detección de fallas (caso 2) 97

Índice de tablas

5.1. Valores posibles para C y s. 82

5.2. Resultados de GBPSO-SVDD y GBPSO-MDDM en iris de Fisher. 82

5.3. Valores de la matriz de correlación coeficiente de Pearson. 83

5.4. Valores p para coeficientes de Pearson 83

5.5. Valores de la matriz de correlación coeficiente de Kendall 83

5.6. Valores p para coeficientes de Kendall 83

5.7. Resultados de GBPSO-SVDD y GBPSO-MDDM en cáncer de mama . . . 85

5.8. Comparación de resultados 85

5.9. Descripción de variables 87

5.10. Variables correlacionadas caso 2. 88

5.11. Valores posibles para C y s. 89

5.12. Valores de clasificación errónea 89

5.13. Variables seleccionadas. 91

5.14. Hiperparámetros seleccionados. 91

5.15. Tiempo computacional en fase de entrenamiento (segundos). 92

5.16. Descripción de variables cabeza de motor. 93

5.17. Variables correlacionadas 94

5.18. Valores de clasificación errónea 94

5.19. Variables seleccionadas. 94

5.20. Hiperparámetros seleccionados. 96

5.21. Tiempo computacional en fase de entrenamiento (segundos). 96

1

Introducción

Uno de los grandes retos que enfrenta el sector industrial es alcanzar un nivel óptimo en la calidad de sus productos. Con el fin de lograrlo se implementa una gran variedad de metodologías y técnicas para la eficiente detección de fallas, y así asegurar el buen funcionamiento de los procesos involucrados en la producción de dichos productos. Sin embargo, debido a la creciente competitividad en los mercados y al gran avance de la tecnología, los procesos se han vuelto altamente complejos presentando características lineales y no lineales, presencia de ruido y correlación entre otras. Esto propicia que algunas técnicas de análisis convencionales queden obsoletas, por lo tanto es imperante analizar estos procesos haciendo uso de metodologías innovadoras y flexibles que permitan llevar acabo el análisis de procesos complejos.

En la actualidad la investigación y desarrollo de técnicas de aprendizaje automático o machine learning (ML)[1] ha crecido exponencialmente gracias al avance de los sistemas computacionales. El reconocimiento de patrones es una de las tareas que ML es capaz de llevar a cabo de manera eficiente en ambientes complejos mediante técnicas como redes neuronales artificiales, máquinas de soporte vectorial, algoritmo k-medias, etc. Por lo tanto dichas técnicas se han aplicado con éxito en la detección de fallas en diversos procesos y dispositivos tecnológicos.

Una falla se define como una desviación no autorizada de al menos una característica o variable del sistema. Los tipos de fallas que ocurren en sistemas industriales incluyen

[1] En este documento algunos conceptos, abreviaturas y siglas se presentan en inglés debido a su amplio uso en la disiplina correspondiente.

cambios de parámetros de proceso, perturbaciones, problemas de actuadores, problemas con sensores, entre otras(Chiang, Russell, y Braatz, 2001). De acuerdo a Vapnik machine learning consta de tres problemas principales: clasificación, regresión y estimación de la densidad. El objetivo del problema de regresión es predecir el valor asociado con una muestra x basada en los datos observados, donde la relación existe pero es inobservable. La estimación de la densidad tiene como objetivo construir una estimación basada en los datos observados, de la función de densidad de probabilidad subyacente. Mientras que la tarea general del problema de clasificación es construir una máquina automática (algoritmo) que pueda clasificar una muestra en un conjunto de clases dadas sin implicación humana.

Dichos algoritmos pueden caracterizarse por una familia de funciones $f(x, \alpha), \alpha \in \Lambda$ donde x es la muestra, y Λ es el dominio de los parámetros. El proceso de aprendizaje tiene como objetivo la nominación del parámetro óptimo α_0 que minimiza el riesgo general o empírico. De acuerdo a la teoría del aprendizaje estadístico desarrollada por Vapnik y Chervonenkis, el proceso de nominación de parámetros se lleva a cabo mediante principios de aprendizaje. Estos principios proporcionan una ruta para la adquisición del parámetro óptimo y también garantizan la coherencia y calidad de la solución, así como la tasa de convergencia. En teoría, el riesgo empírico o general son criterios adecuados para obtener el parámetro α_0. El riesgo empírico se refiere a un conjunto de muestras de entrenamiento. Es posible calcular el riesgo empírico para el conjunto de entrenamiento dado. Por el contrario, no es factible evaluar el riesgo general que se define en todo el espacio de datos (Le, 2012). En las secciones siguientes se definirán otros conceptos de la teoría de aprendizaje estadístico.

Como se mencionó anteriormente diversas técnicas de machine learning han sido aplicadas eficientemente para la detección de fallas en diversos procesos debido a su alta capacidad de llevar a cabo tareas de reconocimiento de patrones. Sin embargo en sistemas de alta dimensionalidad, es decir conjuntos de datos con gran cantidad de variables, puede existir información redundante la cual aumenta de manera innecesaria la complejidad e inexactitud de los modelos matemáticos para la detección de fallas, además de volverlos computacionalmente costosos. Por lo tanto es necesario reducir el número de variables a considerar manteniendo solo las características relevantes y útiles con la mínima perdida de información valiosa, a este proceso se le conoce como selección de características. La selección de características nos proporciona una forma de reducir el tiempo computacional, mejorar el desempeño de la predicción y una mejor comprensión de los datos en

aplicaciones de machine learning.

En aplicaciones de clasificación en la vida real, los conjuntos de datos empleados para el entrenamiento de los algoritmos, pueden contener cientos de variables de las cuales una gran cantidad de ellas pueden estar altamente correlacionadas. Por ejemplo, si dos de estas variables estan perfectamente correlacionadas solo se necesita una de ellas para describir los datos, es decir que las variables dependientes no proporcionan ninguna información adicional sobre las clases y por lo tanto sirven como ruido para el predictor. Esto significa que el contenido total de información se puede obtener a partir de menos características únicas que contienen información de discriminación máxima sobre las clases (Chandrashekar y Sahin, 2014). Además del reconocimiento de patrones y la selección de características en procesos de alta dimensionalidad, es necesario tomar en cuenta algunos otros problemas que se presentan de manera común en dichos procesos. Un ejemplo de estos es el conocido como desequilibrio de clases (class imbalance). Se dice que los datos sufren el problema de desequilibrio de clase cuando las distribuciones de las mismas están altamente desequilibradas. Esto ocurre cuando un conjunto de datos está dominado por una clase principal o por clases que tienen significativamente más instancias que las otras clases minoritarias en los datos.

Para el caso de dos clases, se supone que la clase minoritaria es la clase positiva, y la clase mayoritaria es la clase negativa. Si se aplican los clasificadores tradicionales al conjunto de datos, es probable que predigan todo como negativo (la clase mayoritaria). Otro de los problemas que se presentan al momento de llevar a cabo análisis de datos es el ruido presente en los mismos (noisy data). Esto puede deberse a dispositivos de medición defectuosos, errores de transcripción e irregularidades, entre otras cosas. Sin embargo, la precisión de un clasificador puede depender en gran medida de la calidad de los datos utilizados durante la fase de entrenamiento, por lo que un clasificador construido a partir de un conjunto de entrenamiento ruidoso podría ser menos preciso que uno construido a partir de la versión libre de ruido utilizando un algoritmo idéntico (Bolón-Canedo et al., 2015).

Como se mostró, existe una serie de aspectos a considerar al momento de llevar a cabo la detección de fallas en procesos complejos, por lo tanto es necesario el desarrollo e implementación de metodologías robustas capaces de lidiar con dichos inconvenientes y así aplicarlos de manera eficiente en problemas de la vida real. Por lo tanto, este trabajo de investigación se centrará en desarrollar una metodología para la detección de fallas en

procesos multivariados.

2

Planteamiento del problema

En la industria manufacturera actual el aseguramiento de la calidad es un aspecto fundamental con el que toda organización se enfrenta día a día. Por tal motivo la detección de fallas en procesos, equipos, maquinaria, entre otros. desempeña un papel crucial para lograr que los productos o servicios cumplan con los requisitos mínimos de calidad. Con el avance de la tecnología informática y el desarrollo de nuevas metodologías científicas, han surgido diferentes enfoques para llevar a cabo la detección de fallas. Uno de ellos es la implementación de técnicas de inteligencia artificial que permiten analizar grandes cantidades de datos donde interactúan características de diversa naturaleza. Clasificadores como máquinas de soporte vectorial (SVM), redes neuronales artificiales (ANN), entre otros, han sido implementados con éxito en tareas de detección de fallas en ambientes industriales complejos, donde el dominio de las características se ha expandido de decenas a cientos de variables o características involucradas en dichas tareas.

Debido a que el rendimiento de la mayoría de los clasificadores mejora cuando se eliminan las características redundantes o irrelevantes, es necesario implementar procedimientos para la selección de variables enfocados en seleccionar un subconjunto de variables de entrada que pueden describir eficientemente el conjunto total de características mientras se reducen los efectos del ruido o variables irrelevantes y así proporcionar buenos resultados de clasificación, con el fin de desarrollar sistemas eficientes para la detección de fallas. Además de la gran cantidad de variables presentes en los procesos actuales, otro inconveniente para algunas técnicas de clasificación es el lidiar con datos desequilibrados, es decir cuando se cuenta con un buen muestreo solo para algunas de las clases mientras

que para el resto se cuenta con un muestreo pobre o incluso inexistente.

Un ejemplo de esta situación sería un equipo nuevo donde aun no se cuenta con información alguna de fallas de la misma. En esta situación, los clasificadores como SVM y ANN pueden tener u na buena precisión en la clase mayoritaria pero una precisión muy pobre en la clase minoritaria debido a la influencia que la clase mayoritaria tiene sobre los criterios tradicionales de entrenamiento. En estos casos se han implementado los llamados clasificadores de una clase para llevar a cabo tareas de reconocimiento de patrones en conjuntos de datos desequilibrados. Debido a lo antes descrito esta investigación se centrará en investigar y lidiar con los inconvenientes que generan los distintos escenarios planteados al implementar la detección de fallas en un proceso industrial complejo.

2.1. Problemas específicos

- Pre procesamiento de datos: al obtener datos directamente de procesos industriales complejos, estos pueden presentar características como: datos faltantes, información redundante y ruido entre otras. Esto dificulta el proceso de clasificación de algunas técnicas de reconocimiento de patrones aplicadas a la detección de fallas. Por lo tanto es importante implementar una etapa de pre procesamiento de los datos, y así reducir los efectos negativos de dichas características.

- Detección de fallas: debido a que los procesos se han vuelto altamente complejos presentando características no lineales, gran cantidad de variables y correlación entre las mismas. Es necesario implementar técnicas robustas que permitan reconocer patrones de manera eficiente y así llevar a cabo la detección de fallas.

- Selección de características: es necesario implementar técnicas para la selección de características ya que proporcionan una forma de reducir el tiempo computacional, mejorar el desempeño en el reconocimiento de patrones y una mejor comprensión de los datos en aplicaciones de detección de fallas.

2.2. Estado del arte

Se presenta la revisión de la producción científica concerniente a las diversas técnicas de inteligencia artificial y análisis de señales para la detección de fallas y selección de características.

2.2.1. Clasificación

Una de las aplicaciones mas importantes y de mayor utilidad de la inteligencia artificial es la clasificación. Por ejemplo, en procesamiento de imágnes, las técnicas de inteligencia artificial (IA) se usan con el objetivo de llevar a cabo tareas como reconocer rostros humanos, huellas digitales, digitos alfanuméricos entre otros. Otro campo de aplicación es la detección de fallas, por lo tanto estas metodologías son relevantes en este trabajo de investigación.

(Elhariri et al., 2014) presentan un sistema para la clasificación de tomates dependiendo de su estado de madurez mediante el análisis de imágenes. El sistema consta de tres etapas: pre procesamiento de imágenes, extracción de características y por último la etapa de clasificación. La primera etapa consiste en convertir cada imagen de RGB al espacio de color HSV, ya que es ampliamente utilizado en el campo de visión a color. En la etapa de extracción de características se implementa el analisis de componentes principales (ACP) para generar un vector de características por cada imagen. Finalmente, se utiliza máquinas de soporte vectorial multiclase para llevar a cabo la tarea de clasificar los tomates de acuerdo a su estado de madurés.

(Xie et al., 2013) proponen la hiperesfera multiclase usando support vector data description (HSMC-SVDD por sus siglas en ingles) para mejorar la exactitud en la clasificación y la velocidad de entrenamiento cuando existen más de dos clases. Primero se implementa un proceso de extracción de características donde se obtienen diversas métricas como curtosis, energía y correlación. Después se aplica análisis de componentes principales no lineal con el fin de realizar una reducción dimensional de los datos. Finalmente, se construye el modelo HSMC-SVDD para la clasificación. Las pruebas se realizan en un total de 322 imágenes médicas para el diagnóstico de masas normales, masas benignas y masas malignas. (Zhao et al., 2014) proponen un marco simple, no invasivo y automatizado para la detección del síndrome de Down basado en patrones faciales específicos de la enferme-

dad. Las características geométricas y de textura local se extraen basándose en marcas anatómicas detectadas automáticamente para describir la morfología y estructura facial. Las características faciales se extraen de los bebés con síndrome de Down. Se utilizan diferentes técnicas como ACP, análisis de componentes independientes (ACI) y el modelo de textura local con restricciones de forma (CLM siglas en ingles). CLM consiste en un modelo estadistico de forma y un modelo de textura local. Combinando estos dos modelos, tanto la morfología como la textura local de un rostro son descritas. Para la clasificación de los patrones obtenidos en la etapa de extracción de características, se utilizan máquinas de soporte vectorial.

(Ren, 2012) lleva a cabo la comparación del desempeño de las máquinas de soporte vectorial y redes neuronales artificiales (ANN) en la clasificación de cúmulos de microcalcificación en mamografías, lo cual es parte fundamental para el diagnóstico temprano de cáncer de mama. El autor reporta que las ANN presentan mejor desempeño que SVM cuando la cantidad de datos pertenecientes a las distintas clases es desproporcionada, es decir son datos no balanceados.

Además de imágenes, las técnicas de IA en combinación con otras técnicas de procesamiento de señales permiten analizar otros tipos de características como las señales eléctricas. (Martis y Acharya, 2013) llevaron a cabo el análisis de la señal del electrocardiograma (ECG) para clasificar diversos tipos de padecimientos del corazón. Estos son: bloqueo de la rama izquierda, bloqueo de la rama derecha, contracción prematura y contracción ventricular prematura. Para esta investigación se utilizó la base de datos de arritmia cardiaca del instituto tecnológico de Massachusetts (MIT-BIH). Para la etapa de extracción de características se aplicó la transformada wavelet discreta (DWT) con el fin de obtener información en el dominio del tiempo y frecuencia. Posteriormente, ACP y ACI se usaron para la reducción dimensional. Por último, se implementaron diversos clasificadores como redes neuronales, máquinas de soporte vectorial y una red neuronal probabilística (PNN). Al finalizar las pruebas, se encontro que la combinación de ACI y PNN tuvieron el desempeño más alto en la tarea de clasificación con un 97.97 % de efectividad.

Por su parte (Stewart et al., 2014), analizan la señal del electroencefalograma (EEG) con el fin de detectar la existencia de diversos estimulos visuales en humanos. ICA es aplicado para la extracción de características, posteriormente SVM multiclase es entrenado para realizar la tarea de clasificación. (Nguyen et al., 2015) proponen la clasificación de datos sanitarios (cáncer de mama y enfermedades del corazón) mediante un sistema basado en

lógica difusa y wavelets. La transformada wavelet se emplea para extraer características discriminatorias para conjuntos de datos de alta dimensión y la combinación de algoritmos genéticos (GA) con lógica difusa se implementa para propósitos de clasificación. GA se encarga de optimizar la estructura de la regla difusa, cada gen representa una regla difusa. El número de genes en un cromosoma, es igual al número de reglas difusas iniciales. Cada gen obtiene un valor de 0 o 1 donde 0 significa que la regla difusa correspondiente se omite, mientras que 1 significa que la regla difusa correspondiente se conserva. El cromosoma que se obtiene a través de una serie de generaciones en evolución representa el conjunto óptimo de reglas difusas. Los resultados muestran la dominancia del método wavelet-GSAM frente a otros como máquinas SVM y redes neuronales probabilísticas.

Metodologías antes mencionadas como SVM, PNN, K-medias. Se implementan con el objetivo de clasificar datos en 2 o más clases, sin embargo en la práctica existe otro tipo de clasificación. Este es el llamado problema de descripción de datos o clasificación de una clase. Aquí, el problema es hacer una descripción de un conjunto de entrenamiento de objetos y detectar nuevos objetos que se asemejan a este conjunto de entrenamiento. La descripción de datos se utiliza para tratar un problema de clasificación en donde solo se tiene suficiente información de una de las clases mientras que se tiene muy poca o nula información de la otra. Mahalanobis-Taguchi system (MTS) es una metodología que se ha aplicado de manera exitosa para lidiar con el problema de descripción de datos.

(M. Y. Abu y Ramlie, 2014) presentan un sistema basado en MTS para el reconocimiento de patrones con el fin de distinguir entre dos rangos distintos de remanufacturabilidad de una pieza mecánica y de esta manera evitar desperdicio de material. El estudio se aplicó en la industria automotriz de Malasia. La pieza es medida en tres distintos ejes y se generan patrones que corresponden a diferentes niveles de desgaste. MTS se encarga de reconocer estos patrones y determinar si la pieza es reusable. Sin embargo, este artículo no presenta en detalle la metodología, además de no efectuar comparación con alguna otra técnica.

(Iquebal et al., 2014) muestran una metodología para la clasificación basada en la distancia Mahalanobis (MD) y asociación mediante Rough Sets Theory (RST). La metodología es implementada en cinco casos de estudio, cada uno con dos posibles clases. Además de lidiar con el problema de clasificación, MTS es capaz de lidiar con el problema de selección de características o variables. En MTS estandar, la selección de características para mejorar la clasificación se realiza mediante la relación señal/ruido (S/N) y arreglos

ortogonales (OAs). Sin embargo, se ha debatido por las limitaciones en el manejo de un gran número de variables por parte de los OAs. La contribución de este artículo es precisamente sobre la selección de características relevantes basadas en la maximización del grado de dependencia entre subconjuntos de variables del sistema (SSVs) y categorías. Este artículo deja claro que el estudio de metodologías para mejorar la selección de características en MTS es un campo de investigación importante, al igual que la construcción del espacio de referencia.

(p. Gu et al., 2014) proponen un modelo de optimización del espacio de referencia mediante un hybrido de algoritmos genéticos (GA) y MTS, que tiene como objetivo minimizar la tasa de error de clasificación y maximizar la relación señal / ruido. También presenta una nueva función de ajuste para evitar la convergencia prematura del algoritmo genético. La novedad de este trabajo es el cálculo de MD para el espacio de referencia sobre muestras anormales, que se utiliza para construir una nueva función objetivo.

(Su y Hsiao, 2009) presentan un sistema para la multi clasificación y la selección de características de manera simultanea (MMTS). La escala de medición multi clase se construye estableciendo un espacio Mahalanobis individual para cada clase. Para aumentar la validez de la escala de medición, el proceso de Gram-Schmidt se lleva a cabo para ortogonalizar mutuamente las características y eliminar la multicolinealidad. Finalmente se propone una distancia Mahalanobis ponderada como una métrica para la clasificación. Se estudia un caso real de diabetes mellitus gestacional y los resultados indican la viabilidad del MMTS en aplicaciones de la vida real.

(Ghasemi et al., 2015) por su parte presenta una revisión detallada de la metodología MTS, así como una extensa revisión de estado del arte. Por lo tanto es útil para el investigador ya que amplia el panorama sobre las distintas posibles aplicaciones de la metodología. Otra técnica para la descripción de datos es support vector data description la cual se basa en SVM. (Wang y Tan, 2013) presenta la combinación de support vector data description (SVDD) y K-medias para la clasificación. Debido a que el método de K-medias no toma en cuenta la distribución no uniforme en el tamaño de los grupos. Intuitivamente, los grupos que contienen más datos son probablemente parte de las características con mayor influencia en comparación con los grupos más pequeños. Por lo tanto se implementa SVDD para medir la densidad de cada grupo formado por el método de K-medias. La propuesta es creativa ya que mejora el desempeño del algoritmo K-medias para la clasificación. Sin embargo, en este trabajo se depende de la expertiz humana al

momento de elegir el número de grupos que K-medias formará. Similar a MTS, SVDD puede extenderse para tareas de multi clasificación.

(Yang et al., 2011) desarrollan una nueva función de decisión para SVDD multi clase, usando el concepto de fuerza de atracción. Esto mejora el rendimiento del multi clasificador, ya que toma en cuenta la densidad de cada clase. Por otro lado, la elección de algunos parámetros para los métodos basados en SVM (como el kernel) es de vital importancia para el rendimiento de estos métodos. (Bounsiar y Madden, 2014) presentan una investigación acerca del desempeño de diversos kernel en tareas de clasificación de una clase (OCC). La investigación se justifica ya que otros autores asumen que los kernel usados en clasificadores binarios como SVM son apropiados para la descripción de datos. El estudio presenta la clasificación de datos referentes al cáncer de mama donde se identifica entre cáncer maligno y benigno. En este caso en particular el kernel función de base radial (RBF) tuvo el desempeño más alto, mientras otros kernel como el sigmoide o polinomial no tuvieron un buen desempeño.

Diversas técnicas de clasificación fueron descritas. Estas metodologías también se han implementado de forma exitosa para la detección de fallas en procesos multivariados complejos.

2.2.2. Detección de fallas y selección de características

(Chun-Chin et al., 2010) proponen integrar análisis de componentes independientes (ICA) y máquinas de soporte vectorial (SVM) con el fin de desarrollar un sistema inteligente para la detección de fallas en un proceso multivariado no gaussiano. ICA se implementa como una extensión de análisis de componentes principales (PCA). Sin embargo, ICA al no presentar restricciones de ortogonalidad, puede lidiar con procesos no gaussianos. El sistema ICA-SVM para la detección de fallas consiste de dos fases. Primero se desarrollan las bases de datos correspondientes a datos en operación normal y datos en falla para comenzar el proceso de entrenamiento del sistema. Finalmente se lleva a cabo la etapa de monitoreo y prueba. El desempeño de ICA-SVM es comparado con otros sistemas como SVM y PCA-SVM. El sistema propuesto en esta investigación muestra mejor desempeño en la tarea de detección de fallas que los otros sistemas antes descritos.

Como se mencionó antes, en la actualidad los procesos industriales son altamente com-

plejos. Estos a su vez se componen de dispositivos tecnológicos sofisticados de naturaleza mecánica, electrónica, neumática, entre otros. De tal manera que estos dispositivos pueden ser vistos como procesos, y por tal motivo las técnicas de inteligencia artificial también son implementadas con el fin de analizar y detectar fallas en dichos dispositivos.

(Yacout, Salamanca, y Mortada, 2013) patentan una metodología para la detección de fallas en dispositivos por mantenimiento basado en condiciones (CBM). Una vez que se recopilan datos representativos de condiciones dinámicas del dispositivo, se emplea aprendizaje automático (ML) para extraer patrones de los datos que permitan determinar si el dispositivo presenta alguna falla o no. Algunas de las técnicas analizadas por los autores son: PCA, SVM, ANN y lógica difusa, etc. La aplicación de CBM a equipos electromecánicos mediante el monitoreo de datos de indicadores, ha demostrado ser un método efectivo para proteger maquinaria costosa y asegurar su funcionamiento continuo. Una de las aplicaciones de esta herramienta es la predicción y detección de fallas en un transformador eléctrico mediante el control de varios gases. Esta herramienta es robusta y digna de patente porque su uso no se limita a un solo dispositivo.

(Gaied et al., 2012) presentan un sistema para la detección de fallas basado en la transformada wavelet discreta (DWT). Se simulan las fallas de bobinado del estator en corto y abierto, posteriormente se detecta la corriente del estator. Esta actúa como entrada para la DWT la cual transforma la señal del dominio del tiempo hacia el dominio tiempo-frecuencia, con el objetivo de extraer información crucial acerca de la falla. Lo interesante de este trabajo es el uso del llamado índice wavelet, que se obtiene mediante la energía de la corriente del estator y la descomposición wavelet (Daubechies wavelet). La detección de las distintas fallas depende del valor de este índice.

(Xiaogang y Xuemin, 2013) presentan un novedoso método de monitoreo no lineal basado en análisis de características kernel dinámico y SVDD. Este método fue probado en un proceso simulado de un reactor de tanque agitado continuo no isotérmico (CSTR). La experimentación demuestra que el método es capaz de capturar eficazmente las características no lineales en el proceso y llevar a cabo la detección de fallas eficiente.

Por su parte (Benkedjouh et al., 2012) presentan un metodo para el diagnóstico de fallas en rodamientos basado en PCA y SVDD. El propósito de este trabajo es transformar las señales monitoreadas de vibración en características que pueden usarse para rastrear el estado de salud de los rodamientos y así estimar su vida útil restante (RUL). Después de

que las señales son pre procesadas mediante filtros, se implementa PCA para reducir la dimensionalidad del sistema y posteriormente construir el modelo SVDD. Es decir obtener el valor del radio (R) y el centro de la hiperesfera, ya que mediante R es posible calcular el RUL. Este documento proporciona una visión general para la detección y predicción de fallas.

Por su parte (Ma et al., 2012) proponen un sistema para la detección y predicción de fallas basado en la combinación de PCA y SVM aplicado en turbinas de combustión por gas. Las señales de vibración de las turbinas de gas tienden a ser no estacionarias, por lo tanto, el análisis wavelet se utiliza para eliminar el ruido de dichas señales. Después se aplica PCA para reducir la dimensionalidad de los datos. Estos sirven para la construcción del modelo SVM para la predicción y diagnóstico de fallas.

(Keskes et al., 2013) presentan un sistema inteligente para la detección de barras rotas en motores eléctricos basado en la combinación de transformada wavelet estacionaria (SWPT) y Wavelet-SVM multiclase. La SWPT se utiliza para la extracción de características. Los autores informan que recientemente, la transformada estacionaria wavelet se ha convertido en una de las técnicas de procesamiento de señales más utilizadas para el diagnóstico de fallas en motores de inducción. Sin embargo, la SWPT sólo se utiliza para extraer características sensibles relacionadas con las fallas, pero no permite la detección automática de las mismas. Para realizar la clasificación (diagnóstico) de las fallas, el modelo SVM se construye usando como kernel una función wavelet. Ya que SVM fue originalmente diseñado para la clasificación binaria, los métodos conocidos como uno contra todos (OAA) y uno contra uno (OAO) se implementan con el fin de llevar a cabo una clasificación múltiple.

Por otro lado (Shen et al., 2013) proponen el diagnóstico de fallas para maquinária rotatoria. El esquema para el disgnóstico de fallas inteligente se basa en el cálculo de de parámetros estadísticos a partir de una transformación wavelet de las señales de vibración provenientes de acelerómetros. Después se calculan estadísticos como curtosis, varianza, etc. Estos son usados como parámetros de una regresión basada en vectore de soporte (SVR) para el diagnóstico de fallas. Además de la transformada wavelet bastante mencionada, existen otras transformaciones útiles para la extracción de características.

(Sajjad y Zaidi, 2013) implementan la transformada Choi Williams para analizar señales de corriente en máquinas eléctricas en el dominio tiempo-frecuencia. La máquina ba-

jo estudio esta en servicio y no se cuenta con información hístorica relacionada con las fallas de la misma. Para la etapa de diagnóstico de fallas, las señales se recogen continuamente durante un intervalo de tiempo específico. Posteriormente se realiza el proceso de transformación a tiempo-frecuencia, y se implementa PCA para deducción dimensional de características. Los nuevos vectores de características se proyectan en el vector propio más significativo de PCA y se calcula la media de los puntos proyectados, de esta manera se determina la clase.

(Ganesh et al., 2013) patentan un sistema para el pronóstico de fallas para la detección y diagnóstico de fallas en aislante eléctrico utilizando la transformada rápida de fourier (FFT). El sistema también incluye un módulo de adquisición de datos acoplado a dos transformadores de corriente, en el que el modelo de adquisición de datos recibe las señales de componentes de alta frecuencia y las analiza para identificar componentes defectuosos en el sistema de aislamiento eléctrico. Los autores realizaron el estudio de aislamiento eléctrico en componentes eléctricos pertenecientes a locomotoras y otros vehículos. Algunos de estos componentes son: generadores, rectificadores, convertidores, motores de tracción y ventiladores.

(Al-jonid et al., 2013) proponen un modelo para el diagnóstico de fallas en un control numérico por computadora (CNC). Este sistema involucra la combinación de redes neuronales difusas (NF) y redes bayesianas. Este, se encarga de clasificar las fallas en primer y segundo grado. Las de primer grado son fallas instantaneas mientras las de segundo grado son que evolucionan y aparecen como un fenómeno en desarrollo que comienza desde una etapa inicial hasta una etapa madura. La degradación de ocho herramientas fueron probadas en este estudio. Después de 500 casos simulados, la herramienta número 2 se pronostica como la más degradada con un porcentaje de 46.7 %. La desventaja del sistema es la alta sensibilidad al ruido. Esto puede explicarse porque en este trabajo, no se reporta la implementación de alguna técnica de pre procesamiento. (Yin et al., 2014) presentan un sistema para el diagnóstico de fallas en linea. Este, utiliza aprendizaje automático extremo (ELM) con estructura de salida incremental y SVDD (ISVDD). ISVDD permite desarrollar un sistema capaz de detectar y aprender nuevas fallas.

Por su parte (H. Jiang et al., 2014) proponen el diagnóstico de fallas en rodamientos mediante la aplicación de una descomposición en valores singulares (SVD) para la extracción de características en señales de vibración. Para realizar la clasificación de patrones que permite el diagnóstico de fallas se implementan modelos ocultos de Markov continuos

(CHMM). La novedad de este trabajo radica en la extracción de características utilizando SVD ya que permite obtener patrones con notable diferencia entre datos en funcionamiento normal y datos en falla. (Kutylowska, 2014) presenta la aplicación de redes neuronales artificiales para pronosticar el valor de salida de frecuencia de fallas en un sistema fluvial polaco. En este trabajo se utiliza una red perceptrón multicapa, donde se prueban varias funciones de activación. Incluyendo la tangente logística e hiperbólica. El aspecto interesante de este trabajo es el uso del algoritmo de aprendizaje denominado enfoque cuasi Newton, que según el autor es el más eficaz. El algoritmo cuasi Newton requiere un pequeño número de iteraciones para entrenar la red neuronal dada su rápida convergencia y criterio de búsqueda más inteligente.

Por su parte (Wenliao et al., 2014) utilizan optimización por enjambre de partículas (PSO) para encontrar el valor del parámetro C en SVDD y el parámetro s para el kernel RBF. Los cuales mejoran el diagnóstico de fallas en bombas hidráulicas. El uso de este tipos de algoritmmos bio inspirados, muestran ser una alternativa eficaz para la sintonización de hiperparámetros. (Reséndiz et al., 2013) aplican MTS a un problema dimensional en la industria automotriz. Una versión reciente del algoritmo de optimización por colonia de hormigas binario es implementada para la selección óptima de variables, sustituyendo a los arreglos ortogonales usados en MTS estándar. La metodología híbrida se aplica a un proceso de manufactura en la industria automotriz, donde el componente específico se supervisa con el fin de detectar problemas de calidad dimensional.

(Ren et al., 2011) proponen una sistema para el diagnóstico de fallas basado en multiclass MTS (MMTS), donde se adquiere señales de vibración del proceso. Posteriormente se implementa MMTS para realizar la clasificación de múltiples fallas y de manera simultanea llevar a cabo la selección de características. Esto, se realiza a través de la construcción de espacios de referencia para cada estado y por medio de la distancia Mahalanobis de los datos a clasificar se determina a que clase pertenece, es decir, que tipo de falla se detecta.

En (Aldape-Pérez et al., 2013) se muestra cómo el paradigma de la memoria asociativa y la computación paralela pueden utilizarse para realizar tareas de selección de características. Este enfoque utiliza memorias asociativas para obtener un valor de máscara que representa un subconjunto de características que identifica claramente información irrelevante o redundante para fines de clasificación. A lo largo de la fase experimental, seis conjuntos de datos se utilizaron como conjunto de pruebas para estimar el rendimiento de clasificación de cada uno de los algoritmos comparados. Los resultados experimentales

muestran que las memorias asociativas pueden ser implementadas en la infraestructura de computación paralela, reduciendo los costos computacionales necesarios para encontrar un subconjunto óptimo de características que maximice el desempeño de la clasificación. El trabajo se centra en mostrar resultados en términos de tiempo computacional y exactitud en la clasificación, sin embargo no presenta los resultados de referentes a las características seleccionadas, además de no presentar una aplicación en un proceso industrial.

Por su parte (Gamarra y Quintero, 2013) llevan a cabo la selección de características aplicando algoritmos genéticos para encontrar el subconjunto de variables para optimizar la tarea de clasificación por parte de redes neuronales artificiales. La etapa de experimentación se lleva a cabo en tres bases de datos: partes de auto, frutas y caras. El artículo es completo en experimentación sin embargo solo menciona el número de características consideradas en cada caso pero omite la descripción de dichas características. En (Chandrashekar y Sahin, 2014) se presenta una revisión de diversos métodos para la selección de características para mejorar la clasificación de las máquinas de soporte vectorial y de una red neuronal con función de base radial. El desempeño de los diferentes enfoques se valida en cinco diferentes conjuntos de datos del repositorio de aprendizaje de máquina de UCI. El artículo presenta una detallada descripción de las ventajas de realizar el proceso de selección de características así como de los diferentes tipos de algoritmos para la clasificación de datos, por lo tanto es de gran utilidad para este trabajo de investigación.

Por su parte (Saha y Rajasekaran, 2015) presentan tres algoritmos aleatorizados para tratar el problema de la selección de variables. Dichos métodos pueden ser pensados como un paseo al azar en el espacio de todos los subconjuntos posibles de las características. Estas aplicaciones incluyen, pero no se limitan a la predicción de propiedades de materiales, el procesamiento de datos biológicos y la integración de datos. El estudio demuestra buen desempeño de los métodos propuestos para la selección de características, sin embargo debido a la aleatoridad de los enfoques el tiempo computacional es un parámetro de interes para el investigador que no se reporta en el artículo.

(Liu y Zhang, 2016) proponen tres métodos de tipo filtro para la selección de características los cuales son comparados con métodos convencionales basados en distancias ecuclideas. Los métodos propuestos estan basados en distancias probabilísticas para medir la similitud entre muestras de datos. Las pruebas de la efectividad de la selección de caracterísiticas para el proceso de clasificación se realizan en diez conjuntos de datos del repositorio de aprendizaje de máquina de UCI así como en base de datos de imágenes fa-

ciales. Los resultados muestran mayor efectividad por parte de los tres métodos propuestos en comparación a diversos métodos convencionales para la selección de características.

Como se muestra en el estado del arte, técnicas como Wavelets, Support Vector Data Description y metaheurísticos, Se utilizan con éxito para desarrollar metodologías para la detección de fallas y selección de características aplicadas en diferentes dispositivos y procesos complejos multivariados. Esto, debido a su capacidad de analizar y procesar en línea grandes cantidades de datos. Por lo tanto, se concluye que las metodologías mencionadas son viables para su uso en este proyecto de investigación.

2.3. Preguntas de investigación

- ¿Que tipo de pre procesamiento se requiere para el análsis de los datos obtenidos de procesos multivariados?
- ¿Que técnicas de inteligencia artificial permiten llevar a cabo la detección de fallas en procesos complejos?
- ¿Como realizar la selección de variables relevantes?
- ¿Como evaluar el desempeño del sistema para la detección de fallas?

2.4. Hipótesis

Mediante la combinación de Support Vector Data Description y una metaheurística, es posible llevar a cabo la selección de variables relevantes para mejorar la detección de fallas reduciendo el error en la clasificación.

2.5. Objetivos

2.5.1. Objetivo General

Desarrollar una metodología para la detección de fallas y la selección de variables relevantes e implementarla en un proceso multivariado complejo, obteniendo un error en la clasificación menor que otros enfoques basados en inteligencia artificial.

2.5.2. Objetivos específicos

- Realizar un pre procesamiento de datos.
- Combinar Support Vector Data Description y una metaheurística para la detección de fallas y selección de variables.
- Implementar una técnica para la validación de la metodología.
- Desarrollar una interfaz para la visualización del análisis.

2.6. Justificación

En la actualidad la industria cuenta con procesos altamente complejos, estos se componen de avanzados y costosos dispositivos. Por lo tanto, cuando una falla se presenta en ellos, genera todo tipo de escenarios no deseados como tiempos muertos, costos de reparación y en el peor de los casos la necesidad de comprar nuevos equipos. Por tal motivo, es de interes investigar técnicas que permitan desarrollar una metodología para la detección de fallas, capaz de identificar las variables que contribuyen a estas anomalias, y así llevar a cabo una mejora del proceso para evitar los inconvenientes antes mencionados.

2.7. Alcances y limitaciones

Alcances:

- En este trabajo de investigación, se presenta una metodología para la detección de fallas y selección de variables en procesos multivariados capaz de lidiar con datos con clases no balanceadas. Dicha metodología trabaja con conjuntos de datos completos y de naturaleza continua.

Limitaciones:

- Falta de datos adecuados para el entrenamiento de la metodología propuesta, es decir, falta de muestras del proceso en operación normal y en falla.

3

Marco Téorico

El primer paso para llevar a cabo un análisis de cualquier proceso con el fin de detectar fallas en el mismo, es precisamente contar con información para analizar. Por lo tanto, se debe contar con un sistema para la adquisición y muestreo de las señales relevantes del sistema y/o proceso. Una vez que se cuenta con la información existen diversas metodologías y técnicas para procesarlas, como las descritas en el estado del arte. A continuación se presentan conceptos y teoría de algunas técnicas relevantes para este proceso de investigación, las cuales incluyen procesamiento de señales, reconocimiento de patrones y metaheurísticas.

3.1. Transformada Wavelet

Una de las técnicas más usadas para el procesamiento de señales es el análisis Wavelet debido a su flexibilidad. A continuación se estudiará la idea detrás del análisis Wavelet, este fue tomado de (Hernández, 1996), (Radunoviç, 2009), (E. O. Reséndiz-Flores, 2001), (Walker, 2008), (Daubechies, 1992), (Torrence y Compo, 1998), (Nason, 2008) y (Frazier, 1999).

En 1946 Dennis Gabor definió funciones en tiempo-frecuencia, llamadas Wavelets de Gabor. Su idea fue dividir una onda, representada en notación matemática como $\cos(wt + \varphi)$, en segmentos y retener solo algunos de ellos.

Una Wavelet es una función de onda con un soporte compacto. Se le llama onda debido a

su naturaleza oscilatoria, y el diminutivo se utiliza por el dominio finito donde es diferente de cero (el soporte compacto). La amplitud y la traslación de la Wavelet básica $\psi(x)$ (La Wavelet "madre") define la base Wavelet,

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right), \qquad a > 0.$$

Al elegir los valores adecuados para los parámetros de amplitud a y traslación b, las partes pequeñas de una función con forma compleja pueden ser representadas con una resolución más alta, mientras que las partes sencillas pueden ser representadas con una menor resolución.

La transformada Wavelet es utilizada para descomponer datos, funciones u operadores en varios componentes de frecuencia, luego cada componente es analizado en el mejor ajuste de resolución para su escala. La transformada Wavelet continua (T.W.C) es definida por el producto interno de la función y la base Wavelet, es decir,

$$T.W.C._f(a,b) = (f, \psi_{a,b}) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(x)\overline{\psi\left(\frac{x-b}{a}\right)}dx. \tag{3.1}$$

$T.W.C._f(a,b)$(ecuación (3.1)) es una función de amplitud a y posición b, y muestra que tan estrecha es la Wavelet y la función de correlación en el intervalo determinado por el soporte de la Wavelet. La transformada Wavelet mide la similitud entre el contenido de frecuencias de la función y la Wavelet base $\psi_{a,b}(x)$ en el dominio tiempo-frecuencia.

Por otro lado podemos definir la transformada Wavelet continua inversa (T.W.C.I.) como:

$$f(x) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} T.W.C._f(a,b)\psi_{a,b}(x)\frac{da\ db}{a^2},$$

esta existe si la condición de admisibilidad se cumple,

$$C_\psi = \int_{-\infty}^{\infty} \frac{\left|\widehat{\psi}(w)\right|^2}{w}dw < \infty. \tag{3.2}$$

La condición (3.2) implica que,

$$\widehat{\psi}(0) = \int_{-\infty}^{\infty} \psi(x)dx = 0, \tag{3.3}$$

lo cual brinda el carácter oscilatorio de la función $\psi(x)$. La ecuación (3.2) también implica que esta función necesita ser diferente de cero fuera del intervalo finito, pero esta debe converger a cero lo suficientemente rápido cuando $|x| \to \infty$. Por lo tanto, se llega a la generalización del concepto Wavelet. La Wavelet básica $\psi(x)$ es una función arbitraria que tiene un valor medio igual a cero y que disminuye lo suficientemente rápido en el infinito, estas son propiedades de las Wavelet base $\psi_{a,b}(x)$. Esta es una de las diferencias entre la transformada Wavelet y la de Fourier u otras transformaciones, mientras que en las otras transformaciones las bases son determinadas, las bases Wavelet no son dadas explícitamente. La teoría sólo da propiedades generales de las Wavelets y las transformaciones, y define las ideas generales con las que cualquiera puede construir una Wavelet de acuerdo a sus necesidades. Cuando se construye o se elige la Wavelet básica, la base es definida con dicha función.

Sin embargo, la transformada Wavelet continua (T.W.C.) no es de mayor uso práctico, debido a que la correlación entre la función y la Wavelet es calculada mientras esta es continuamente trasladada y ampliada, es decir, los parámetros a y b son variables continuas. Por lo tanto la mayoría de los coeficientes calculados son redundantes y hay un número infinito de ellos, por esa razón se lleva a cabo una discretización. El plano amplitud-tiempo se cubre con una malla y en los nodos de la malla la T.W.C. es calculada para valores discretos de los parámetros a y b.

Las Wavelets discretas son usualmente funciones continuas parte por parte y no pueden ser ampliadas ni trasladadas continuamente, solo en pasos discretos como se observa a continuación,

$$\psi_{j,k}(x) = \frac{1}{\sqrt{a_0^j}}\psi\left(\frac{x - kb_0a_0^j}{a_0^j}\right) \tag{3.4}$$

donde j y k son enteros, mientras que $a_0 = 2$ así la división sobre el eje frecuencia es diádica. El factor traslación es usualmente elegido como $b_0 = 1$, de manera que la división

sobre el eje tiempo de la amplitud seleccionada es uniforme,

$$\psi_{jk}(x) = 2^{-j/2}\psi(2^{-j}x - k), \quad \text{y} \quad \psi_{jk}(x) \neq 0 \quad \text{para} \quad x \in [2^j k, 2^j(k+1)]. \tag{3.5}$$

Cuando el parámetro amplitud a es discretizado en escala logarítmica, el parámetro b es discretizado dependiendo del parámetro amplitud, es decir un número diferente de puntos se utilizan en varias amplitudes.

Se lleva a cabo la discretización a través de la malla, primero el parámetro de amplitud es discretizado eligiendo puntos enteros en la escala logarítmica. La base del logaritmo depende del usuario, sin embargo usualmente se utiliza 2 como base, por lo tanto las amplitudes para las cuales la transformación se calculará son: 2, 4, 8, 16, 32,... Después de esto, una discretización uniforme del eje tiempo se lleva a cabo en función de la discretización aplicada al eje amplitud. Si la base del logaritmo es 2, es decir, los valores discretos del parámetro amplitud cambian para un factor de 2, el número de puntos en el eje tiempo es reducido por un factor de 2 sobre la amplitud siguiente. Por ejemplo, si tenemos 32 puntos equidistantes en el eje tiempo y la amplitud $a = 2$, entonces en la siguiente amplitud $a = 4$ el número de puntos en el eje tiempo es reducido por un factor de 2, es decir, 16 puntos equidistantes serán usados y así sucesivamente.

Los límites de la discretización y el número de puntos son importantes si se espera que la función sea reconstruida. El número de puntos de Nyquist es el número de puntos mínimos que garantizan la posibilidad de reconstruir la función continua. La regla de Nyquist establece que si el plano tiempo-amplitud necesita tener N_1 puntos en la amplitud a_1, entonces es suficiente tener N_2 puntos en la amplitud a_2,

$$a_1 N_1 = a_2 N_2, \qquad \text{o} \qquad \frac{N_1}{w_1} = \frac{N_2}{w_2}, \tag{3.6}$$

donde $a_1 < a_2$, esto significa que $N_2 < N_1$. La frecuencia w es igual al valor inverso de la amplitud, $w = 1/a$. Para frecuencias bajas, el número de puntos puede ser reducido, lo que a su vez reduce significativamente el número de cálculos.

3.2. Multirresolución

La transformada Wavelet, tal como se definió anteriormente, se basa en la representación de componentes de una función en diferentes escalas, es decir, con diferentes resoluciones, para comprender mejor estos conceptos se estudiará el análisis multirresolución, este fue desarrollado en base a (Radunoviç, 2009), (Hubbard, 1996), (Kobayashi, 1998) y (Misiti et al., 2007).

La idea detrás de la llamada multirresolución puede ser explicada a través de un ejemplo. Los mapas geográficos contienen información diferente a diferentes escalas, por tal motivo se ha realizado una normalización de los datos cartográficos mediante la división en categorías como ciudad, región, país, continente. Estas categorías no son totalmente independientes, ya que datos importantes que existen en una determinada escala se repiten en la siguiente escala más grande. Por lo tanto, eso es suficiente para determinar las conexiones entre información dada en dos escalas colindantes por ejemplo que ciudad pertenece a cual región, que región pertenece a cual pais (Radunoviç, 2009).

3.2.1. Análisis multirresolución

Definición 1. El análisis multiresolución es una descomposición del espacio de Hilbert $\mathcal{L}_2(\mathbb{R})$ en una serie de subespacios cerrados $\{\mathcal{V}_j\}_{j\in\mathbb{Z}}$ tales que

$$\cdots \subset \mathcal{V}_2 \subset \mathcal{V}_1 \subset \mathcal{V}_0 \subset \mathcal{V}_{-1} \subset \mathcal{V}_{-2} \subset \ldots \tag{3.7}$$

$$\cap_{j\in\mathbb{Z}}\mathcal{V}_j = \{0\}\,, \qquad \overline{\cup_{j\in\mathbb{Z}}\mathcal{V}_j} = \mathcal{L}_2(\mathbb{R}) \tag{3.8}$$

$$\forall f \in \mathcal{L}_2(\mathbb{R}) \quad \text{y} \quad \forall j \in \mathbb{Z}, \qquad f(x) \in \mathcal{V}_j \Leftrightarrow f(2x) \in \mathcal{V}_{j-1} \tag{3.9}$$

$$\forall f \in \mathcal{L}_2(\mathbb{R}) \quad \text{y} \quad \forall k \in \mathbb{Z}, \qquad f(x) \in \mathcal{V}_0 \Leftrightarrow f(x-k) \in \mathcal{V}_0 \tag{3.10}$$

$$\exists\ \varphi \in \mathcal{V}_0 \text{ tal que } \{\varphi(x-k)\}_{k\in\mathbb{Z}} \quad \text{es una base Riesz del subespacio } \mathcal{V}_0. \tag{3.11}$$

En la propiedad (3.11) la base puede ser elegida de tal manera que sea una base ortonormal del espacio $\mathcal{V}_0$.

Por lo tanto, cada aproximación espacial $\mathcal{V}_j$, donde $j \in \mathbb{Z}$, es una versión escalada del espacio básico $\mathcal{V}_0$, esto se obtiene por medio de la ampliación diádica, es decir, por un factor de 2^j, dicho espacio básico se contrae o expande dependiendo del signo de j. El espacio $\mathcal{V}_j$ contiene detalles dos veces más finos que los contenidos por su predecesor en la escala $\mathcal{V}_{j+1}$, así cuando $j \to -\infty$ la aproximación se vuelve una representación ya que $\mathcal{V}_j$, tiende a $\mathcal{L}_2(\mathbb{R})$.

La base del espacio $\mathcal{V}_0$ es formado por la función $\varphi(x)$ y su traslación $\varphi(x-k)$, donde $k \in \mathbb{Z}$ (ec. 3.11), por lo tanto si denotamos la función $\varphi(x)$, ampliada j veces y trasladada por k como

$$\varphi_{j,k}(x) = 2^{-\frac{j}{2}}\varphi\left(2^{-j}x-k\right), \qquad j,k \in \mathbb{Z},$$

las bases de Riesz del espacio $\mathcal{V}_j$ son el conjunto de funciones $\{\varphi_{j,k}(x)\}$, $k \in \mathbb{Z}$.

Una vez descrito lo anterior podemos definir la dilatación como la amplitud de la función $f(x)$ por el factor de amplitud 2^j, escrito como $f(x/2^j)$. Por lo tanto el resultado será la expansión ($j > 0$) o contracción ($j < 0$) de la función. En la figura (3.1) podemos observar el efecto de la dilatación en las funciones $sen(x)$ y la Wavelet $\psi(x)$.

Por otro lado la traslación es definida como el desplazamiento de la función $f(x)$ por k, denotada como $f(x-k)$, resultando un retraso ($k < 0$) o un adelanto ($k > 0$) de la función (figura 3.2). La función $\varphi_{j,k}(x) = 2^{-j/2}\varphi(2^{-j}(x-k2^j))$ es obtenida por la traslación diádica, es decir, trasladada por un factor $k2^j$.

Pasando del espacio $\mathcal{V}_{j-1}$ al espacio $\mathcal{V}_j$ ciertos detalles se pierden debido a la reducción en la resolución. Ya que $\mathcal{V}_{j\subset}\mathcal{V}_{j-1}$, los detalles perdidos se conservan en los complementos ortogonales del subespacio $\mathcal{V}_j$ en relación con el espacio $\mathcal{V}_{j-1}$. Este complemento ortogonal es llamado el espacio Wavelet, y se denota como $\mathcal{W}_j$ en la amplitud j.

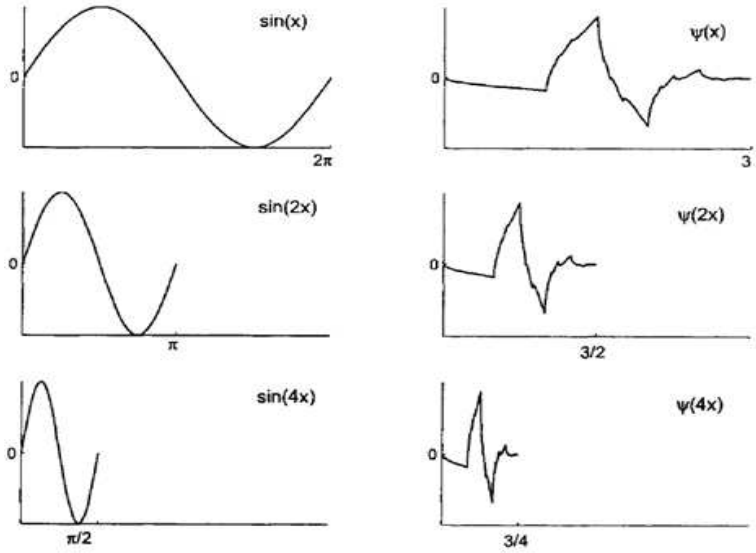

Figura 3.1: Dilatación diádica de la función seno y la wavelet Db2. Adaptada de (Radunoviç, 2009)

Por lo tanto

$$\mathcal{V}_{j-1} = \mathcal{V}_j \oplus \mathcal{W}_j, \tag{3.12}$$

donde $\oplus$ denota la suma directa. La ecuación (3.12) establece propiedades de multirresolución importantes:

1. Los espacios Wavelet $\mathcal{W}_j$ son la diferencia de los espacios de aproximación $\mathcal{V}_j$.

2. Los espacios de aproximación $\mathcal{V}_j$ son la suma de los espacios Wavelet $\mathcal{W}_j$.

Basados en la ecuación (3.12) para una J arbitraria tenemos que

$$\mathcal{V}_J \oplus \mathcal{W}_J = \mathcal{V}_{J-1}, \qquad \mathcal{V}_{J-1} \oplus \mathcal{W}_{J-1} = \mathcal{V}_{J-2}. \tag{3.13}$$

Sustituyendo la segunda relación en la primera, representamos el espacio $\mathcal{V}_{J-2}$ como una

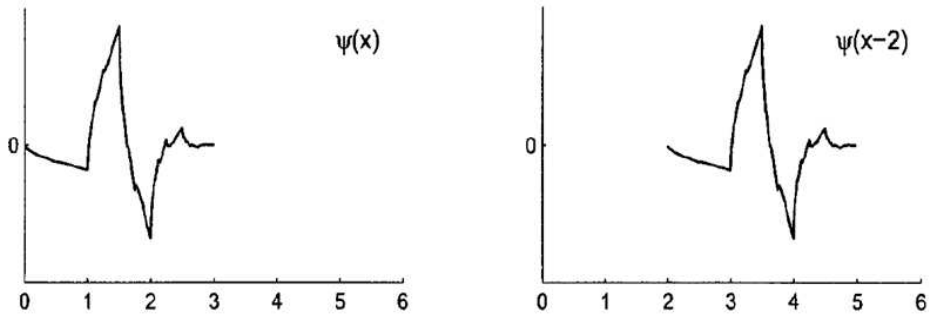

Figura 3.2: Traslación de la wavelet Db2. Adaptada de (Radunoviç, 2009)

suma de tres subespacios ortogonales

$$\mathcal{V}_J \oplus \mathcal{W}_J \oplus \mathcal{W}_{J-1} = \mathcal{V}_{J-2}. \tag{3.14}$$

Una mayor descomposición de los espacios de aproximación de acuerdo al mismo algoritmo, nos lleva a el espacio $\mathcal{V}_{j-1}$,

$$\mathcal{V}_J \oplus \mathcal{W}_J \oplus \mathcal{W}_{J-1} \oplus \cdots \oplus \mathcal{W}_j = \mathcal{V}_{j-1}, \quad J > j. \tag{3.15}$$

Todos los espacios $\mathcal{W}_k$, $k \geq j$, son ortogonales al espacio $\mathcal{W}_{j-1}$, debido a que este es ortogonal al espacio $\mathcal{V}_{j-1}$ que los contiene. Por lo tanto, como una consecuencia de la relación (3.15) se llega a la ortogonalidad de los espacios $\mathcal{W}_j$,

$$\mathcal{W}_k \perp \mathcal{W}_j, \quad k, j \in \mathbb{Z} \quad k \neq j. \tag{3.16}$$

La ecuación (3.8) nos lleva a la descomposición del espacio $\mathcal{L}_2(\mathbb{R})$ de las siguientes maneras. Cuando $j \to -\infty$ tenemos la descomposición,

$$\mathcal{L}_2(\mathbb{R}) = \mathcal{V}_J \oplus \sum_{j=-\infty}^{J} \mathcal{W}_j, \tag{3.17}$$

y cuando $J \to \infty$ tenemos

$$\mathcal{L}_2(\mathbb{R}) = \sum_{j=-\infty}^{\infty} \mathcal{W}_j. \tag{3.18}$$

Similar a los espacios de aproximación $\mathcal{V}_j$, los espacios Wavelet $\mathcal{W}_j$ son generados por amplitudes y traslaciones diádicas de otra función $\psi(x) \in \mathcal{L}_2(\mathbb{R})$, llamada la Wavelet básica ("madre")

$$\mathcal{W}_j = \overline{\{\psi_{j,k} \in \mathcal{L}_2 \mid k \in \mathbb{Z}\}}, \qquad \psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x - k), \quad j \in \mathbb{Z}.$$

Los subespacios $\mathcal{V}_0$ y $\mathcal{W}_0$ se encuentran contenidos en el espacio $\mathcal{V}_{-1}$ (ecuación (3.12) para j=0), así las funciones $\varphi(x) \in \mathcal{V}_0$ y $\psi(x) \in \mathcal{W}_0$ pertenecen al espacio $\mathcal{V}_{-1}$. Por lo tanto pueden ser representadas por funciones base $\varphi_{-1,k}(x) = \sqrt{2}\varphi(2x - k)$ del espacio $\mathcal{V}_{-1}$. Dichas representaciones son llamadas ecuación de dilatación y ecuación wavelet, dadas por 3.19 y 3.20 respectivamente.

$$\varphi(x) = \sum_k c(k)\sqrt{2}\varphi(2x - k) \tag{3.19}$$

$$\psi(x) = \sum_k d(k)\sqrt{2}\varphi(2x - k) \tag{3.20}$$

La ecuación de dilatación (3.19) tiene un número infinito de soluciones siempre y cuando $\varphi(x)$ sea una solución de la ecuación, entonces la función $c\varphi(x)$, donde c es una constante arbitraria, es también una solución. Con el fin de definir una sola solución, se requiere que la solución sea normada, es decir, que su valor medio sea igual a uno.

$$\int_{-\infty}^{\infty} \varphi(x)dx = 1. \tag{3.21}$$

integrando la ecuación (3.19), tomando en cuenta la ecuación (3.21),

$$1 = \int \varphi(x)dx = \sum_k c(k)\sqrt{2}\int \varphi(2x - k)dx = \frac{1}{\sqrt{2}}\sum_k c(k),$$

se obtiene la condición de los coeficientes $c(k)$ de la ecuación de dilatación

$$\sum_k c(k) = \sqrt{2}. \tag{3.22}$$

La función de amplitud $\varphi(x)$ es determinada por los coeficientes $c(k)$ de la ecuación de dilatación.

Determinando los coeficientes $c(k)$, la ecuación de dilatación (3.19) se resuelve por la función de ampliación $\varphi(x)$, que determina las bases $\{\varphi_{j,k}(x)\}$ de los espacios de aproximación $\mathcal{V}_j$. Entonces eligiendo los coeficientes $d(k)$ a través de la expresión (3.20), la Wavelet básica $\psi(x)$ es definida, por lo cual la base $\{\psi_{j,k}(x)\}$ del espacio wavelet $\mathcal{W}_j$ es determinada. A través de las dilataciones y las traslaciones estos espacios son ampliados, terminados y conectados:

f(x) $\in \mathcal{V}_j \quad \rightarrow \quad f(x), f(2x), f(x-k), f(2x-k) \in \mathcal{V}_{j-1}$.(3.23)Si $f(x) \in \mathcal{V}_0$, entonces $f(2x) \in \mathcal{V}_{-1}$, y $f(2x-k) \in \mathcal{V}_{-1}$. De lo anterior deducimos que $f(2^j x) \in \mathcal{V}_{-j}$ y, $f(2^j x-k) \in \mathcal{V}_{-j}$. Por lo tanto la dilatación y la traslación son construidas en los espacios $\mathcal{V}_j$. La ecuación (3.11) se relaciona con la base $\varphi(x)$ de tal manera que $\{\varphi(x-k)\}_k$ es una base ortonormal del espacio $\mathcal{V}_0$. Entonces $\left\{\sqrt{2}\varphi(2x-k)\right\}_k$ es una base ortonormal del espacio $\mathcal{V}_{-1}$, y en general, $\left\{2^{-j/2}\varphi(2^{-j}x-k)\right\}_k$ es una base ortonormal del espacio $\mathcal{V}_j$. El concepto unidimensional de multiresolución puede ser generalizado en espacios multidimensionales.

3.3. Limpieza de señales usando Wavelet

La eficiencia de las wavelets en el análisis de señales se debe principalmente al hecho que estas representan una manera uniforme y fácil de extraer componentes de frecuencia variable en el tiempo. Esta información puede usarse para llevar a cabo tareas de limpieza y compresión de manera eficiente. Esto de una forma totalmente diferente a los métodos convencionales de filtrado o compresión. El concepto principal de estos métodos es que al transformar la señal a procesar del dominio temporal al dominio tiempo-frecuencia mediante Wavelets. Las señales espurias (como el ruido) que corrompen la información útil, presentan coeficientes pequeños, de tal manera que al eliminarlos y posteriormente

llevar a cabo la transformación wavelet inversa, es posible limpiar la señal, al mismo tiempo que la distorsión en la señal es mínima (Giaouris et al., 2008). Sin embargo en aplicaciones de la vida real los procesos generalmente generan múltiples señales, es decir son multivariados por naturaleza. Por tal motivo deben ser analizadas con herramientas apropiadas, como el enfoque que presenta la combinación de wavelets y componentes principales. Antes de presentar dicho enfoque se desarrolla el concepto teórico del análisis de componentes principales (PCA).

3.3.1. Análisis de componentes principales (PCA)

El análisis de componentes principales es un método estadístico multivariado que puede utilizarse para la reducción de la dimension de los datos. Por reducción de datos se refiere que estos métodos son muy poderosos para extraer un pequeño número de factores ocultos de una cantidad masiva de datos multivariados. Estos factores explican la mayor parte de la variación en los datos. Las estructuras de estos factores ocultos pueden ayudar mucho en la búsqueda de las causas de la variación (K. Yang, 2004).

Los componentes principales son una combinación lineal particular de las p variables aleatorias $x_1, x_2, ..., x_p$. Geométricamente estas combinaciones lineales representan la selección de un nuevo sistema de coordenadas obtenido por rotación del sistema original, con $x_1, x_2, ..., x_p$ como nuevos ejes. Estos representan las direcciones con la máxima variabilidad y proporcionan una descripción más simple y más parsimoniosa de la estructura de la covarianza (Richard y Dean, 2007).

Considere una matriz de datos $\mathbf{X}$ de tamaño $m \times n$, para m variables del proceso con n mediciones de cada variable. Asumimos que los datos para cada variable han sido escalados con media 0 y varianza 1. La matriz de covarianza de $\mathbf{X}$ está definida como

$$\sum = \frac{\mathbf{X}^T \cdot \mathbf{X}}{n-1} \tag{3.24}$$

Sea $\lambda_i (i = 1, 2, , m)$ los valores propios de la matriz de covarianza, los cuales son organizados en orden descendente para determinar los componentes principales (CPs), y los componentes principales $\mathbf{p}_i$ (*loadings*) los cuales son los correspondientes vectores propios. Entonces, los primeros K CPs son seleccionados para construir el modelo ACP, por lo

que la matriz de datos $\mathbf{X}$ puede ser expandida usando los componentes principales $\mathbf{p}_i$, los vectores $\mathbf{t}_i$ (*scores*), y la matriz residual $\mathbf{E}$

$$\mathbf{X} = \sum_{i=1}^{K} \mathbf{t}_i \mathbf{p}_i + \mathbf{E} \tag{3.25}$$

Los $\mathbf{t}_i$ contienen información sobre como las muestras estan relacionadas entre si. Mientras los $\mathbf{p}_i$ contienen información sobre la relación existente entre las variables. El número de componentes principales debe ser igual o menor que las variables de $\mathbf{X}$.

Para una nueva observación $\mathbf{x}_{new}$, la predicción del modelo ACP está dada por

$$\hat{\mathbf{x}}_{new} = \mathbf{t}_{new}\mathbf{P}^T = \mathbf{x}_{new}\mathbf{P}\mathbf{P}^T \tag{3.26}$$

donde $\mathbf{t}_{new}$, es el vector de *scores*. El residual esta dado por

$$\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new} \tag{3.27}$$

3.3.2. Limpieza de señales multuvariada usando wavelets y PCA

Considere el siguiente modelo p dimensional:

$$X(t) = f(t) + \varepsilon(t), \qquad t = 1, ..., n. \tag{3.28}$$

donde $X(t)$, $f(t)$ y $\varepsilon(t)$ son vectores de tamaño $1 \times p$ y $\varepsilon(t)$ es el ruido blanco Gaussiano con matriz de covarianza desconocida $E\left(\varepsilon(t)^T\varepsilon(t)\right) = \Sigma_\varepsilon$. Cada componente de $X(t)$ es de la forma previa (Ec. 3.28), para $1 \leq i \leq p$:

$$X^i(t) = f^i(t) + \varepsilon^i(t), \qquad t = 1, ..., n. \tag{3.29}$$

Donde f^i pertenece a un espacio funcional. La matriz de covarianza Σ_ε, que se supone positiva definida, captura la relación estocástica entre los componentes de $X(t)$ y modela la correlación espacial.

La idea de usar PCA, no es para descubrir nuevas variables que podrían ser de interés,

sino para eliminar los componentes principales no significativos y de esta manera obtener un efecto de limpieza adicional. A continuación se presentan los pasos para llevar a cabo la limpieza de señales usando wavelets y PCA (Aminghafari et al., 2006).

1. Realice la transformación Wavelet al nivel J sobre cada columna de X.
2. Defina el estimador de la matriz de covarianza de ruido $\hat{\Sigma}_\varepsilon = MCD(D_1)$ y calcule V como $\hat{\Sigma}_\varepsilon = V\Lambda V^T$ donde $\Lambda = diag(\lambda_i,\ 1 \leq i \leq p)$.
3. Implemente PCA sobre la matriz A_J y seleccione el número apropiado p_{J+1} de componentes principales útiles.
4. Reconstruya la matriz libre de ruido $\check{X}$, A partir de las matrices simplificadas de detalle y aproximación, mediante el cambio de base V^T e invirtiendo la transformada Wavelet.

El procedimiento descrito generaliza tanto la limpieza de ruido univariado como la propiedad de reducción de dimensionalidad de PCA, lo cual conduce a un efecto de limpieza adicional.

3.4. Correlación

En las diferentes áreas del conocimiento existen problemas que requieren el análisis de más de una variable, como por ejemplo; un sociólogo puede estar interesado en saber qué clase de relación existe entre la tasa de delincuencia juvenil que hay en la comunidad y el grado de hacinamiento de los hogares que allí se encuentran; un agrónomo desea conocer si existe relación entre la cantidad de lluvia caída y el rendimiento de ciertos productos agrícolas, es decir, si es afectado desfavorablemente tanto por la excesiva lluvia (humedad), como por la excesiva sequía del suelo. Estas relaciones y muchas otras se pueden investigar por medio del análisis de correlación.

3.4.1. Coeficiente de correlación lineal de Pearson

De los diversos coeficientes de correlación que existen, el más popular y utilizado es el coeficiente de correlación de Pearson. El coeficiente de correlación de Pearson, pensado

para variables cuantitativas, es un índice que mide el grado de covariación entre distintas variables relacionadas linealmente. Esto significa que puede haber variables fuertemente relacionadas, pero no de forma lineal. Por ejemplo, si relacionamos población y tiempo la relación será de forma exponencial. En cuyo caso no es conveniente aplicar la correlación de Pearson. Si se tienen dos variables X e Y, el coeficiente de correlación de Pearson entre estas dos variables se define como r_{XY} donde sus valores absolutos oscilan entre 0 y 1, entonces: $0 \leq r_{XY} \leq 1$. Se ha especificado los términos valores absolutos ya que en realidad si se contempla el signo el coeficiente de correlación de Pearson oscila entre 1 y +1. No obstante ha de indicarse que la magnitud de la relación viene especificada por el valor numérico del coeficiente, reflejando el signo la dirección de tal valor. En este sentido, tan fuerte es una relación de +1 como de -1. En el primer caso la relación es perfecta positiva y en el segundo perfecta negativa. La correlación entre las variables X e Y es perfecta positiva cuando exactamente en la medida que aumenta una de ellas aumenta la otra. Mientras que es perfecta negativa cuando exactamente en la medida que aumenta una variable disminuye la otra (Rosas-Snell y Zúñiga-Contreras, 2004). En forma práctica, el coeficiente de correlación de Pearson es:

$$r_{XY} = \frac{N\sum_{i=1}^{N}(XY) - \left(\sum_{i=1}^{N} X\right)\left(\sum_{i=1}^{N} Y\right)}{\sqrt{\left[N\sum_{i=1}^{N} X^2 - \left(\sum_{i=1}^{N} X\right)^2\right]\left[N\sum_{i=1}^{N} Y^2 - \left(\sum_{i=1}^{N} Y\right)^2\right]}} \tag{3.30}$$

Donde N es el número de datos.

Por otro lado el coeficiente de Pearson estudia la correlación paramétrica, es decir, se aplica para casos en donde la distribución de los datos sigue una curva Gausiana o normal. Sin embargo, los datos conseguidos en las ciencias naturales y ciencias sociales, raramente se ajustan a la curva normal, y por esto cuando se trata de estudiar el grado de correlación en estas ciencias se debe utilizar técnicas de correlación no paramétrica, las cuales son libre de distribución, es decir, no existe la necesidad de que los datos tengan una distribución normal.

3.4.2. Correlación no paramétrica: coeficiente de Kendall

El coeficiente τ de kendall se basa más en los intervalos jerarquizados de las observaciones que en los propios datos, esto hace que la distribución de τ sea independiente de la que presentan las variables X y Y, siempre y cuando los datos representados por estas 2 variables sean independientes continuos. Este coeficiente cuenta con la ventaja de que tiende más rápido a la distribución normal que otros coeficientes, especialmente, en el caso de la certeza de Ho. EL coeficiente de Kendall se calcula de la siguiente manera:

$$\tau = \frac{(S_{\mathrm{a}} - S_{\mathrm{b}})}{[n(n-1)/2]} \tag{3.31}$$

donde,
n = número de casos.
S_{a} = sumatoria de rangos más altos
S_{b} = sumatoria de rangos más bajos.
Ver ejemplos numéricos en (Badii et al., 2014).

3.4.3. Significancia del coeficiente de correlación

Al calcular el coeficiente de correlación es de interés el determinar si dicho valor muestra que las variables X y Y están relacionadas en realidad o tan solo presentan dicha relación como consecuencia del azar, es decir, la significancia de dicho coeficiente de correlación. Se dice que un coeficiente de correlación es significativo si se puede afirmar, con una cierta probabilidad, que dicha correlación es diferente de cero. Por lo tanto se tienen dos hipótesis posibles:

$\mathrm{H}_0 :\Rightarrow$ El coeficiente de correlación obtenido procede de una población cuya correlación es cero ($\rho = 0$).

$\mathrm{H}_1 :\Rightarrow$ El coeficiente de correlación obtenido procede de una población cuya correlación es distinta cero ($\rho \neq 0$).

Desde el supuesto de la hipótesis nula se demuestra que la distribución muestral de co-

rrelaciones procedentes de una población caracterizada por una correlación igual a cero ($\rho = 0$) sigue una ley de Student con $N - 2$ grados de libertad, con media el valor poblacional (μ) y desviación tipo:

$$S_r = \sqrt{\frac{1 - r_{XY}^2}{N - 2}} \tag{3.32}$$

Dado un coeficiente de correlación r_{XY} obtenido en una determinada muestra, se trata de comprobar si dicho coeficiente se encuentra dentro de la distribución muestral especificada por la hipótesis nula. Entonces se calcula el número de desviaciones y se encuentra el coeficiente obtenido del centro de la distribución como:

$$t = r_{XY}/S_r \tag{3.33}$$

y se compara el valor obtenido con el existente en las tablas para un cierto nivel de significancia α y $N - 2$ grados de libertad: $t_{(\alpha,N-2)}$, que como se sabe, marca el límite de pertenencia de un cierto coeficiente r_{XY} a la distribución muestra de correlaciones procedentes de una población con $\rho = 0$. De esta forma si: $t > t_{(\alpha,N-2)}$ se rechaza la hipótesis nula. La correlación obtenida no procede de una población cuyo valor de ρ es igual a cero. Por lo tanto se determina que las variables tiene correlación.

$t \leq t_{(\alpha,N-2)}$ Se acepta la hipótesis nula. La correlación obtenida procede de una población cuyo valor $\rho = 0$. Por lo tanto las variables no estan relacionadas.

3.5. Inteligencia artificial

Aun en la actualidad no existe una definición exacta que nos permita discernir con claridad que sistemas o programas se consideran inteligentes y cuáles no. Una forma sencilla de definir la inteligencia artificial (AI) sería la capacidad de razonamiento que posee un agente que no está vivo. Sin embargo ante la incapacidad de los humanos de definir con exactitud el propio concepto de inteligencia (natural), hace que el definir inteligencia artificial de manera concisa sea una tarea casi imposible. Por lo que prácticamente cada científico e

investigador del tema puede tener su propio concepto del mismo. Una de las definiciones más aceptadas es la que brinda la *Encyclopedia Of Artificial Intelligence*:

"La AI es un campo de la ciencia y la ingeniería que se ocupa de la comprensión, desde el punto de vista informático, de lo que se denomina comúnmente comportamiento inteligente. También se ocupa de la creación de artefactos que exhiben este comportamiento".

El concepto de Inteligencia Artificial, también conocido por las siglas AI, se le debe al informático estadounidense John McCarthy, quien en el año 1956 lo pronunció por primera vez en una conferencia. Sin embargo no fue el primer científico en trabajar con sistemas inteligentes. Ya en 1943 Warren McCulloch y Walter Pitts, propusieron el primer modelo de red neuronal artificial. Era un modelo simple pero capaz de resolver y aprender funciones lógicas.

A continuación se presenta el concepto básico del funcionamiento de las redes neuronales artificiales.

3.6. Redes neuronales artificiales (ANN)

Las redes neuronales artificiales emulan el comportamiento de sistemas neuronales biológicas a través de *software* o *hardware*. Estas redes pueden aprender las relaciones complejas que existen entre los datos, ya que son eficientes cuando se mapea información numérica no lineal entre entradas y salidas. Esta característica las hace muy útiles en procesos de modelado para los cuales el modelado matemático es difícil o imposible de determinar, como el reconocimiento de patrones para operaciones de clasificación (K. Yang, 2004).

Una red neuronal está compuesta por múltiples capas de nodos interconectados con una función de activación en cada nodo y pesos en los bordes que conectan los nodos de la red. La salida de cada nodo es una función no lineal de todas sus entradas, y la red representa una expansión de la relación no lineal entre x entradas desconocidas, y las salidas y en un espacio abarcado por las funciones de activación (Nunes et al., 2017).

El aprendizaje es visto como una aproximación de la síntesis de una función multidimensional en un espacio dado por la función de activación (Berenji y Wang, 2006).

Se tienen las entradas de la neurona $x_1, x_2, ..., x_m$, las cuales simulan los niveles de una neurona biológica. Cada entrada x_i es multiplicada por el peso correspondiente w_i, enton-

ces el producto $x_i w_i$ alimenta la neurona. Los pesos representan la fuerza de las sinapsis en una neurona natural biológica. La neurona suma todos los productos para $i = 1, m.$, la suma ponderada de los productos es usualmente llamada *net*. Finalmente la neurona brinda su salida y como una función determinada de *net* , ie, $y = f(net)$ (Munakata, 2008).

El proceso de aprendizaje de la ANN se puede dividir principalmente en: supervisado (se cuenta con datos para entrenamiento) y no supervisado. En la primera categoría, la red más utilizada es conocida como Perceptron multicapa con regla de retropropagación, este tipo de red requiere grandes tiempos de aprendizaje pero puede ser bastante precisa (Barrera y Carreón, 2005).

Uno de los algoritmos con aprendizaje de tipo supervisado ampliamente utilizado en aplicaciones diversas como reconocimiento de rostros, detección de fallas y predicción. Debido a su eficiencia y flexibilidad, son las máquinas de soporte vectorial (SVM). Antes de profundizar en el concepto de SVM se presenta la teoría del aprendizaje estadístico.

3.7. Teoría del aprendizaje estadístico (SLT)

Presentada a finales de 1960 la teoría del aprendizaje estadístico, hasta el momento se ha utilizado como una herramienta para el análisis teórico del problema de la estimación de la función de una determinada colección de datos, y para la creación y el razonamiento de nuevos algoritmos prácticos, por ejemplo los mencionados anteriormente ANN y SVM. El siguiente análisis es tomado de (Le, 2012) y (Deng, Tian, y Zhang, 2013).

3.7.1. Problema de aprendizaje

El modelo de problema de aprendizaje se puede describir utilizando sus tres componentes: 1) Un generador de un vector aleatorio x que puede ser representado por una distribución fija pero desconocida $P(x)$, 2) Un supervisor que describe cómo asignar la etiqueta y al vector x de acuerdo con una distribución condicional $P(y\,|x)$, fija pero desconocida y 3) Una máquina de aprendizaje la cual puede ser caracterizada por una familia de funciones $f(x, \alpha), \alpha \in \Lambda$. La tarea del aprendizaje estadístico es elegir de un conjunto de todos los parámetros Λ un parámetro α_0 tal que su función correspondiente $f(x, \alpha_0)$ pueda

predecir una salida y en la mejor forma posible. La nominación se basa en un conjunto de entrenamiento $T = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$ que se obtiene de una distribución de probabilidad independiente e idénticamente distribuida (i.i.d.) $P(x, y) = P(x) \times P(y|x)$.

3.7.2. Minimización del riesgo general

Para nominar el mejor parámetro, se debe proponer un criterio para hacer una comparación de los parámetros. El riesgo general, que se refiere a todo el espacio de características, se considera un buen criterio para este proceso de nominación. El riesgo general de una función $f(x, \alpha)$ sobre un dominio T puede ser definido como la probabilidad de $f(x) \neq y$ donde, x y y son aleatoriamente seleccionados de $X, Y = \{-1, 1\}$. El riesgo general se define como:

$$R(\alpha) = \int L(y, f(x, \alpha)) dP(x, y) \tag{3.34}$$

Sin embargo la evaluación del riesgo general mediante la ecuación. 3.34 no es factible ya que la distribución de probabilidad i.i.d. $P(x, y)$ es desconocida y sólo se es capaz de recolectar un conjunto de datos en un momento dado, es decir un conjunto de entrenamiento T. En otras palabras, sólo sabemos que existe una distribución en la que los puntos de entrenamiento (x_i, y_i), $i = 1, ..., l$ y el punto característico (x, y) son independientes e idénticamente generados, pero no sabemos que distribución es. Por lo tanto, es necesario introducir principios de aprendizaje que dependen de un conjunto de entrenamiento finito para guiar el proceso de aprendizaje.

3.7.3. Minimización del riesgo empírico

El problema de clasificación en SLT es encontrar una función de decisión que (aproximadamente) alcance el menor riesgo esperado. Dado que la distribución que genera el conjunto de entrenamiento es desconocida, el riesgo esperado no es calculable y, en consecuencia, es imposible encontrar la función de decisión directamente a través del riesgo esperado. Así que se debe encontrar algún sustituto computable y aproximado para reemplazar el riesgo esperado, este es el llamado riesgo empírico.

Dada una función $f(x, \alpha)$ y un conjunto de entrenamiento $T = \{(x_1, y_1), (x_2, y_2), ..., (x_l, y_l)\}$ el riesgo empírico se puede calcular como:

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^{l} L\left(y_i, f(x_i, \alpha)\right) \tag{3.35}$$

Es posible calcular el error empírico para un conjunto de entrenamiento dado. Como resultado, no es difícil determinar el mejor parámetro α_l en términos de minimizar el riesgo empírico sobre el conjunto de entrenamiento de tamaño l. El principio de minimización del riesgo empírico (ERM) tiene como objetivo minimizar el riesgo empírico sobre el conjunto de entrenamiento de tamaño l. Similar a la regla de los grandes números en estadística, al aumentar el tamaño l del conjunto de entrenamiento, se espera que el riesgo empírico converja al riesgo general.

Teoría clave de la teoría del aprendizaje. Sea $Q(z, \alpha)$, $\alpha \in \Lambda$ una familia de funciones de pérdida que tiene una pérdida delimitada para la medida de probabilidad $P(z)$

$$A \leq \int Q(z, \alpha) dP(z) \leq B \tag{3.36}$$

Entonces, para que el ERM sea consistente, es necesario y suficiente que el riesgo empírico $R_{emp}(\alpha_l)$ converja uniformemente hacia el riesgo actual $R(\alpha)$ de la siguiente manera:

$$\lim_{l \to \infty} \Pr\left[\sup_{\alpha \in \Delta} \left(R(\alpha) - R_{emp}(\alpha)\right) > \varepsilon\right] = 0, \text{ para todo } \varepsilon > 0 \tag{3.37}$$

Donde $\Pr[E]$ representa la probabilidad de un evento E. El teorema anterior se llama teorema clave, ya que proporciona la condición necesaria y suficiente para la consistencia del principio ERM.

3.7.4. Dimensión Vapnik-Chervonenkis (VC)

El principio de ERM tiende a elegir la función de decisión que minimiza el riesgo empírico $R_{emp}(\alpha)$ sobre un conjunto candidato de funciones de decisión. Para medir la complejidad de una familia de funciones se introdujo la dimensión VC la cual se define como sigue:

Definición 3.7.1. La dimensión VC h es el número máximo de vectores $x_1, x_2, ..., x_h$ que

pueden ser separados en 2^h formas usando funciones de dicha familia de funciones.

Definición 3.7.2. La dimensión VC es el tamaño del subconjunto más grande del espacio de entrada T para el cual la restricción de $Q(z, \alpha)$, $\alpha \in \Lambda$ a ese subconjunto es el conjunto de todas las funciones $-1, 1$.

3.7.5. Límites del riesgo general

Es imposible evaluar el riesgo general como se muestra en la Ecuación 3.34, pero se pueden estimar los límites del riesgo general. Estos límites juegan un papel importante en el parámetro de guía del proceso de selección. Para encontrar el mejor parámetro, es necesario encontrar un parámetro que minimice el límite superior en lugar del que minimiza el riesgo general. Existen diversos límites propuestos, como el propuesto por V. Vapnik en (Vapnik, 2000).

Con probabilidad de al menos $1 - \eta$, la siguiente desigualdad dice:

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{B\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{B\varepsilon}}\right) \tag{3.38}$$

donde $R(\alpha)$, $R_{emp}(\alpha)$ son el riesgo general y empírico respectivamente. $\varepsilon = 4\frac{h\left(ln\frac{2l}{h}+1\right)-\ln\ \eta}{l}$, para problemas de clasificación $B = 1$.

En dicho límite aparece una compensación entre la complejidad (dimensión VC h) de la familia de funciones $f(x, \alpha)$ y la diferencia (gap) entre $R(\alpha)$ y su límite superior. Es decir si h es grande, $R_{emp}(\alpha)$ es pequeño pero el gap puede ser grande. Por otra parte si h es pequeña, el gap es pequeño pero el riesgo empírico $R_{emp}(\alpha)$ puede ser grande. Por lo tanto se opta por una familia de funciones que tenga balance entre $R_{emp}(\alpha)$ y el gap, es decir una familia de funciones con complejidad muy baja pero aún capaz de describir bien los datos.

3.7.6. Principio de Inducción de Minimización del Riesgo Estructural

El principio de minimización del riesgo estructural (SRM) se emplea para controlar el balance entre el riego empírico y la confianza de la VC (gap). En los límites superiores cuando h se incrementa suavemente, el riesgo empírico tiende a decrecer pero la confianza de la VC se incrementa. Idealmente se puede encontrar un h particular que puede minimizar la suma del riesgo empírico y la confianza de la VC. Desafortunadamente no se puede variar h suavemente ya que es un entero. En cambio, al aplicar el principio de SRM una estructura se presenta dividiendo toda la clase de funciones en subconjuntos anidados. Para cada subconjunto, la dimensión h se debe poder calcular o al menos estimar. La aplicación del principio SRM consiste entonces en elegir un subconjunto y posteriormente elegir una función de este subconjunto para que el límite superior se minimice. Es importante tener en cuenta que cuando se utiliza el principio SRM los subconjuntos anidados deben construirse antes de recopilar datos y esta secuencia es independiente de los datos.

3.8. Máquinas de soporte vectorial (SVM)

Las máquinas de soporte vectorial son un algoritmo de aprendizaje automático que se utiliza para la clasificación y la regresión de conjuntos de datos de alta dimensionalidad. Este ofrece resultados satisfactorios debido a su capacidad para lidiar con problemas como mínimo local, pocas muestras de datos y problemas no lineales. Las MSV estándar son utilizadas para problemas de clasificación binaria (Elhariri et al., 2014). La idea central de esta técnica es determinar una separación lineal (hiperplano separador) el cual es orientado en dirección tal que su distancia a los puntos de datos más cercanos en cada una de las dos clases (margen) sea la máxima. Los puntos de datos más cercanos son conocidos como vectores de soporte (Andre, Beltrame, y Wainer, 2013).

3.8.1. Caso linealmente separable.

Se tienen vectores de entrada $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, 2, ..., n)$ correspondientes con las etiquetas $\mathbf{y}_i \in \{-1, +1\}$. Existe un plano separador cuya función es

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.39}$$

Donde $\mathbf{w} \in \mathbb{R}^n$ es un vector normal que define la frontera, b es un umbral escalar y $\frac{|b|}{\|\mathbf{w}\|}$ representa la distancia perpendicular desde el hiperplano separador hacia el origen. Mientras que la distancia del hiperplano al margen está dada por $d = \frac{1}{\|\mathbf{w}\|}$.

Dos hiperplanos paralelos pueden ser representados como:

$$\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \tag{3.40}$$

Como se definió anteriormente SVM trata de maximizar la distancia entre dos clases, donde la amplitud del margen entre dos hiperplanos paralelos es $d = \frac{2}{\|\mathbf{w}\|}$, por lo tanto para el caso linealmente separable se puede encontrar el hiperplano óptimo resolviendo el siguiente problema de optimización cuadrático

$$\begin{gathered} \text{mín } \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{sujeto a: } \mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \end{gathered} \tag{3.41}$$

Utilizando multiplicadores de Lagrange $\alpha_i (i = 1, 2, ..., n)$ para la restricción, el problema primal (Ecuación (3.42)) se convierte en encontrar el punto de silla de Lagrange. Por lo tanto se vuelve un problema dual de la forma

$$\begin{aligned} \text{máx } L(\alpha) &= \sum_{i}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j \mathbf{y}_i \mathbf{y}_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ \text{sujeto a: } & \sum_{i=1}^{n} \alpha_i \mathbf{y}_i = 0, \ \alpha_i \geq 0 \end{aligned} \tag{3.42}$$

Resolviendo las condiciones de Karush-Kukn-Tucker (KKT), se tiene

$$\alpha_i \left[\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 \right] = 0 \tag{3.43}$$

Si $\alpha_i > 0$, los puntos correspondientes a los datos son llamados vectores de soporte. Por lo tanto, la solución óptima para el vector normal está dada por

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i \mathbf{y}_i \mathbf{x}_i \tag{3.44}$$

Donde N es el número de vectores de soporte. De la ecuación (3.43), eligiendo cualquier vector de soporte $(\mathbf{x}_k, \mathbf{y}_k)$, se obtiene $b^* = \mathbf{y}_k - \mathbf{w}^* \cdot \mathbf{x}_k$. Después $(\mathbf{w}^*, b^*)$ es determinado, a partir de la ecuación (3.44), el hiperplano separador óptimo puede escribirse como

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{sgn} \left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i (\mathbf{x} \cdot \mathbf{x}_i) + b^* \right) \\ \text{donde sgn(.) es la función signo, y } \mathbf{x} &\in \begin{cases} +1 & \text{si} \quad f(\mathbf{x}) > 0 \\ -1 & \text{si} \quad f(\mathbf{x}) < 0 \end{cases} \end{aligned} \tag{3.45}$$

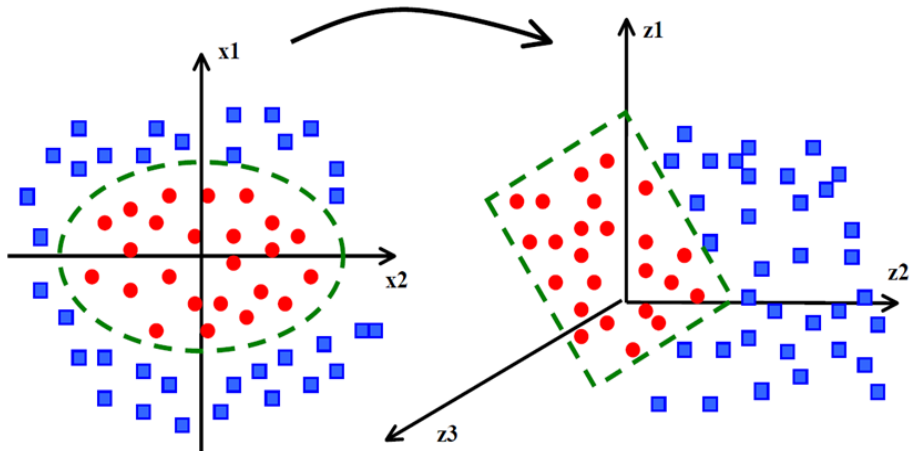

Figura 3.3: Transformación no lineal. Adaptada de (García-Díaz, 2005)

3.8.2. Caso no linealmente separable.

Las MSV también pueden manejar casos donde los datos no son linealmente separables. Estas intentan mapear el vector de entrada $\mathbf{x}_i \in \mathbb{R}^d$ sobre un espacio de mayor dimensionalidad. Este proceso está basado en la elección de una función kernel. Algunas de las funciones kernel (K) más usadas son

- Kernel lineal
- Kernel polinomial
- Función de base radial
- Kernel sigmoideo

Por lo tanto el hiperplano óptimo (ec(3.46)) toma la forma

$$f(\mathbf{x}) = \mathbf{sgn}\left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i \cdot K(\mathbf{x} \cdot \mathbf{x}_i) + b^*\right) \tag{3.46}$$

La Figura 3.3 muestra un ejemplo de una transformación no lineal.

Por otra parte, el problema de la clasificación de una clase o descripción de datos es un tipo especial de problema de clasificación. En el cual siempre se trata con un problema

de clasificación de dos clases, donde cada una de estas dos clases tienen un significado especial. Estas son la clase objetivo y los datos atípicos.

Clase objetivo: Se supone que esta clase se muestrea bien, en el sentido de la cantidad sustanciosa de datos para entrenamiento disponibles. No significa necesariamente que el muestreo del conjunto de entrenamiento se haga completamente de acuerdo con la distribución objetivo que se encuentra en la práctica. Puede ser que el usuario muestreó la clase de destino de acuerdo con su idea de lo representativos que son estos objetos. Sin embargo, se supone que los datos de entrenamiento reflejan el área que los datos objetivo cubren en el espacio de características.

Datos atípicos: Esta clase puede ser muestreada muy escasamente, o puede estar totalmente ausente. Puede ser que esta clase sea muy difícil de medir, o podría ser muy costoso hacer las mediciones en este tipos de objetos. En principio, un clasificador de una clase debe ser capaz de trabajar, sólo sobre la base de ejemplos de destino. Otro caso extremo es también posible, cuando los valores atípicos son tan abundantes que no es posible un buen muestreo de estos (Tax y Duin, 2004).

Existen algoritmos diseñados para llevar a cabo la descripción de los datos como Support Vector Data Description, Mahalanobis-Taguchi system, entre otros.

3.9. Support Vector Data Description (SVDD)

SVDD plantea un enfoque similar a las máquinas de soporte vectorial (SVM), el cual estima un hiperplano para separar las clases con el margen máximo. Sin embargo, en vez de usar un hyperplano, SVDD usa una hiperesfera con el volumen máximo para encontrar un límite cerrado que contenga casi a todos los objetos de interés. Se tiene un conjunto de entrenamiento $\mathbf{x}_i$, $i = 1, ..., N$ para los cuales se desea obtener una descripción. Para obtener la descripción, se define un modelo que brinde un límite cerrado alrededor de los datos, es decir una hiperesfera. Esta se caracteriza por un centro $\mathbf{a}$ y un radio $R > 0$. Ahora se minimiza el volumen de la hiperesfera mediante la minimización de R^2 con el requisito de que la hiperesfera contenga todos objetos de entrenamiento $\mathbf{x}_i$ (Lei et al., 2016).

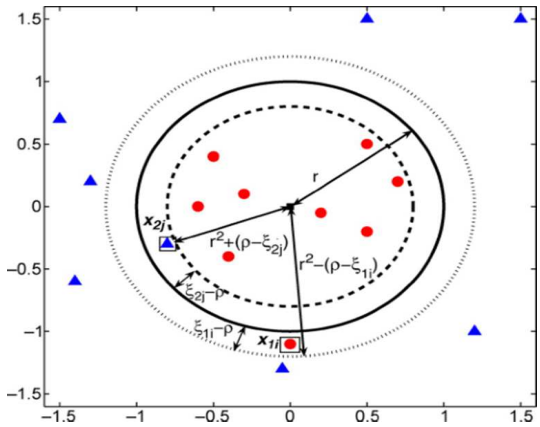

Figura 3.4: Hiperesfera y margen (Tingting Mu, 2009).

Analogamente a SVM se define la función a minimizar:

$$\begin{aligned} \min\ F(R, \mathbf{a}, \xi_i) &= R^2 + C\sum_i \xi_i \\ \text{s.a. } \|\mathbf{x}_i - \mathbf{a}\|^2 &\leq R^2 + \xi_i,\ \xi_i \geq 0\ \forall i \end{aligned} \tag{3.47}$$

Donde ξ_i son variables de holgura, y el parámetro C se encarga de compensar el volumen de la hiperesfera y los errores (S. Jiang y Benlian, 2010). Si C es muy grande, entonces la exactitud en la clasificación es muy alta en la etapa de entrenamiento pero muy baja en la etapa de prueba. Si por el contrario C es muy pequeña la exactitud en la clasificación es no satisfactoria, haciendo al modelo inútil (Deng et al., 2013).

Usando multiplicadores de Lagrange:

$$\begin{aligned} (\mathrm{R,a,}\alpha_i, \gamma_i, \xi_i) \quad &= \quad R^2 \ + \ C\sum_i \xi_i \ - \ \sum_i \alpha_i \\ \left\{R^2 + \xi_i - \left(\|\mathbf{x}_i\|^2 - 2\mathbf{a}\cdot\mathbf{x}_i + \|\mathbf{a}\|^2\right)\right\} &- \sum_i \gamma_i \xi_i \end{aligned} \tag{3.48}$$

Resolviendo las condiciones KKT:

$$\frac{\partial L}{\partial R} = 0 : \qquad \sum_i \alpha_i = 1 \tag{3.49}$$

$$\frac{\partial L}{\partial \mathbf{a}} = 0 : \quad \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \sum_i \alpha_i \mathbf{x}_i \tag{3.50}$$

$$\frac{\partial L}{\partial \xi_i} = 0 : \qquad C - \alpha_i - \gamma_i = 0 \tag{3.51}$$

de la ecuación 3.51 $\alpha_i = C - \gamma_i$ y dado que $\alpha_i \geq 0$, $\gamma_i \geq 0$, por lo tanto γ_i es removido cuando se tiene

$$0 \leq \alpha_i \leq C \tag{3.52}$$

Sustituyendo 3.49, 3.50 y 3.51 en 3.48 se obtiene

$$\begin{aligned} \max \quad & L = \sum_i \alpha_i \left(\mathbf{x}_i \cdot \mathbf{x}_i\right) - \sum_{i,j} \alpha_i \alpha_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \\ \text{s.a.} \quad & \begin{cases} 0 \leq \alpha \leq C \\ \sum_i \alpha_i = 1 \end{cases} \end{aligned} \tag{3.53}$$

Cuando un objeto $\mathbf{x}_i$ cumple las desigualdad $\|\mathbf{x}_i - \mathbf{a}\| < R^2 + \xi_i$, la restricción se satisface y el multiplicador de Lagrange correspondiente será cero ($\alpha_i = 0$). Para los objetos que satisfagan la igualdad $\|\mathbf{x}_i - \mathbf{a}\| = R^2 + \xi_i$ el multiplicador de Lagrange será distinto de cero ($\alpha_i > 0$). Solo los objetos $\mathbf{x}_i$ con $\alpha_i > 0$ son necesarios en la descripción y estos serán llamados vectores de soporte de la descripción (SV's) (Yin et al., 2014).

Para clasificar un objeto $\mathbf{z}$, se calcula su distancia hacia el centro de la hiperesfera. Se determina que el objeto pertenece a la descripción o clase objetivo cuando su distancia es menor o igual al radio:

$$\|\mathbf{z} - \mathbf{a}\|^2 = (\mathbf{z} \cdot \mathbf{z}) - 2 \sum_{i=1}^{l} \alpha_i \left(\mathbf{z} \cdot \mathbf{x}_i\right) + \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \leq R^2 \tag{3.54}$$

Note que en las ecuaciones 3.53 y 3.54 los objetos $\mathbf{x}_i$ solo aparecen en la forma de productos interiores con otros objetos ($\mathbf{x}_i \cdot \mathbf{x}_j$). Análogamente a SVM los productos interiores pueden

ser reemplazados por funciones kernel para obtener descripciones más flexibles:

$$L = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_i \sum_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$
$$\text{s.t.} \qquad 0 \leq \alpha \leq C \tag{3.55}$$

Donde $K(\mathbf{x}_i, \mathbf{x}_j)$ es una función kernel que calcula los productos interiores de los vectores y α_i representan a los multiplicadores de Lagrange (S. Jiang y Benlian, 2010).

3.9.1. SVDD con ejemplos negativos

Cuando se tiene información acerca de los objetos que se desea estén fuera de la descripción (llamados ejemplos negativos), estos se pueden incorporar al proceso de entrenamiento para mejorar la descripción. En contraste con los ejemplos de entrenamiento de la clase objetivo los cuales deben de estar dentro de la hiperesfera, los ejemplos negativos deben estar fuera de esta. Dicha descripción de los datos difiere del enfoque de SVM ya que SVDD siempre obtiene un límite cerrado alrededor de una sola de las clases (clase objetivo). Por otro lado SVM estándar solo distingue entre dos clases y no es capaz de detectar datos atípicos, los cuales no pertenecen a ninguna de las clases (Tax y Duin, 2004).

Los objetos de la clase objetivo son enumerados por los índices i, j y los ejemplos negativos por p, q. Por conveniencia se asume que los primeros se etiquetarán como $y_i = 1$, mientras que los ejemplos negativos $y_p = -1$. De igual manera se permiten errores tanto en la clase objetivo como en los datos atípicos, por lo tanto se introducen variables de holgura ξ_i y ξ_p:

$$\min F(R, \mathbf{a}, \xi_i, \xi_p) = R^2 + C_1 \textstyle\sum_i \xi_i + C_2 \sum_p \xi_p$$
$$\text{s.t.} \ \|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \|\mathbf{x}_p - \mathbf{a}\|^2 \geq R^2 - \xi_p \tag{3.56}$$

$\xi_i \geq 0, \xi_p \geq 0 \quad \forall i, p$. Estas restricciones son incorporadas en la ecuación 3.56 y los respectivos multiplicadores de Lagrange α_i, α_p, γ_i, γ_p se introducen:

$$L\left(R, \mathbf{a}, \xi_i, \xi_p, \alpha_i, \alpha_p, \gamma_i, \gamma_p\right) = R^2 + C_1 \sum_i \xi_i + C_2 \sum_p \xi_p - \sum_i \gamma_i \xi_i - \sum_p \gamma_p \xi_p$$
$$- \sum_i \alpha_i \left[R^2 + \xi_i - (\mathbf{x}_i - \mathbf{a})^2\right] - \sum_p \alpha_p \left[(\mathbf{x}_p - \mathbf{a})^2 - R^2 + \xi_p\right] \quad (3.57)$$

with $\alpha_i \geq 0, \alpha_j \geq 0, \gamma_i \geq 0, \gamma_i \geq 0$.

Resolviendo las condiciones KKT:

$$\sum_i \alpha_i - \sum_l \alpha_l = 1 \quad (3.58)$$
$$\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i - \sum_l \alpha_l \mathbf{x}_l \quad (3.59)$$
$$0 \leq \alpha_i \leq C_1, \ 0 \leq \alpha_l \leq C_2 \ \ \forall i, l \quad (3.60)$$

Sustituyendo 3.58, 3.59 y 3.60 en 3.57 se obtiene

$$\min L = \sum_i \alpha_i \left(\mathbf{x}_i \cdot \mathbf{x}_i\right) - \sum_p \alpha_p \left(\mathbf{x}_p \cdot \mathbf{x}_p\right) - \sum_i \sum_j \alpha_i \alpha_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$$
$$+2 \sum_p \sum_j \alpha_p \alpha_j \left(\mathbf{x}_p \cdot \mathbf{x}_j\right) - \sum_p \sum_q \alpha_p \alpha_q \left(\mathbf{x}_p \cdot \mathbf{x}_q\right) \quad (3.61)$$

Finalmente se definen nuevas variables $\alpha_i' = y_i \alpha_i$ donde el índice i ahora enumera tanto a la clase objetivo como a los ejemplos negativos. De esta manera SVDD con ejemplos negativos se resuelve de manera idéntica a SVDD estándar.

3.10. Distancia Mahalanobis en algoritmos de vectores de soporte

Note que los modelos basados en vectores de soporte presentados en las secciones 3.8 y 3.9, se construyen en base a distancias euclideas. Por lo tanto, dichos modelos son influenciados fácilmente por datos atípicos presentes en las muestras de datos, lo cual puede ocasionar errores en la predicción. Con el propósito de lidiar con este inconveniente, estudiosos del área han propuesto diversos procedimientos. Por ejemplo, llevar a cabo un pre procesamiento el cual consta de normalizar los datos para remover los datos atípicos y posteriormente implementar los métodos basados en vectores de soporte estándar. Sin embargo los datos atípicos no son el único inconveniente que se presenta al trabajar con dichos métodos. Debido a que la distancia euclidea es a menudo sub óptima en problemas de aprendizaje de alta dimensión, es necesario encontrar una forma eficiente y robusta para resolver estos problemas. La distancia Mahalanobis es superior que la distancia euclidea al tratar con datos atípicos, además de tomar en consideración la correlación existente entre las características. Esto ha motivado a desarrollar procedimientos que hacen uso de dicha distancia en tareas de clasificación y reconocimiento de patrones. Algunos de estos enfoques proponen el uso de kernels basados en distancia mahalanobis en lugar de los kernels tradicionales, otros hacen uso de dicha distancia en la etapa de post procesamiento. Uno de los enfoques más robusto es el propuesto por **Xunkai Wei et al. (2010)** donde los modelos basados en vectores de soporte se construyen en base a la distancia mahalanobis, lo cual los hace más robustos, ya que no es necesario implementar etapas de pre o post procesamiento para remover datos atípicos. Estos modelos son capaces de identificarlos y considerarlos automáticamente mejorando y simplificando el proceso de aprendizaje. Los modelos que se presentan en esta sección fueron desarrollados por **Xunkai Wei et al. (2010)**. Las demostraciones matemáticas correspondientes se presentan en (Wei et al., 2010),

3.10.1. Distancia Mahalanobis (MD)

Sea $\mathbf{X} \in R^{m \times n}$ una matriz de muestras que contiene m observaciones aleatorias y n características, la MD escalada se calcula de la siguiente manera:

$$\mathbf{MD} = \frac{1}{n}\left(z_0^T R^{-1} z_0\right) \tag{3.62}$$

donde z_0 es un vector columna de variables estandarizadas y R es la matriz de correlación correspondiente. Sí se define la media muestral μ y la matriz de covarianzas $\mathbf{\Sigma}$ como:

$$\begin{aligned} \mu = & \ \tfrac{1}{m}\mathbf{X}^T\mathbf{1} \\ \mathbf{\Sigma} = & \ \tfrac{1}{m}\mathbf{X}^T\mathbf{X} - \tfrac{1}{m^2}\mathbf{X}^T\mathbf{1}\mathbf{1}^T\mathbf{X} \end{aligned} \tag{3.63}$$

donde $\mathbf{1}$ es un vector de unos de tamaño $m \times 1$. Usando la matriz de covarianzas la distancia Mahalanobis de una muestra $\mathbf{x}$ hacia la población $\mathbf{X}$ es:

$$MD^2(\mathbf{x}, \mathbf{X}) = (\mathbf{x} - \mu)^T \mathbf{\Sigma^{-1}} (\mathbf{x} - \mu) \tag{3.64}$$

Si la matriz de covarianzas es singular se puede usar la pseudoinverza $\mathbf{\Sigma}^+ = \mathbf{P\Sigma^{-1}P}^T$ usando la inverza de los valores propios.

3.10.2. Distancia Mahalanobis en el espacio de referencia kernel

En espacios de referencia de alta dimensionalidad definido por las funciones kernel, la distancia Mahalanobis puede ser representada en términos de productos interiores. Suponga que $\mathbf{X^\Phi}, \mu^\Phi, \mathbf{\Sigma^\Phi}$ son la matriz de muestras, el vector de medias y la matriz de covarianzas en el espacio de características respectivamente. La matriz kernel centrada se define como:

$$\mathbf{K} - \frac{1}{m}\mathbf{EK} - \frac{1}{m}\mathbf{KE} + \frac{1}{m^2}\mathbf{EKE} \tag{3.65}$$

donde $\mathbf{E}$ es una matriz de unos de tamaño $n \times n$, $\mathbf{K} = \{\mathbf{\Phi}(\mathbf{x}_i) \cdot \mathbf{\Phi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1,\ldots,m}$

es una matriz simétrica de dimensión $n \times n$. Usando 3.63 se obtiene:

$$\mathbf{\Sigma}^{\mathbf{\Phi}} = \mathbf{X}^{\mathbf{\Phi}}\mathbf{Z}^2\mathbf{X}^{\mathbf{\Phi}} \tag{3.66}$$

donde $\mathbf{Z} = \left(\frac{1}{m}\left(\mathbf{I} - \frac{1}{m}\mathbf{E}\right)\right)^{\frac{1}{2}}$ es una matriz simétrica de tamaño $m \times m$, $\mathbf{I}$ es una matriz unitaria $m \times m$. La distancia Mahalanobis en el espacio de referencia se define como

$$\begin{aligned} MD^2(\mathbf{\Phi}(\mathbf{x}), \mathbf{X}^{\mathbf{\Phi}}) &= (\mathbf{\Phi}(\mathbf{x}) - \mu^{\mathbf{\Phi}})\mathbf{\Sigma}^{\mathbf{\Phi}-1}(\mathbf{\Phi}(\mathbf{x}) - \mu^{\mathbf{\Phi}}) \\ &= (K(\mathbf{X}, \mathbf{x}) - \tfrac{1}{m}\sum_{i=1}^{m} K(\mathbf{X}, \mathbf{x}_i))^T \times (\mathbf{Z}\mathbf{M}^{-2}\mathbf{Z}) \times \\ &\quad (K(\mathbf{X}, \mathbf{x}) - \tfrac{1}{m}\sum_{i=1}^{m} K(\mathbf{X}, \mathbf{x}_i)) \end{aligned} \tag{3.67}$$

donde $\mathbf{x}_i$ es la i-ésima mustra de $\mathbf{X}$, $\mathbf{M} = \mathbf{ZKZ}$ es una matriz simétrica semidefinida y por lo tanto $\mathbf{M}^{-2}$ se puede calcular mediante la descomposición en valores singulares, es decir, $\mathbf{M} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, $\mathbf{M}^{-2} = \mathbf{U}\mathbf{\Lambda}^{-2}\mathbf{U}^T$.

3.10.3. Mahalanobis Data Description Machine (MDDM)

Usando la distancia Mahalanobis como métrica en lugar de la distancia euclidea, el problema a resolver para SVDD estándar (Ec. 3.47) se vuelve:

$$\begin{aligned} &\min\ F(R, \mathbf{a}, \xi_i) = R^2 + C\sum_i \xi_i \\ &\text{s.a. } (\mathbf{x}_i - \mathbf{a})^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \xi_i,\ \xi_i \geq 0\ \forall i \end{aligned} \tag{3.68}$$

donde $\mathbf{\Sigma}$ es la matriz de covarianza. El problema dual de SVDD estándar (Ec. 3.53) ahora se define como:

$$\begin{aligned} \max \quad & L = \sum_i \alpha_i \mathbf{x}_i^T \mathbf{\Sigma}^{-1} \mathbf{x}_i - \sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{\Sigma}^{-1} \mathbf{x}_j \\ \text{s.a.} \quad & \begin{cases} 0 \leq \alpha \leq C \\ \sum_i \alpha_i = 1 \end{cases} \end{aligned} \tag{3.69}$$

Debido a que en general existe incertidumbre al estimar $\mathbf{\Sigma}$. Se asume que sólo se sabe

que $\mathbf{\Sigma}$ está dentro del conjunto

$$\left\{\mathbf{\Sigma} : \left\|\mathbf{\Sigma} - \mathbf{\Sigma}^0\right\|_F \leq r\right\} \tag{3.70}$$

donde $r > 0$ es una constante de regularización y $\|\cdot\|_F$ es la norma Frobenius, $\mathbf{\Sigma}^0$ se calcula mediante la Ec. 3.63.

Ahora el problema de optimización a resolver es

$$\begin{aligned} &\min\ F(R, \mathbf{a}, \xi_i) = R^2 + C\sum_i \xi_i \\ &\text{s.a. } (\mathbf{x}_i - \mathbf{a})^T \mathbf{\Sigma}_r^{-1} (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \xi_i,\ \xi_i \geq 0\ \forall i \end{aligned} \tag{3.71}$$

donde $\mathbf{\Sigma}_r^{-1} = \mathbf{\Sigma}^0 + r\mathbf{I}$, por lo tanto la matriz de covarianza siempre es no singular para $r > 0$. El dual correspondiente es:

$$\begin{aligned} &\max \quad L = \sum_i \alpha_i \mathbf{x}_i^T \mathbf{\Sigma}_r^{-1} \mathbf{x}_i - \sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{\Sigma}_r^{-1} \mathbf{x}_j \\ &\text{s.a.} \qquad \begin{cases} 0 \leq \alpha \leq C \\ \sum_i \alpha_i = 1 \end{cases} \end{aligned} \tag{3.72}$$

Usando el kernel la Ec. 3.72 se vuelve

$$\begin{aligned} &\text{máx} \quad L = \sum_i \alpha_i \left(K(\mathbf{x}_i, \mathbf{x}_i) - K(\mathbf{x}_i, \mathbf{X})\mathbf{Z}\mathbf{M}_r^{-1}\mathbf{Z}K(\mathbf{X}, \mathbf{x}_i)\right) \\ &\qquad - \sum_{i,j} \alpha_i \alpha_j \left(K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{X})\mathbf{Z}\mathbf{M}_r^{-1}\mathbf{Z}K(\mathbf{X}, \mathbf{x}_j)\right) \\ &\text{s.a.} \qquad \begin{cases} 0 \leq \alpha \leq C \\ \sum_i \alpha_i = 1 \end{cases} \end{aligned} \tag{3.73}$$

donde $\mathbf{M}_r = r\mathbf{I} + \mathbf{ZKZ}$. La inversa de la matriz simétrica definida positiva $\mathbf{M}_r$ puede ser estimada mediante la descomposición en valores singulares.

3.11. Funciones kernel

En las secciones previas se mostró que los modelos basados en vectores de soporte pueden ser más robustos y flexibles al sustituir los productos internos de los modelos $(\mathbf{x}_i \cdot \mathbf{x}_i)$ y $(\mathbf{x}_i \cdot \mathbf{x}_j)$ por $K(\mathbf{x}_i, \mathbf{x}_j)$, donde K es una función kernel. La idea es mapear los datos del espacio de entrada hacia espacios de alta e incluso infinita dimensionalidad (Espacios de Hilbert); un algoritmo de aprendizaje se ejecuta en el espacio de características por el cual un modelo simple o lineal puede convertirse en un modelo complicado o no lineal en el espacio de entrada. La transformación no lineal Φ, la cual mapea el espacio de entrada en el espacio de referencia, es invisible en la función de decisión final. En su lugar $K(\cdot, \cdot)$ la cual mapea $\mathcal{X}^2$ en R donde $\mathcal{X}$ es el dominio de los datos en el espacio de entrada, aparece como reemplazo. Es decir

$$K(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')) \tag{3.74}$$

Definición 3.11.1. Matriz de Gram: para una función $K(\mathbf{x}, \mathbf{x}') : R^n \times R^n \to R$ y l puntos $x_1, ..., x_l \in R^n$, la matriz K de dimensión $l \times l$ donde $K_{ij} = K(x_i, x_j)$, es conocida como la matriz de Gram de la función $K(\mathbf{x}, \mathbf{x}')$ con respecto a $x_1, ..., x_l$.

Una función simétrica $K(\mathbf{x}, \mathbf{x}')$ definida en $R^n \times R^n$ es un kernel si y solo si la matriz de Gram de $K(\mathbf{x}, \mathbf{x}')$ con respecto a $x_1, ..., x_l$ es semidefinida positiva para cualquier l y cualquier $x_1, ..., x_l \in R^n$.

3.12. Construcción de kernels

La pregunta natural es ¿que tipo de funciones son kernels?. Los siguientes tres pasos responden a esta pregunta:

1. Investigue los kernels básicos.
2. Investigue las operaciones que mantienen los kernels.
3. A partir de los kernels básicos construya los kernels usados comunmente aplicando las operaciones del paso anterior.

3.12.1. Kernels básicos

La función $K(\mathbf{x}, \mathbf{x}' = \mathbf{x} \cdot \mathbf{x}')$ definida en $R^n \times R^n$ es un kernel. Sea $\Phi(\mathbf{x}) = x$, entonces $K(\mathbf{x}, \mathbf{x}')$ se puede expresar como

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \mathbf{x}') \tag{3.75}$$

3.12.2. Operaciones en kernels

Suponga que $K_1(\mathbf{x}, \mathbf{x}')$ y $K_2(\mathbf{x}, \mathbf{x}')$ son kernels en $R^n \times R^n$, entonces su suma

$$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$$

y su producto

$$K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}')K_2(\mathbf{x}, \mathbf{x}')$$

son kernels también.

3.12.3. Kernels usados

Ahora se pueden contruir dos kernel comunmente usados basados en los kernels básicos mediante las operaciones definidas anteriormente.

Kernel polinomial:
Suponga que d es un entero positivo, entonces la función polinomial homogenea de orden d es

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d \tag{3.76}$$

y la función polinomial no homogenea de orden d

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + 1)^d \tag{3.77}$$

son kernels.

Kernel función de base radial Gaussiana:
El kernel función de base radial Gaussiana con parámetro σ

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\left\|\mathbf{x} - \mathbf{x}'\right\| / \sigma^2) \tag{3.78}$$

es un kernel. Las definiciones presentadas en esta sección fueron tomadas de (Le, 2012), (Bounsiar y Madden, 2014) y (Deng et al., 2013).

Como se mostró anteriormente el problema de optimización a resolver tanto para las máquinas de soporte vectorial (SVM) como para Support Vector Data Description y Mahalanobis Data Description Machine (Ecuaciones 3.43, 3.53 y 3.73 respectivamente) es un problema de optimización cuadrática. Por lo tanto en la siguiente sección se presentarán conceptos relevantes a dichos problemas.

3.13. Problema de optimización

Los conceptos y el análisis de esta sección se tomaron de (Deng et al., 2013), (Lee, Tam, y Yen, 2005), (Merino-Maestre, 2013), y (Gill y Wong, 2015).

Un problema de optimización general se define como:

$$\min \quad f_0(x),\ x = (x_1, ..., x_n)^T \ \in R^n, \tag{3.79}$$

$$s.t. \qquad f_i(x) \leq 0,\ i = 1, ..., m, \tag{3.80}$$

$$h_i(x) = 0,\ i = 1, ..., p. \tag{3.81}$$

En este caso x es un vector y es conocido como variable de optimización del problema, a la función f_0 en 3.79 se le conoce como función objetivo. Las ecuaciones 3.80 y 3.81 son llamadas restricciones de desigualdad e igualdad respectivamente. $f_i(x) \leq 0,\ i = 1, ..., m$ y $h_i(x) = 0,\ i = 1, ..., p$ son llamadas funciones de restricción. Al problema 3.79, 3.80 y 3.81 se le llama problema sin restricciones si $m + p = 0$.

Definición 3.13.1. Punto factible y región factible: Un punto que satisface todas las restricciones se llama un punto factible. El conjunto de todos estos puntos constituye la región factible D

$$D = \{x | f_i(x) \leqslant 0, i = 1, ..., m; h_i(x) = 0, i = 1, ..., p; x \in R^n\} \tag{3.82}$$

Definición 3.13.2. Valor óptimo: El valor óptimo p^* del problema 3.79-3.81 se define como el ínfimo de la función objetivo f_0 en la región factible D, cuando esta no es vacía y p^* se define como infinito:

$$p^* = \begin{cases} \inf \{f_0(x) | x \in D\}, & \text{cuando } D \neq \phi, \\ \infty, & \text{Si no} \end{cases} \tag{3.83}$$

Definición 3.13.3. Solución global y solución local: Considere el problema 3.79-3.81. El punto x^* se conoce como solución global si x^* es un punto factible y

$$f_0(x^*) = \inf \{f_0(x) | x \in D\} = p^*, \tag{3.84}$$

donde D es la región factible. El punto x^* es conocido como solución local si x^* es un punto factible y existe un $\varepsilon > 0$ tal que

$$f_0(x^*) = \inf \{f_0(x) | x \in D; \|x - x^*\| \leqslant \varepsilon\} \tag{3.85}$$

El conjunto de todas las soluciones globales y el conjunto de todas las soluciones locales se llaman el conjunto de soluciones correspondientes respectivamente. En concreto una solución local es un punto en el que el valor de la función objetivo es menor o igual que en todos los demás puntos factibles en su vecindad. La mejor de todas las soluciones locales es la solución global.

3.13.1. Programación no lineal

Debido a que la programación lineal presenta restrictividad de la hipótesis de linealidad y la dificultad de definir una única función objetivo, surge la programación no lineal. Un supuesto importante en programación lineal es que todas sus funciones (objetivo y restricciones) son lineales. Este supuesto se cumple en el caso de muchos problemas prácticos,

con frecuencia no es así. Por lo que es necesario abordarlos desde la programación no lineal. Si un problema de programación no lineal no tiene restricciones, el hecho de que la función objetivo sea cóncava garantiza que un máximo local es un máximo global. (De igual manera, una función objetivo convexa asegura que un mínimo local es un mínimo global.) Si existen restricciones, se necesita una condición más para dar esta garantía, a saber, que la región factible sea un conjunto convexo. Por esta razón, los conjuntos convexos tienen un papel fundamental en la programación no lineal.

Definición 3.13.4. Sea una función de una sola variable $f(x)$ y se considera cada par de valores x, x' y x''. Se dice que

1. Es una **función convexa** si se cumple

$$f(\lambda x'' + (1-\lambda)x') \leq \lambda f(x'') + (1-\lambda)f(x'),\ \forall \lambda \in (0,1). \tag{3.86}$$

2. Es una **función estrictamente convexa** si se cumple

$$f(\lambda x'' + (1-\lambda)x') < \lambda f(x'') + (1-\lambda)f(x'),\ \forall \lambda \in (0,1). \tag{3.87}$$

3. Es una **función cóncava** si se cumple

$$f(\lambda x'' + (1-\lambda)x') \geq \lambda f(x'') + (1-\lambda)f(x'),\ \forall \lambda \in (0,1). \tag{3.88}$$

4. Es una **función estrictamente cóncava** si se cumple

$$f(\lambda x'' + (1-\lambda)x') > \lambda f(x'') + (1-\lambda)f(x'),\ \forall \lambda \in (0,1). \tag{3.89}$$

$\lambda x'' + (1-\lambda)x'$ es el segmento que une los puntos x' y x'', por lo que en una función convexa el segmento queda por encima de la función, mientras que en una función cóncava el segmento queda por debajo. La Figura 3.5 muestra los diferentes casos de funciones.

Existen diversos tipos de problemas de programación no lineal, algunos de ellos son:

- Optimización no restringida
- Optimización restringida linealmente

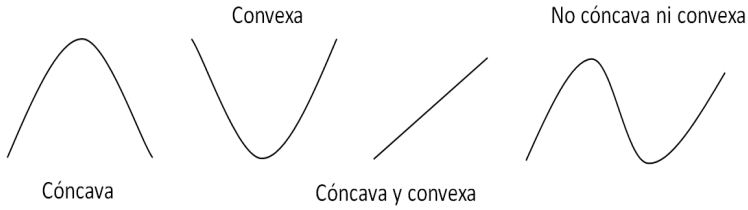

Figura 3.5: Convexidad de funciones.

- Programación cuadrática
- Programación convexa
- Programación no convexa
- programación geométrica

Para mayores detalles acerca de este tipo de problemas de programación no lineal ver (Merino-Maestre, 2013). A continuación se presenta la definición y otros conceptos acerca de la programación cuadrática, problema de programación no lineal de interés en esta investigación debido a la base teórica de las máquinas de soporte vectorial.

3.13.2. Programación cuadrática

Un programa cuadrático (QP) implica la minimización o maximización de una función objetivo cuadrática sujeta a restricciones lineales de igualdad y desigualdad sobre las variables. Los QP surgen en muchas áreas, incluyendo economía, ciencia aplicada e ingeniería. Las aplicaciones importantes incluyen análisis de cartera, máquinas de soporte vectorial, análisis estructural y control óptimo. La programación cuadrática también forma un componente computacional principal de muchos métodos de programación cuadrática secuencial para la programación no lineal. Los métodos de interior y los métodos de conjunto activo son dos enfoques alternativos para manejar las restricciones de desigualdad de un QP. Se dice que $f : R^n \to R$ es una función cuadrática si existe una matriz $D \in R^{n\times n}$, un vector $c \in R^n$ y un número real α tal que:

$$f(x) = \frac{1}{2}x^T Dx + c^T x + \alpha \tag{3.90}$$

Para todo $x \in R^n$. Sí

$$D = \begin{bmatrix} d_{11} & ... & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & ... & d_{nn} \end{bmatrix}, \; c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \; x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Entonces significa que 3.90 es:

$$f(x) = \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_ix_j\right) + \sum_{i=1}^{n} c_ix_i + \alpha$$

Debido a que $x^TDx = \frac{1}{2}x^T(D + D^T)x$ para cada $x \in R^n$, en 3.90 es válido reemplazar D por la matriz simétrica $\frac{1}{2}(D + D^T)$. Por esta razón, se asume que la matriz cuadrada en la representación de la función cuadrática es simétrica.

Definición 3.13.5. Sea $f : R^n \to R$ una función cuadrática se dice que es

- **definida positiva** si $f(x) > 0$, $\forall x \in R^n \setminus\{0\}$
- **semidefinida positiva** si $f(x) \geq 0$, $\forall x \in R^n$
- **definida negativa** si $f(x) < 0$, $\forall x \in R^n \setminus\{0\}$
- **semidefinida negativa** si $f(x) \leq 0$, $\forall x \in R^n \setminus\{0\}$
- **indefinida** si no entra en ninguna de las clasificaciones anteriores

En las matrices definidas positivas todos los valores propios λ_j son positivos. Por el contrario, la caracterización de matrices definidas negativas todos los valores propios son negativos.

EL problema QP genérico se define como:

$$\begin{array}{ll} \text{minimizar} & f(x) = \frac{1}{2}x^TDx + c^Tx + \alpha \\ \text{Sujeto a :} & Ax \geq b \\ & x \geq 0 \end{array} \tag{3.91}$$

donde $b \in R^n$ es un vector columna y $A = (a_{ij}) \in R^{m\times n}$. Los demas términos fueron definidos anteriormente.

Condiciones de Karush-Kuhn-Tucker (KKT)

Las condiciones necesarias que deben satisfacer los óptimos de problemas de optimización no lineal con restricciones de desigualdad fueron publicadas por primera vez (1939) en la tesis de Maestría de William Karush y de manera independiente por Kuhn y Tucker (1951). Las condiciones de Karush-Kuhn-Tucker (KKT) son una generalización del método de los multiplicadores de Lagrange para restricciones de desigualdad. El siguiente análisis fue tomado de (Merino-Maestre, 2013), (Seebach, 2006) y (Taha, 2012).

Considere el problema de optimización

$$\begin{array}{ll} \min & f\left(x_1, x_2, ..., x_n\right) \\ \text{sujeto a} & g_1\left(x_1, x_2, ..., x_n\right) \leq 0 \\ & g_2\left(x_1, x_2, ..., x_n\right) \leq 0 \\ & \vdots \\ & g_m\left(x_1, x_2, ..., x_n\right) \leq 0 \end{array} \tag{3.92}$$

Suponga que $f(x), g_j(x)$ con $j = 1, ..., m$ son funciones diferenciables que satisfacen ciertas condiciones de regularidad. Entonces, x^* puede ser una solución óptima para el problema de programación no lineal, sólo si existen m números $\alpha_1, ..., \alpha_m$ que satisfagan las siguientes condiciones.

Condiciones de KKT necesarias

$$\left.\begin{array}{lll} \frac{\partial \mathcal{L}}{\partial x_i} & = \frac{\partial f(x^*)}{\partial x_i} + \sum\limits_{j=1}^{m} \alpha_j \frac{\partial g_j(x^*)}{\partial x_i} & \geq 0 \\ x_i^* \frac{\partial \mathcal{L}}{\partial x_i} & = x_i^* \left(\frac{\partial f(x^*)}{\partial x_i} + \sum\limits_{j=1}^{m} \alpha_j \frac{\partial g_j(x^*)}{\partial x_i} \right) & = 0 \\ x_i^* \geq 0 & & \end{array}\right\} \forall i = 1, ..., n \tag{3.93}$$

$$\left.\begin{array}{lll} \frac{\partial \mathcal{L}}{\partial \alpha_i} & = g_j(x^*) - b_j & \leq 0 \\ \alpha_j \frac{\partial \mathcal{L}}{\partial \alpha_j} & = \alpha_j (g_j(x^*) - b_j) & = 0 \\ x_i^* \geq 0 & & \end{array}\right\} \forall j = 1, ..., m \tag{3.94}$$

Condiciones KKT suficientes

Sea $f(x)$ cóncava y $g_1(x), ..., g_m(x)$ convexas, donde estas funciones satisfacen las condiciones de regularidad y x^* satisface las condiciones KKT. Entonces, x^* es una solución óptima. Para ejemplos numéricos ver (Merino-Maestre, 2013) y (Seebach, 2006).

3.13.3. Métodos de punto interior para optimización

Durante los últimos veinte años, ha habido una revolución en los métodos utilizados para resolver problemas de optimización. A principios de los años ochenta, la programación cuadrática secuencial y los métodos lagrangianos aumentados fueron favorecidos para problemas no lineales, mientras que el método simplex era básicamente incuestionable para la programación lineal. Desde entonces los métodos de punto interior (IPM) se han aplicado prácticamente en cada área de optimización continua (Nemirovski y Todd, 2009). La idea principal de los métodos de punto interior es iterar en el interior del conjunto factible y luego acercarse progresivamente al límite (si el mínimo realmente se encuentra en el límite). Lo hacen transformando el problema original en una secuencia de problemas de optimización sin restricciones, en los que el objetivo utiliza una función de barrera que va al infinito en los límites de la región factible. Al reducir la resistencia de esta barrera en cada optimización subsiguiente, la secuencia de mínimos se aproxima arbitrariamente al mínimo del problema original. Este análisis se tomó de (Hauser, 2012), (Palencia, Hing, y Rojas, 2008), (Altman y Gondzio, 1999).

Funciones objetivo con barreras logarítmicas

Para el problema con restricciones de desigualdad

$$\begin{aligned} \min_{x \in R^n} \quad & f(x) \\ \text{s.a.} \quad & g_i(x) \leq 0, \ i = 1, ..., m \end{aligned} \tag{3.95}$$

se modifica f para obtener una función barrera:

$$f_\alpha(x) = f(x) - \alpha \sum_{i=1}^{m} \ln(-g_i(x)) \tag{3.96}$$

Debido a que el logaritmo va hacia $-\infty$ cuando su argumento se aproxima a cero, la función objetivo modificada se vuelve arbitrariamente grande cuando x se aproxima a los límites de la región factible. El parámetro λ proporciona la barrera. Su significado es que en la medida que α se aproxima a cero, el óptimo de $f_\alpha(x)$ se aproxima al óptimo de 3.95, y $x^*_\alpha \equiv \text{mín}_x f_\alpha(x)$:

$$\lim_{\alpha \to 0} f(x^*_\alpha) = f^*. \tag{3.97}$$

Optimizando la función barrera

Utilizando el método de Newton para encontrar el mínimo de $f_\alpha(x)$. El gradiente es

$$\nabla f_\alpha(x) = \nabla f(x) - \alpha \sum_{i=1}^{m} \frac{\nabla g_i(x)}{g_i(x)} \tag{3.98}$$

Un inconveniente es que x^*_α se acerca cada vez más al límite, la estabilidad numérica del problema de optimización restringido se vuelve cada vez peor debido a que el denominador se acerca a cero. Por lo tanto, sería muy difícil aplicar gradiente descendiente a una pequeña tolerancia. En lugar se introduce otro conjunto de multiplicadores KKT $\lambda = (\lambda_i, ..., \lambda_n)$ con $\lambda_i \equiv \alpha / g_i(x)$. Así en el espacio (x, λ), se quiere encontrar una raíz de la ecuación

$$\nabla_x f_\alpha(x, \lambda) = \nabla f(x) - \sum_{m}^{i=1} \nabla g_i(x) \lambda_i = 0 \tag{3.99}$$

sujeto a la igualdad

$$\lambda_i g_i(x) = \alpha \text{ para } i = 1, ..., m$$

Estas ecuaciones son numéricamente más estables que 3.98 cerca del límite. Para encontrar una raíz $(^*_\alpha, \lambda^*_\alpha)$ donde ambas ecuaciones se satisfacen, se aplica el método de Newton. Primero, se obtienen las derivadas parciales de las funciones anteriores:

$$\nabla^2_{xx} f_\alpha(x,\lambda) = \nabla^2 f(x) - \sum_{i=1}^{m} \nabla^2 g_i(x)\lambda_i, \tag{3.100}$$

$$\nabla_\lambda \nabla_x f_\alpha(x) = -\sum_{i=1}^{m} \nabla g_i(x) \tag{3.101}$$

y

$$\nabla_x(\lambda_i g_i(x) - \alpha) = \lambda_i \nabla g_i(x) \tag{3.102}$$

$$\frac{\partial}{\partial \lambda_i}(\lambda_i g_i(x) - \alpha) = g_i(x) \text{ para } i = 1, ..., m \tag{3.103}$$

En la iteración actual (x_t, λ_t), se deriva el paso de Newton $(\Delta x, \Delta\lambda)$ del siguiente sistema de ecuaciones:

$$\begin{bmatrix} H & -G \\ diag(\lambda_t)G^T & diag(g) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x_t) + G\lambda_t \\ diag(g)\lambda_t - \alpha\mathbf{1} \end{bmatrix} \tag{3.104}$$

Donde H es la ecuación 3.100 evaluada en (x_t, λ_t), g es el vector $m \times 1$ de $g_i's$. G es una matriz $n \times m$, donde la i-ésima columna es $\nabla g_i(x_t)$. $diag(u)$ produce una matriz cuadrada cuya diagonal es un vector u. Y $\mathbf{1}$ es un vector $m \times 1$ de unos.

Inicialización en un punto factible

Los métodos de punto interior deben ser inicializados en un punto interior, o bien la función barrera está indefinida. Para encontrar un punto factible inicial x podemos utilizar la siguiente optimización:

$$\begin{gathered} \min_{x \in R^n} \sum_{i=1}^{m} s_i \\ \text{Tal que} \\ g_i(x) - s_i \leq 0,\ i = 1, ..., m s_i \geq 0 \end{gathered} \tag{3.105}$$

Si el problema es factible, entonces los s_i óptimos serán todos 0. Para encontrar un

conjunto inicial de s_i para cualquier x dado, se ajusta $s_i = \text{máx}(g_i(0), 0)$ (Hauser, 2012). Para detalles específicos acerca de algoritmos de punto interior ver (Qian, Liao, y Sun, 2017), (Vanderbei y Carpenter, 1993), (Domínguez-Moreno, 2004).

3.14. Error en clasificadores

3.14.1. Matriz de confusión

Para conocer el resultado de la clasificación de un algoritmo, uno de los métodos más usados en el aprendizaje supervisado es la matriz de confusión. Este es una herramienta visual en donde cada columna de la matriz representa el número de predicciones para cada clase, y cada fila representa la clase real. Mediante esta matriz se vuelve fácil visualizar si el algoritmo de clasificación tuvo errores en la misma.

3.14.2. Error en clasificadores de una clase

Con el fin de encontrar un buen clasificador de una clase, dos tipos de errores deben de ser minimizados:

1. La fracción de falsos positivos
2. La fracción de falsos negativos

en la figura 3.6 se muestran las situaciones posibles de clasificación para clasificadores de una clase (OCC).

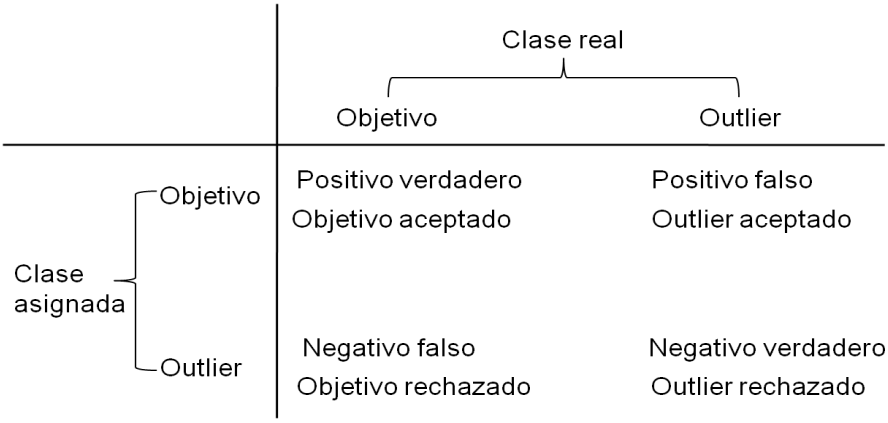

Figura 3.6: Posibles situaciones de clasificación para algoritmos OCC.

3.14.3. Validación cruzada de K subconjuntos (k-fold cross validation)

En este procedimiento el conjunto de datos es dividido en K grupos. De estos, K-1 se utilizan para entrenar al clasificador, y el restante se usa para evaluarlo. Este proceso es repetido K veces, entonces se promedian los resultados para obtener una métrica final. La ventaja de este método es que al contar con un conjunto de datos de entrenamiento limitado, es posible estimar el desempeño del clasificador sobre un conjunto de datos independiente. En la práctica, este procedimiento de validación cruzada se aplica una y otra vez. No sólo para evaluar y comparar el rendimiento de los clasificadores, sino también para optimizar los hiperparámetros de los algoritmos si es necesario.

3.15. Selección de características

Una característica es una propiedad individual mensurable del proceso que se observa. Usando un conjunto de características, cualquier algoritmo de aprendizaje de máquina puede realizar la tarea de clasificación. En los últimos años en las aplicaciones de aprendizaje de máquina o reconocimiento de patrones, el dominio de las características se han expandido de decenas a cientos de variables o características utilizadas en esas aplicaciones (Chandrashekar y Sahin, 2014). El enfoque de la selección de características es obtener un subconjunto de variables de la entrada que puedan describir eficientemente los datos

de entrada mientras se reducen los efectos del ruido o variables irrelevantes y aun así proporcionar buenos resultados de predicción. La selección de características también se conoce como selección de variables, selección de atributos o selección de subconjuntos de variables, en los campos de aprendizaje automático y estadísticas. Algunas de las ventajas de llevar a cabo este procedimiento se enlistan a continuación (Saha y Rajasekaran, 2015).

1. Tiempos de entrenamiento más cortos: cuando se eliminan las características irrelevantes y redundantes, el tiempo de aprendizaje disminuye.
2. Mejora de la creación de modelos: el modelo construido es más preciso y eficiente.
3. Generalización mejorada: produce modelos sencillos y generalizados.

Los algoritmos de selección de características suelen clasificarse como pertenecientes a los de tipo filtros (filters), envoltura (wrappers) o enfoques integrados (embedded approaches). También se construyen combinaciones de enfoques, en los que, por ejemplo, primero se emplea un filtro, luego envoltura, o cuando se usa un envoltura como filtro. A continuación se presenta un breve panorama de dichos tipos; para más detalle ver (Stańczyk y Jain, 2015).

3.15.1. Tipo filtro

Los filtros son procesos completamente ajenos de las técnicas utilizados para la clasificación, trabajando independientemente en su rendimiento y otros parámetros. Pueden tratarse como una especie de procedimientos de pre procesamiento. Estos explotan la información contenida en los conjuntos de datos de entrada buscando, por ejemplo, ganancia de información, entropía, consistencia, entre otros. La naturaleza de los filtros los hace aplicables en todos los casos, sin embargo, el hecho de que ignoran totalmente el rendimiento de un sistema de clasificación que emplea el conjunto de variables seleccionadas, provoca usualmente peores resultados que otros enfoques y esto se considera como una desventaja.

3.15.2. Tipo envoltura

En un enfoque de envoltura para la selección de características, se dice que la mejor evaluación de algún subconjunto candidato de variables se obtiene comprobando su utilidad en la clasificación, ya que la precisión predictiva estimada suele considerarse el indicador más importante de relevancia para los atributos. Dado que el proceso de búsqueda y selección se ajusta a las características específicas del método de clasificación, pueden mostrar un sesgo, dando como resultado un rendimiento alto para dicho clasificador, pero peores resultados para otro, especialmente cuando varían significativamente en propiedades. Es decir, los métodos tipo envoltura tienden a construir conjuntos de atributos adaptados a una tarea en particular y para un sistema en particular.

3.15.3. Soluciones integradas

Algunos predictores tienen sus propios mecanismos inherentes, incorporados en el algoritmo de aprendizaje, dedicados a la selección de características. Cuando este mecanismo se utiliza activamente, se tiene una solución integrada. Tal es el caso de los árboles y algoritmos como random forest.

3.16. Metaheurísticas

La optimización estocástica es la clase general de algoritmos y técnicas que emplean cierto grado de aleatoriedad para encontrar soluciones óptimas (o tan óptimas como sea posible) a problemas difíciles. Las metaheurísticas son los más generales de este tipo de algoritmos, y se aplican a una gama muy amplia de problemas (Luke, 2013). En la literatura no existe una única definición de metaheurísticas, una de ellas es la que brindan Osman y Laporte (Osman y Laporte, 1996): “Una metaheurística es un conjunto de conceptos que pueden ser utilizados para definir métodos heurísticos que se pueden aplicar a un amplio conjunto de problemas diferentes. En otras palabras, un metaheurístico puede ser visto como un marco algorítmico general que puede aplicarse a diferentes problemas de optimización con relativamente pocas modificaciones para hacerlas adaptables a un problema específico”. Los algoritmos metaheurísticos son capaces de resolver problemas complejos de optimización donde otros métodos de optimización no son eficaces o eficientes.

Las metaheurísticas son conocidas por ser uno de los enfoques más prácticos para resolver diversos problemas complejos del mundo real. Estos algoritmos inician el proceso de búsqueda mediante la obtención de una solución inicial o un conjunto inicial de la solución, y posteriormente mejorar la búsqueda guiada por ciertos principios (Onar et al., 2016). El problema a resolver se puede expresar a menudo como un problema de optimización. Aquí se define una función objetivo (o varias funciones de este tipo), o función de coste, que se pretende minimizar o maximizar con respecto a todos los parámetros implicados.

La definición del problema de optimización suele complementarse con información en forma de restricciones. Todos los parámetros de las soluciones adoptadas deben satisfacer estas restricciones, de lo contrario estas soluciones no son factibles. Algunos métodos, particularmente algoritmos genéticos, se han inspirado en principios biológicos tales como selección, cruce y mutación. Sin embargo, algunos otros métodos han intentado aprovechar los comportamientos que han demostrado ser eficientes para la supervivencia y el desarrollo de las poblaciones biológicas. Desde este punto de vista, la optimización de enjambre de partículas (PSO) es el primer método basado en la cooperación sin selección (Siarry, 2016).

3.16.1. Optimización por enjambre de partículas (PSO)

El algoritmo de optimización con enjambre de partículas (PSO) fue desarrollado por J. Kennedy y R. C. Eberhart, el cual es un método de optimización metaheurístico basado en la simulación del comportamiento social de las aves, abejas, bancos de peces, entre otros. A diferencia de otros algoritmos evolutivos como algoritmos genéticos (GA), PSO es más simple ya que no contiene operaciones de mutación o cruce, lo que reduce la complejidad del modelo (Haixiang et al., 2016).

PSO intenta simular gráficamente la gracia y la impredecible coreografía de una parvada de pájaros, con el objetivo de descubrir patrones que gobiernan la habilidad de las aves para volar de una manera sincronizada, y el cambio repentino de dirección y reagrupación en una formación óptima. Los individuos (partículas) en un enjambre de partículas, siguen un comportamiento muy simple: emular el éxito del individuo vecino y el propio. El comportamiento colectivo que se genera de éste simple comportamiento, es con el objetivo de descubrir las regiones óptimas de un alto espacio de búsqueda.

Como se ha mencionado, la técnica PSO se basa en el comportamiento e imitación de enjambres, la cual se basa en una teoría socio-cognoscitiva en dos niveles que abarca tanto al enjambre completo así como a cada partícula del mismo.

El nivel general o alto es el que se observa en la formación de patrones entre los individuos y la habilidad para resolver problemas, es decir, se ocupa de la información que se transmite entre todos los individuos del enjambre, lo cual genera los patrones y convergencia, mientras que el nivel bajo o particular maneja el desempeño individual de cada una de las partículas dentro del enjambre (Rull-FLores, 2013). Para que ambos niveles socio-cognoscitivos se lleven a cabo es necesario que tres principios se cumplan:

1. Evaluar
2. Comparar
3. Imitar

Cada individuo en el enjambre es representado por un vector en un espacio de búsqueda multidimensional. Este vector tiene un vector de velocidad asignado que determina el siguiente movimiento de la partícula. Cada partícula actualiza su velocidad basándose en la velocidad actual y la mejor posición que ha explorado hasta ahora. Y también basándose en la mejor posición explorada por todo el enjambre (E. O. Reséndiz-Flores y Rull-Flores, 2013). PSO fue originalmente desarrollado para espacios de valores continuos, sin embargo muchos problemas son definidos para espacios de valor discreto donde el dominio de las variables es finito.

3.16.2. PSO binario

En 1997, Kennedy and Eberhart (Kennedy y Eberhart, 1995) introducen una versión binaria de PSO para problemas de optimización discretos. En PSO binario (BPSO) cada partícula representa su posición en valores binarios, donde cada valor de la partícula puede ser cambiado de 1 a 0 o viceversa.

Considere un espacio de búsqueda d-dimensional y a la i-ésima partícula del enjambre denotada como $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{id})^t$ cuya velocidad es $\mathbf{v}_i = (v_{i1}, v_{i2}, ..., v_{id})^t$. Por lo tanto

las ecuaciones que guían la dinámica de la partícula en PSO continuo son:

$$\begin{aligned}\mathbf{v}_i(t+1) &= w\mathbf{v}_i(t) + c_1\phi_1(\mathbf{p}_{ibest} - \mathbf{y}_i(t)) \\ &\quad + c_2\phi_2(\mathbf{p}_{gbest} - \mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) &= \mathbf{y}_i(t) + \mathbf{v}_i(t+1)\end{aligned}$$

donde c_1 y c_2 son constantes positivas, y ϕ_1 y ϕ_2 son variables aleatorias con distribución uniforme entre 0 y 1. Las mejores posiciones locales y globales son $\mathbf{p}_{ibest}$ y $\mathbf{p}_{gbest}$ respectivamente. El peso de inercia sobre un nuevo vector se denota por w(Das, Abraham, y Konar, 2008).

La principal diferencia entre PSO y BPSO es la forma en que las velocidades cambian para actualizar las posiciones de las partículas. La versión binaria necesita una función para generar valores en el conjunto binario $\{0, 1\}$ que son los únicos estados posibles de la partícula, por lo tanto la función sigmoide es utilizada para decidir la probabilidad de generar tales números binarios. Esta función se aplica bit a bit en una partícula dada y se define como sigue:

$$sigm(v_{ik}) = \frac{1}{1 + e^{-v_{ik}}}$$

La posición de la partícula correspondiente se actualiza en base a la siguiente regla

$$y_{ik}(t+1) = \begin{cases} 1 & \text{if}\, r < sigm(v_{ik}) \\ 0 & \text{otro} \end{cases}$$

donde r es un número pseudo aleatorio seleccionado de una distribución uniforme entre $[0, 1]$.

Khanesar propone una versión binaria de PSO donde emplea una modificación a la regla para la actualización de la posición:

$$y_{ik}(t+1) = \begin{cases} 1 - y_{ik}(t) & \text{if}\, r < sigm(v_{ik}) \\ y_{ik}(t) & \text{if}\, r > sigm(v_{ik}) \end{cases}$$

3.16.3. PSO binario Gompertz (GBPSO)

El algoritmo GBPSO se basa principalmente en la idea de reemplazar la función sigmoide que se utiliza en el PSO binario estándar para generar números binarios por la función Gompertz (GF). La función de Gompertz se aplica extensamente en bioinformática para modelar el fenómeno de supervivencia de individuos con recursos limitados y condiciones ambientales restringidas, donde la tasa de nacimiento inicialmente aumenta y luego se ralentiza cuando se alcanzan los límites de recursos. La función Gompertz es:

$$Gmpz(x) = ae^{be^{cx}} \tag{3.106}$$

donde a es la asíntota superior fijada como $a = 1$; b, c son números negativos que controlan el desplazamiento en el eje x y la tasa de crecimiento respectivamente. Donde se proponen las siguientes formas:

$$b(t) = -\left(0{,}2 + (2{,}0 - 0{,}2) \cdot \frac{k}{N}\right)$$
$$c(t) = -\left(0{,}2 + (1{,}5 - 0{,}2) \cdot \frac{k}{N}\right)$$

k, N denotan el número actual y el número máximo de iteraciones, respectivamente. En la figura 3.7 se muestra el efecto de la variación de los parámetros b y c. Ya que en PSO binario la probabilidad de cambiar un bit a 0 o 1, en velocidades cercanas a cero es 0.5. En la función de Gompertz, es posible aumentar o disminuir esta probabilidad en el rango (0,1) manipulando el parámetro b como se muestra en la figura 3.7 a). Por ejemplo para $a = 1$, $c = -1$ y b incrementándose en un rango [-0.2, -2.0] la probabilidad de generar 1's será mayor en la etapas iniciales, mientras que en las etapas finales se incrementará la probabilidad de generar 0's. Al disminuir b en el rango [-2.0, -0.2] se incrementa la probabilidad de generar 0's en las etapas iniciales y 1's en las etapas finales. Este último caso es más adecuado para problemas de minimización (binarios y continuos.). Por otro lado a medida que aumentamos c negativamente la inclinación de la función gompertz aumenta y para valores menores de c la GF es como una función lineal mientras que a medida que aumentamos c comienza a tomar forma de "S".

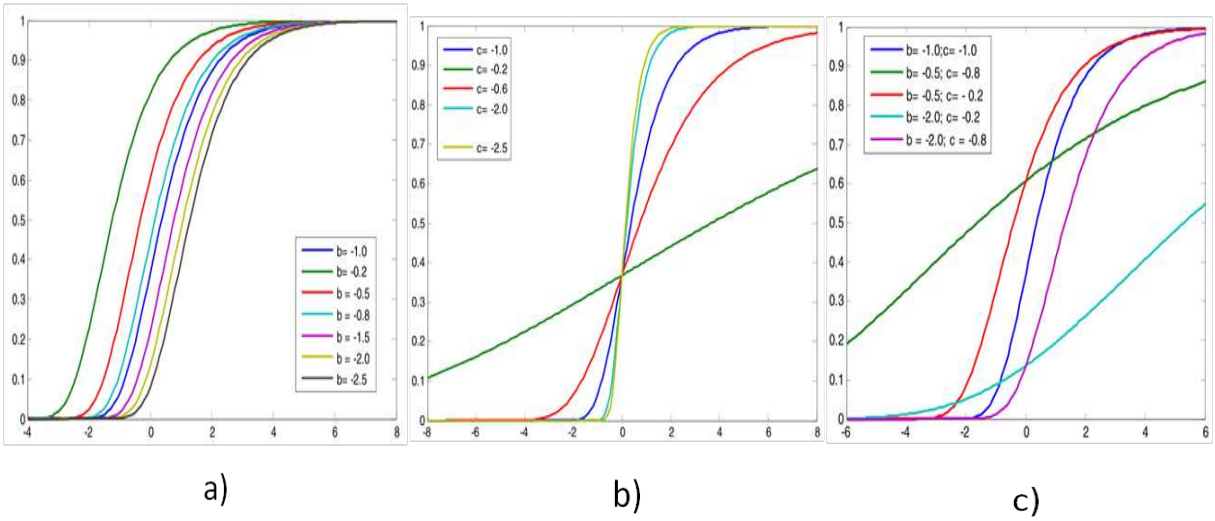

Figura 3.7: Función Gompertz, $a = 1$. a) $c = -1$ y b variable. b) $b = -1$ y c variable. c) b y c variables (Chauhan et al., 2012).

Al variar c entre iteraciones, se logra una mejor exploración en las iteraciones tempranas que a su vez mantiene la diversidad en la población. El hecho más notable es que al efectuar esta variación se obtiene el beneficio de la función lineal y la función sigmoide en un solo algoritmo.

La variación de los parámetros b y c con respecto a las iteraciones ayudará al algoritmo a producir una población más diversificada ya que la función Gompertz mantiene una variación prolongada de las probabilidades dentro de un mayor rango de velocidades máximas (Chauhan, Pant, y Deep, 2012).

3.16.4. Algoritmo genético

El algoritmo genético que se desarrolló en 1975 por John Holland (Holland, 1975), es una de las metaheurísticas basadas en población más utilizadas para resolver diversos problemas de optimización usando técnicas inspiradas en la evolución natural, como la selección, el cruce y la mutación, que se aplican repetidamente a una población de cadenas binarias que representan posibles soluciones. La evolución generalmente comienza en una población de individuos generados aleatoriamente y ocurre en generaciones. En cada generación, la aptitud de cada individuo, se selecciona estocásticamente de la población actual y se

modifica (recombinada y posiblemente mutada aleatoriamente) para formar una nueva población. La nueva población se usa en la próxima iteración del algoritmo. Comúnmente, el algoritmo finaliza cuando se ha producido un número máximo de generaciones o se ha alcanzado un nivel de aptitud satisfactorio para la población (Bhajantri y Nalni, 2014). A continuación se presenta una breve descripción de los pasos de este algoritmo:

1. Inicializar y evaluar: una población de cromosomas se genera al azar y cada cromosoma representa una posible solución. Después de la generación de los cromosomas, cada cromosoma se evalúa utilizando una función objetivo.

2. Operador de cruce: se usa una operación de cruce para generar nuevas soluciones. Para una probabilidad de cruce de P_c, se selecciona primero un número aleatorio uniforme entre 0 y 1 para cada cromosoma. Si P_c es menor que el número aleatorio, el cromosoma se selecciona para la operación de cruce.

3. Operador de mutación: los cromosomas que no se seleccionan para la operación de cruce se pueden seleccionar en el operador de mutación. Para una probabilidad de mutación de P_m, primero se selecciona un número aleatorio uniforme entre 0 y 1 para cada cromosoma. Si P_m es menor que esto, el cromosoma se selecciona para la operación de mutación.

4. Selección: mientras que algunos cromosomas se seleccionan para la operación de cruce y algunos otros para el operador de mutación, el resto se selecciona del proceso de elitismo.

Para detalles específicos ver (Gamarra y Quintero, 2013) y (Yan Xuesong y Qinghua, 2016).

4

Metodología

La metodología propuesta para la detección de fallas y selección de variables es aplicable a diversos sistemas multivariados complejos con sus restricciones y limitaciones propias. Dicha metodología lleva a cabo la optimización del espacio de referencia y la selección de variables con el fin de mejorar la detección de fallas. Los objetivos son la maximización del rendimiento del modelo de clasificación y la minimización del número de variables utilizadas. La precisión y estabilidad del clasificador SVDD[1] para la detección de fallas depende del valor de los hiperparámetros. En este estudio se utiliza un kernel de base radial gaussiano (RBF). Por lo tanto los parámetros C y s deben ser ajustados apropiadamente. Por otro lado, al encontrar el conjunto de variables que contribuyen con la falla, se pueden detectar las variables que están contribuyendo a defectos de calidad. Siendo esto un requisito básico para llevar a cabo la mejora de cualquier proceso de manufactura.

La metodología consta de tres etapas:

1. Pre procesamiento
2. Aprendizaje
3. Monitoreo y prueba

[1]En el capítulo 5 se muestran los resultados de la experimentación usando SVDD y MDDM en la fase de aprendizaje. Por simplicidad, en la descripción y diagrama de la metodología general solo se hace mención a SVDD.

La figura 4.1 muestra el diagrama general de la metodología a implementar. A continuación se presenta una breve descripción de cada una de las etapas.

1. **Pre procesamiento Entrada: datos crudos**

   - **Limpieza de datos:** Eliminación de ruido en señales usando limpieza multivariada Wavelet.
   - **Conjunto de datos en operación normal**
   - **Conjunto de datos en falla**

   Salida: Conjunto de datos estructurado e informativo

2. **Estapa de aprendizaje**

   Entrada: Conjunto de datos estructurado e informativo

   - **Entrenamiento:** Reconocimiento de patrones mediante SVDD con hiperparámetros óptimos.
   - **Selección de variables:** Mejora la precisión en la clasificación con mediante un subconjunto de variables apropiado.

   Salida: Modelo para la detección de fallas y detección de variables relevantes

3. **Etapa de monitoreo y prueba**

   Input: modelo para la detección de fallas y variables relevantes

   - **Datos para verificar:** modo de monitoreo.
   - **Prueba para la detección de fallas (con subconjunto de variables):** Detecta si existe una falla en la muestra.
   - **¿Falla detectada?:** No, regresa al paso 3.1. De otra manera continúa.

   Salida: Alerta en interfaz de usuario gráfica (GUI)

A continuación se explicará la forma en que las técnicas se implementaron para construir la metodología propuesta.

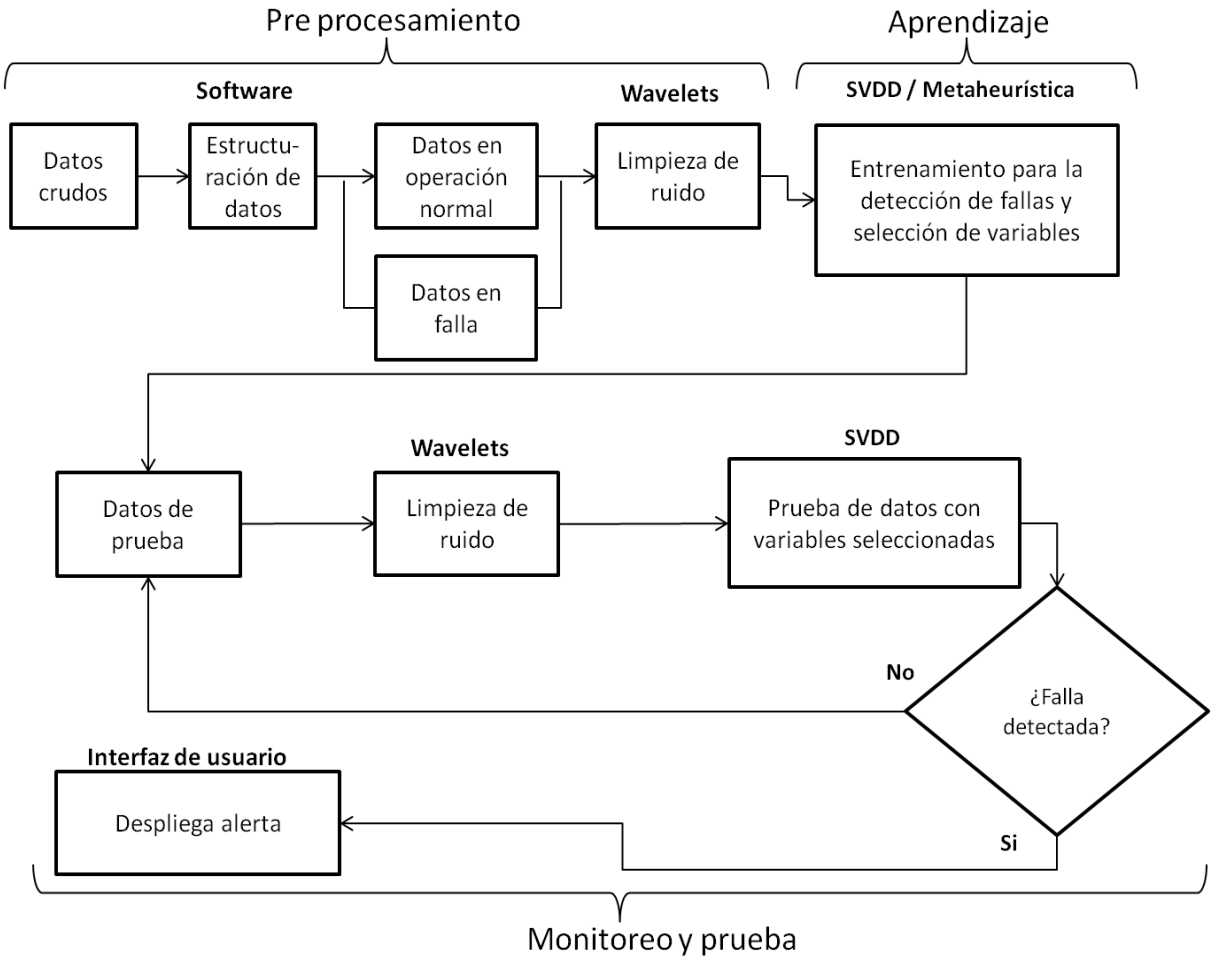

Figura 4.1: Metodología.

4.1. Pre procesamiento

Sea $X \in \mathbb{R}^{n \times p}$ el conjunto de datos en modo normal con p variables y n número de observaciones. Para $X = F(t) + \varepsilon(t)$ se aplica limpieza wavelet multivariada usando análisis de componentes principales con el objetivo de mejorar el proceso de detección de fallas. F es la señal a ser reuperada, ε es el ruido correlacionado. Para este procedimietno es necesario elegir la función wavelet ψ, nivel de descomposición J y seleccionar el procedimiento para elegir el número útil de componenetes principales. Para detalles específicos revisar (Aminghafari et al., 2006). Después de la limpieza se lleva a cabo la etapa de aprendizaje implementando GBPSO-SVDD.

4.2. Aprendizaje

En esta etapa GBPSO-SVDD es implementado con el propósito de optimizar el espacio de referencia para la detección de fallas y de manera simultanea, llevar a cabo la selección de variables relevantes.

Sea $\mathbf{x}_f = (x_f^1, x_f^2, ..., x_f^p)^t$ un vector de dimensión p, donde

$$xf_i = \begin{cases} 0 & \text{si la variable } i \text{ no es seleccionada} \\ 1 & \text{si la variable } i \text{ es seleccionada} \end{cases}$$

$\mathbf{x}_C$ y $\mathbf{x}_s \in \mathbb{R}^3$ son vectores con los valores codificados de C y s. Entonces la partícula GBPSO-SVDD se expresa como: $\mathbf{x_p} = \mathbf{x}_C \cup \mathbf{x}_s \cup \mathbf{x}_f$, ver la figura 4.2. Para optimizar el espacio de referencia en la detección de fallas y de manera simultánea llevar a cabo la selección de variables relevantes, es necesario resolver el siguiente problema:

$$\min\ f(\mathbf{x_p}) = \beta f_1(\mathbf{x}) + (1-\beta)\frac{p_{\text{ seleccionadas}}}{p} \tag{4.1}$$

sujeto a:

$$\sum_{i=1}^{p} x_f^i \leq p, \quad \sum_{i=1}^{p} x_f^i = p_{\text{ seleccionadas}}, \quad f_1(\mathbf{x}) \leq f_1^{\max}$$

Donde $f_1(\mathbf{x_p})$ es el desempeño del modelo SVDD con hiperparámetros seleccionados y subconjunto de variables seleccionadas y $0 \leq \beta \leq 1$

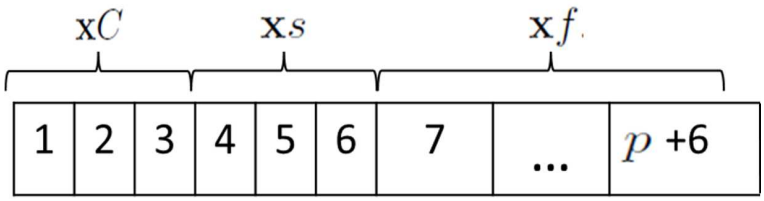

Figura 4.2: Partícula GBPSO-SVDD.

4.2.1. Entrenamiento de SVDD para detección de fallas

SVDD es implementado usando ejemplos negativos. Con el fin de realizar la detección de fallas el radio (R^2) es usado como métrica. Cuando una observación $\mathbf{z}$ es analizada, la distancia de dicha muestra al centro de la hiperesfera es calculada. La observación es rechazada cuando su distancia es mayor a R^2, entonces una falla es detectada. Para mejorar el modelo SVDD, los parámetros C y s deben ser elegidos apropiadamente. Además de encontrar las variables relevantes, el proceso de selección de variables permite obtener un modelo libre de información redundante o no útil. Para llevar a cabo estas tareas de manera simultánea, es necesario tener un criterio de optimización.

Clasificación errónea

En el contexto del desarrollo de un modelo de detección de fallos, es deseable que la escala de medición resulte en una clasificación errónea mínima. La clasificación errónea se define como una falsa alarma causada por la identificación errónea de una observación hacia una clase o categoría cuando pertenece realmente a un grupo diferente. Hay dos tipos de clasificación errónea,

1. Clasificar una observación *normal* como *falla*
2. Clasificar una observación en *falla* como *normal*

Por lo tanto, una función que mide la clasificación errónea total es

$$f_1(\mathbf{x_p}) = a_1 \frac{o_1^e}{o_1} + a_2 \frac{o_2^e}{o_2} \tag{4.2}$$

donde

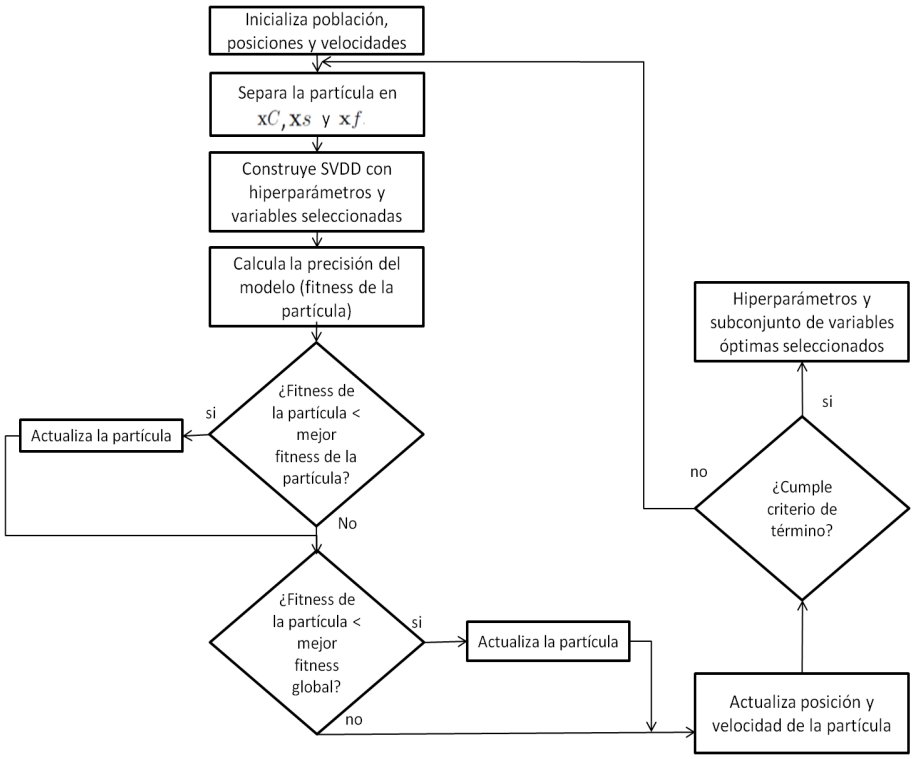

Figura 4.3: Funcionamiento de GBPSO-SVDD

o_1^e = número de observaciones en estado *normal* clasificadas como *falla*

o_2^e = número de observaciones en *falla* clasificadas como *normal*

o_1 = número de observaciones en estado *normal*

o_2 = número de observaciones en *falla*

$a_1 = b_1/(b_1 + b_2)$

$a_2 = b_2/(b_1 + b_2)$

$b_1, b_2 > 0$ son pesos asociados con la clasificación errónea 1 y 2 respectivamente. La figura 4.3 muestra como la combinación GBPSO-SVDD resuelve el problema de la optimización del espacio de referencia y la selección de variables de manera simultánea.

4.3. Monitoreo y prueba

En esta última etapa se toma del proceso un conjunto de datos, el cual contiene n observaciones y las variables seleccionadas ($p_{\text{seleccionadas}}$) con las que se minimizó la ecuacion 4.1. Estos datos son clasificados mediante el modelo SVDD construido con los hiperparámetros seleccionados en la etapa anterior. Las observaciones que superen el valor R^2 serán consideradas como muestras en falla.

5

Experimentación

5.1. Bases de datos de referencia

La metodología propuesta en este trabajo se aplica a dos conjuntos de bases de datos de referencia para llevar a cabo la selección de características relevantes y reducir la clasificación errónea.

5.1.1. Conjunto de datos: iris de Fisher

Iris de Fisher es un conjunto de datos multivariado introducido por Ronald Fisher en su artículo *The use of multiple measurements in taxonomic problems* (1936) (el uso de medidas múltiples en problemas taxonómicos) como un ejemplo de análisis discriminante lineal. A veces se denomina conjunto de datos de Iris de Anderson porque Edgar Anderson recopiló los datos para cuantificar la variación morfológica de las flores de iris de tres especies relacionadas. El conjunto de datos consta de 50 muestras de cada una de las tres especies de iris (iris setosa, iris virginica e iris versicolor). Se midieron cuatro características de cada muestra: la longitud y el ancho de los sépalos y pétalos en centímetros, es decir

- x_f^1: largo de sépalo
- x_f^2: ancho de sépalo
- x_f^3: largo de pétalo

- x_f^4: ancho de pétalo

Basado en la combinación de estas cuatro características, Fisher desarrolló un modelo discriminante lineal para distinguir las especies de las otras (Fisher, 1936).

La metodología propuesta se implementa para identificar el iris setosa del resto de las especies, además de identificar las características relevantes que determinan que flores pertenecen a la especie iris setosa. Debido a que las variables se tratan de mediciones dimensionales y no de señales complejas como temperatura, voltaje, corriente, entre otras. La fase de limpieza multivariada Wavelets se omite. Los parámetros para medir la clasificación errónea se fijan de la siguiente manera $\beta = a_1 = a_2 = 0{,}5$ y los posibles valores para C y s se presentan en la tabla 5.1.

Tabla 5.1: Valores posibles para C y s.

Vectores codificados $\mathbf{x}_C$, $\mathbf{x}_s$	Valor C	Valor s
000	0.0001	1
100	0.001	10
010	0.01	50
110	0.1	100

La tabla 5.2 muestra el valor de clasificación errónea, los valores de C, de s y las variables relevantes seleccionadas por GBPSO-SVDD y GBPSO-MDDM.

Tabla 5.2: Resultados de GBPSO-SVDD y GBPSO-MDDM en iris de Fisher.

Algoritmo	Clasificación erronea	C	s	Variables
GBPSO-SVDD	0.1250	0.0001	1	x_f^3
GBPSO-MDDM	0.2188	0.0001	1	x_f^3

Ambos algoritmos detectan que la variante que minimiza el error en la clasificación es la variable x_3, es decir que el largo de pétalo es la característica relevante para clasificar una muestra como iris setosa. Para observar el efecto de dichas variables, se calcula la matriz de correlación R con los coeficientes de Pearson y Kendall (tablas 5.3 y 5.4 respectivamente) y su respectiva matriz P de valores p para el conjunto de datos iris de Fisher (tablas 5.5 y 5.6).

Tabla 5.3: Valores de la matriz de correlación coeficiente de Pearson.

1	-0.1176	0.8718	0.8179
-0.1176	1	-0.4284	-0.3661
0.8718	-0.4284	1	0.9629
0.8179	-0.3661	0.9629	1

Tabla 5.4: Valores p para coeficientes de Pearson

0	0.1519	0.0000	0.0000
0.1519	0	0.0000	0.0000
0.0000	0.0000	0	0.0000
0.0000	0.0000	0.0000	0

Tabla 5.5: Valores de la matriz de correlación coeficiente de Kendall

1	-0.0770	0.7185	0.6553
-0.0770	1	-0.1860	-0.1571
0.7185	-0.1860	1	0.8069
0.6553	-0.1571	0.8069	1

Tabla 5.6: Valores p para coeficientes de Kendall

0	0.1835	0.0000	0.0000
0.1835	0	0.0013	0.0076
0.0000	0.0013	0	0.0000
0.0000	0.0076	0.0000	0

La matriz P permite probar la hipótesis de no correlación contra la alternativa de que existe una correlación no nula. Cada elemento de dicha matriz es el valor de p para el elemento correspondiente de la matriz de correlación. Si el valor $P(i, j)$ es pequeño, digamos menos de 0.05 para un nivel de confianza del 95 %, entonces la correlación $R(i, j)$ es significativamente diferente de cero. Podemos observar que las variables x_f^1, x_f^4 están altamente correlacionadas con la variable x_f^3, lo que explica el hecho de que al entrenar a los algoritmos SVDD y MDDM solo con dicha variable, ambos logran resultados satisfactorios

en el reconocimiento de la especie iris setosa.

5.1.2. Base de datos cáncer de mama

En el ejemplo anterior se mostró la selección de variables relevantes para la clasificación de especies de plantas en un conjunto de datos de dimensión pequeña (4 características). A continuación se llevará a cabo el mismo estudio en un conjunto de datos de mayor dimensionalidad. Se utilizara la base de datos de cáncer de mama (*Breast Cancer Wisconsin (Original) Data Set*). La base de datos consta de 699 observaciones, 9 características y las correspondientes etiquetas de cada observación, indicando si el cáncer es maligno o benigno. Por lo tanto, el objetivo es diagnosticar cuando el cáncer es maligno e identificar las características medicas (variables) relevantes para realizar dicho diagnóstico con el menor error de clasificación. Para descargar el conjunto de datos y obtener información a detalle revise (Lichman, 2013). A continuación se enlistan las variables medidas para dicho estudio así como el rango de sus valores:

1. x_f^1: Grueso del grupo (1-10)
2. x_f^2: Uniformidad del tamaño de la célula (1-10)
3. x_f^3: Uniformidad de la forma celular (1-10)
4. x_f^4: Adhesión marginal (1-10)
5. x_f^5: Tamaño de célula epitelial único (1-10)
6. x_f^6: Núcleos no cubiertos (1-10)
7. x_f^7: Cromatina suave (1-10)
8. x_f^8: Núcleo normal (1-10)
9. x_f^9: Mitosis (1-10)

Los valores de los parámetros (β, a_1, a_2, C, s) se ajustan como se muestra en 5.1. La tabla 5.7 muestra el valor de clasificación erróneo, los valores de C, de s y las variables relevantes seleccionadas por GBPSO-SVDD y GBPSO-MDDM.

Tabla 5.7: Resultados de GBPSO-SVDD y GBPSO-MDDM en cáncer de mama

Algoritmo	Clasificación erronea	C	s	Variables
GBPSO-SVDD	0.0993	0.0001	1	x_f^2
GBPSO-MDDM	0.0556	0.0001	1	x_f^3

Los resultados obtenidos en el proceso de selección de características son comparados con el estudio realizado por (Glander, 2017), estos se resumen en la tabla 5.8. En dicho estudio se aplica el algoritmo random forest para realizar un análisis conocido como importancia de las variables, dando como resultado que las variables uniformidad del tamaño de la célula y uniformidad de la forma celular (x_f^2 y x_f^3) son las de mayor importancia. Posteriormente realiza selección de características usando tres enfoques distintos, el primero es eliminar las variables altamente correlacionadas, el segundo es realizar eliminación de características recursiva usando random forest y por último se implementa algoritmos genéticos para optimizar el proceso de selección de características. Siendo este último el que ofrece mejores resultados en términos de exactitud en la clasificación, involucrando nuevamente las características (x_f^2 y x_f^3), las cuales fueron seleccionadas por la metodología propuesta. Para ver en detalle los resultados del estudio mencionado revisar (Glander, 2017).

Tabla 5.8: Comparación de resultados

	Exactitud en clasificación	Variables seleccionadas
GBPSO-SVDD	90.07 %	x_f^2
GBPSO-MDDM	94.44 %	x_f^3
Propuesto por (Glander, 2017)	96.00 %	x_f^2 y x_f^3

5.2. Aplicaciones en procesos industriales

A continuación se presenta la metodología propuesta para la detección de fallas y selección de características aplicada a procesos industriales complejos.

5.2.1. Moldeo por inyección de plástico de componentes para pedales de coche.

La metodología para la detección de fallas y selección de variables propuesta ha sido aplicada a un proceso de moldeo por inyección de plástico de pedales de coche en una compañía automotriz de la región. El proceso se lleva a cabo en una máquina de inyección de plástico con la capacidad de producir cuatro piezas por ciclo y guardar la información relevante en cada ejecución, es decir los parámetros utilizados para producir tales piezas se almacenan en una base de datos en tiempo real. Posteriormente, las piezas se clasifican como buenas o malas a través de diversas pruebas internas de calidad por parte de la compañía. Las características involucradas en el proceso de producción son de naturaleza variada, incluyendo temperaturas, presiones, distancias y velocidades. El proceso consta 36 variables enlistadas en la tabla 5.9. La base de datos recolectada consiste en un total de 193 observaciones en modo normal de operación y 25 observaciones en falla.

Resultados numéricos

Para la fase de limpieza de señales, el nivel de descomposición se fijó en $J = 10$ y se utilizó daubechies 5 (db5) como función Wavelet ψ. Para seleccionar el número de componentes principales, se tomó el criterio de Kaiser. El cual selecciona los componentes correspondientes a los valores propios mayores a la media del total de ellos (Karlis et al., 2003). Estos parámetros se tomaron basándose en el estudio realizado por (Aminghafari et al., 2006). La figura 5.1 muestra el proceso de limpieza sobre la señal de valor punta contrapresión (x_f^{17}).

En la etapa de aprendizaje, el problema de optimización combinatoria (ecuación 4.1) se resuelve mediante GBPSO-MDDM, GBPSO-SVDD, BPSO-SVDD y GA-SVDD. Tres configuraciones han sido consideradas para el cálculo de la clasificación errónea total, estas son: 1) $a_1 = 0{,}05$, $a_2 = 0{,}95$, 2) $a_1 = 0{,}95$, $a_2 = 0{,}05$ y 3) $a_1 = a_2 = 0{,}5$ que representan la importancia (porcentaje) de cada función de clasificación errónea. Dado que se da la misma importancia a minimizar la clasificación errónea y la selección de características, el peso en la función objetivo es $\beta = 0{,}5$. El número de partículas del enjambre es de 50 y el número máximo de iteraciones es de 80, $c_1 = c_2 = 2$ y los posibles valores para C y s se presentan en la tabla 5.11. Los valores de los parámetros así

Tabla 5.9: Descripción de variables

	Descripción
x_f^1	[F°], Boquilla 1
x_f^2	[F°], Boquilla 2
x_f^3	[Percent], Potencia calefactora zona 1
x_f^4	[Percent], Potencia calefactora zona 2
x_f^5	[Percent], Potencia calefactora zona 3
x_f^6	[Percent], Potencia calefactora zona 4
x_f^7	[Percent], Potencia calefactora zona 5
x_f^8	[Percent], Potencia calefactora zona 6
x_f^9	[in], Valor de posición de molde
x_f^{10}	[in], Valor punta carrera apertura
x_f^{11}	[US ton], Valor punta fuerza cierre
x_f^{12}	[US ton], Valor real fuerza cierre
x_f^{13}	[s], Valor real tiempo protección molde
x_f^{14}	[F°], Temperatura aceite
x_f^{15}	[F°], Traversa
x_f^{16}	[s], Valor real tiempo enfriamiento
x_f^{17}	[psi], Valor punta contrapresión
x_f^{18}	[in^3], Volumen husillo final postpresión
x_f^{19}	[psi], Valor punta postpresión
x_f^{20}	[in^3/s], Potencia dosificación
x_f^{21}	[psi], Presión al conmutar
x_f^{22}	[s], Tiempo ciclo hasta final desmoldeo
x_f^{23}	[lbf-ft], Valor medio momento giro
x_f^{24}	[lbf-ft], Valor punta momento giro
x_f^{25}	[psi], Valor punta presión especifica de inyección
x_f^{26}	[in^3], Valor real volumen dosificación
x_f^{27}	[in^3], Valor real volumen inyección
x_f^{28}	[s], Valor real tiempo dosificación
x_f^{29}	[s], Valor real tiempo inyección
x_f^{30}	[F°], Volumen de inyección
x_f^{31}	[F°], Zona cilindro 1
x_f^{32}	[F°], Zona cilindro 2
x_f^{33}	[F°], Zona cilindro 3
x_f^{34}	[F°], Zona cilindro 4
x_f^{35}	[ft/s], Revoluciones valor punta
x_f^{36}	[Wh], Trabajo inyección

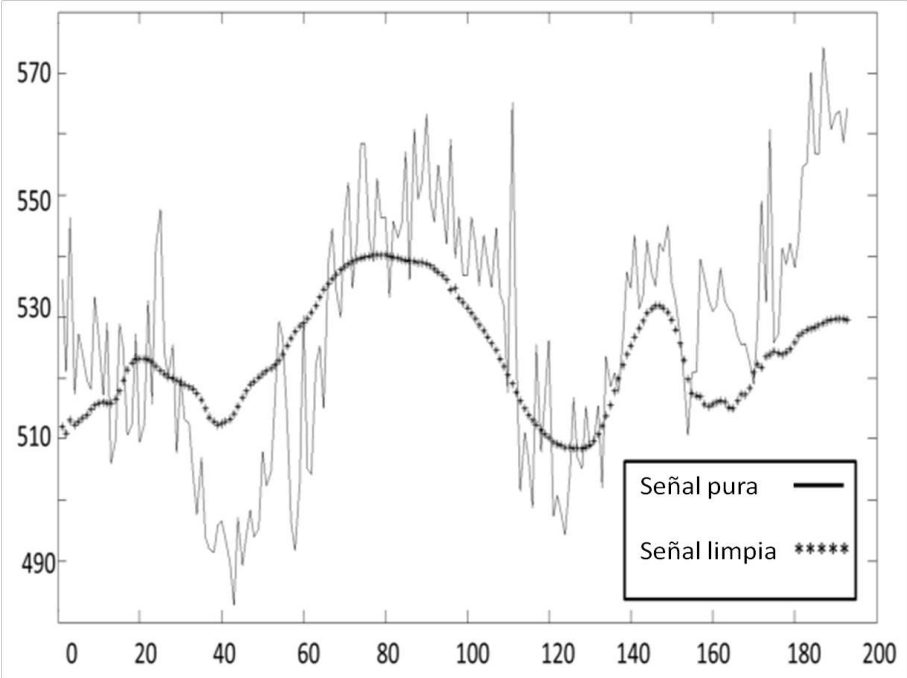

Figura 5.1: Limpieza Wavelet en la señal valor punta contrapresión (x_f^{17}).

como las tres configuraciones consideradas, se tomaron de las investigaciones presentadas en (E. O. Reséndiz-Flores y Rull-Flores, 2013) y (E. Reséndiz-Flores y López-Quintero, 2016). Para el algoritmo genético $P_c = 0{,}8$, $P_m = 0{,}2$, el tamaño de la población es 50 y el número máximo de iteraciones es igual a 80. Además SVDD fue implementado para la detección de fallas en su forma estándar es decir, utilizando el total de las variables, esto con el fin de verificar la efectividad de llevar a cabo el proceso de selección de variables relevantes. Al realizar un análisis de correlación se encuentra que existen tres grupos de variables altamente correlacionadas. Estos grupos se presentan en la Tabla 5.10.

Tabla 5.10: Variables correlacionadas caso 2.

Variables correlacionadas
x_f^{18}, x_f^{27}
x_f^{19}, x_f^{21}, x_f^{36}
x_f^{25}, x_f^{36}

Para el cálculo de los valores de clasificación errónea de cada uno de los algoritmos se implementó validación cruzada de 10 iteraciones (10-fold cross validation). La tabla 5.12

Tabla 5.11: Valores posibles para C y s.

Vectores codificados $\mathbf{x}_C$, $\mathbf{x}_s$	Valor C	Valor s
000	0.0001	1
100	0.001	10
010	0.01	50
110	0.1	100
001	1	200
101	10	300
011	100	400
111	150	450

Tabla 5.12: Valores de clasificación errónea

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
SVDD	0.0554	0.0811	0.0865
GA-SVDD	0.0520	0.0450	0.1017
BPSO-SVDD	0.0433	0.0394	0.0438
GBPSO-MDDM	0.0135	0.0179	0.0263
GBPSO-SVDD	0.0135	0.0135	0.0135

muestra los valores de clasificación errónea alcanzados por cada algoritmo, mientras que las figuras 5.2, 5.3 y 5.4 muestran el correspondiente comportamiento de convergencia para las distintas configuraciones antes descritas.

Como se puede ver en la tabla 5.2, SVDD sin selección de características y GA-SVDD presentan un menor rendimiento para la detección de fallas que los algoritmos BPSO-SVDD, GBPSO-SVDD y GBPSO-MDDM. Aunque BPSO converge más rápido, este converge a un óptimo local cuyo valor es mayor que el alcanzado por GBPSO (ver figura 5.2). En este caso en particular SVDD y MDDM presentan errores en la clasificación muy similares cuando son combinados con GBPSO, siendo SVDD el que presenta mejores resultados.

Las variables e hiperparámetros seleccionados por estos algoritmos a través de la solución del problema de optimización combinatoria, así como los tiempos de computo se presentan en las tablas 5.13, 5.14 y 5.15 respectivamente.

Finalmente para la fase de monitoreo y prueba 50 nuevas observaciones del subconjunto

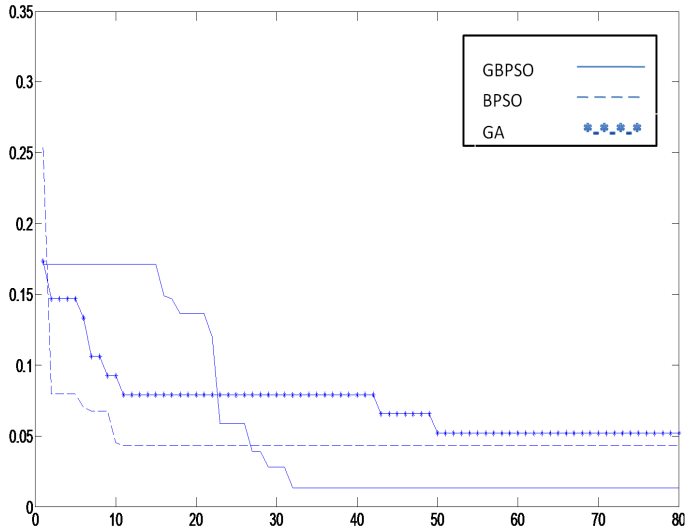

Figura 5.2: Convergencia de GBPSO, BPSO and GA para caso 1).

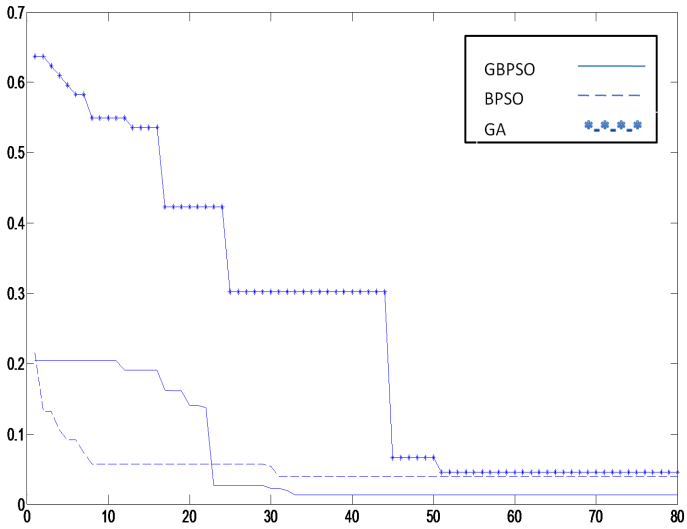

Figura 5.3: Convergencia de GBPSO, BPSO and GA para caso 2).

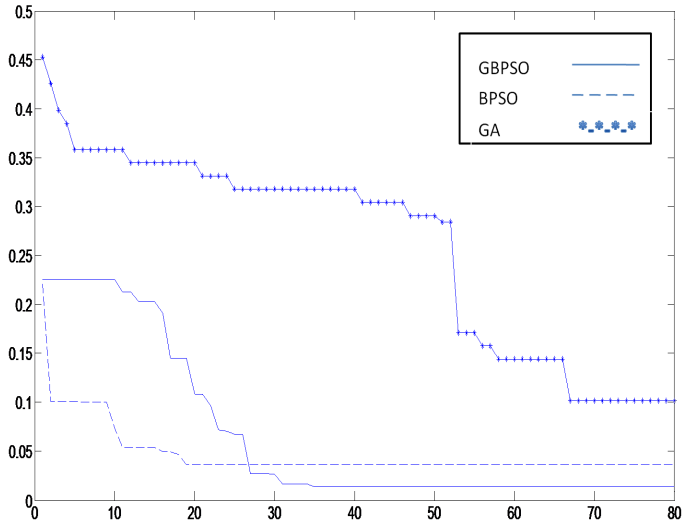

Figura 5.4: Convergencia de GBPSO, BPSO and GA para caso 3).

Tabla 5.13: Variables seleccionadas.

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GA	x_f^{21}, x_f^{24}	x_f^{18}, x_f^{28}	x_f^{10}, x_f^{24}
BPSO	x_f^{22}, x_f^{36}	x_f^{22}, x_f^{27}	x_f^{36}
GBPSO	x_f^{20}	x_f^{20}	x_f^{20}

Tabla 5.14: Hiperparámetros seleccionados.

Algoritmo		$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GA	C	0.01	0.1	1
	s	450	1	1
BPSO	C	0.0001	0.001	0.1
	s	10	50	100
GBPSO	C	0.0001	0.0001	0.0001
	s	1	1	1

de variables seleccionadas (x_f^{20} en este caso) son analizadas para la detección de fallas. Se encontró que observaciones 21, 22, 23 se encuentran por encima del valor permitido, que en este caso es el valor R^2. Lo cual significa que el sistema ha detectado una falla en

Tabla 5.15: Tiempo computacional en fase de entrenamiento (segundos).

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GA	995.10	980.90	975.80
BPSO	1303.10	1322.90	1388.10
GBPSO	1211.00	1290.20	1300.70

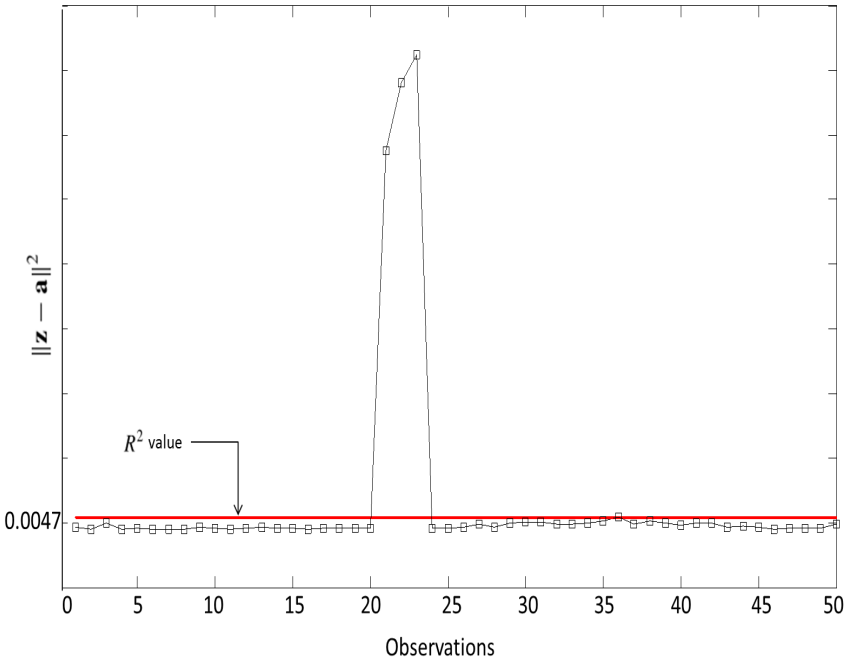

Figura 5.5: Gráfico de detección de fallas (caso 1)

dichas corridas de producción (ver figura 5.5). De acuerdo con los expertos del proceso, el sistema tuvo un valor de clasificación errónea de 0. Una vez que las muestras defectuosas se identifican, y además se conoce el subconjunto de variables que contribuyen a los defectos de calidad, es posible llevar a cabo una mejora efectiva del proceso para evitar quejas de los clientes.

5.2.2. Maquinado de cabeza de motor

La metodología propuesta se aplica para llevar a cabo el control de calidad dimensional en el proceso de maquinado de una cabeza de motor. Dicho proceso involucra 12 características, y el conjunto de datos consta de 189 observaciones en operación normal y 16 observaciones en modo de falla. La tabla 5.16 muestra las variables consideradas, la descripción de las mismas se presenta tal y como se maneja en el interior de la planta manufacturera. Por motivos de confidencialidad no se presentan más detalles.

Tabla 5.16: Descripción de variables cabeza de motor.

	Descripción
x_f^1	187-VERTICAL 6MM HOLE 506R INT SIDE
x_f^2	188-TRANSVERSAL 6MM HOLE 506R INTSIDE
x_f^3	191-VERTICAL 6MM HOLE 507R EXH SIDE
x_f^4	192-TRANSVERSAL 6MM HOLE 507R EXH SIDE
x_f^5	229-LONG.HOLE 124L CAM.L.ADM. REAR 7MM HOLE
x_f^6	230-TRANS.HOLE 124L CAM.L.ADM. REAR 7MM HOLE
x_f^7	205-LONG.HOLE 124L CAM.L.ADM. REAR 4MM HOLE
x_f^8	206-TRANS.HOLE 124L CAM.L.ADM. REAR 4MM HOLE
x_f^9	235-LONG.HOLE 123L CAM.L.ADM.MIDDLE 7MM HOLE
x_f^{10}	236-TRANS .HOLE 123L CAM.L.ADM. MIDDLE 7MM HOLE
x_f^{11}	213-LONG.HOLE 123L CAM.L.ADM.MIDDLE 4MM HOLE
x_f^{12}	214-TRANS .HOLE 123L CAM.L.ADM. MIDDLE 4MM HOLE

Resultados numéricos

En este caso de estudio, los parámetros se ajustan de la misma manera que el caso anterior, así mismo se consideran los mismos tres escenarios antes descritos, es decir: 1) $a_1 = 0{,}05$, $a_2 = 0{,}95$, 2) $a_1 = 0{,}95$, $a_2 = 0{,}05$ y 3) $a_1 = a_2 = 0{,}5$. Primero se realiza un análisis de correlación el cual muestra que existen cuatro grupos de variables altamente correlacionadas. Estos grupos se presentan en la tabla 5.17.

Debido al desempeño superior de GBPSO sobre otros algoritmos como BPSO y GA mostrado en el caso anterior, para este caso de estudio se utilizará únicamente GBPSO para resolver el problema de optimización combinatoria para la selección de características. Por otro lado, al contar con variables altamente correlacionadas se usara SVDD y MDDM para

Tabla 5.17: Variables correlacionadas

Variables correlacionadas
x_f^1, x_f^3
x_f^5, x_f^7, x_f^9, x_f^{11}
x_f^6, x_f^8
x_f^{10}, x_f^{12}

llevar a cabo la tarea de clasificación. Debido a que las características medidas en este estudio son de naturaleza dimensional, la etapa de limpieza de señales multivariada se omite. En la etapa de aprendizaje se aplica SVDD y MDDM para el cálculo de la clasificación errónea total. Al igual que el caso de estudio anterior se utiliza validación cruzada de 10 iteraciones para la validación del sistema. La Tabla 5.18 muestra los valores de clasificación errónea alcanzados por cada algoritmo mientras que las Figuras 5.6-5.8 muestran el correspondiente comportamiento de convergencia para las distintas configuraciones antes descritas.

Tabla 5.18: Valores de clasificación errónea

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GBPSO-SVDD	0.0916	0.0833	0.1294
GBPSO-MDDM	0.1793	0.1147	0.1848

Tabla 5.19: Variables seleccionadas.

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GBPSO-SVDD	x_f^1, x_f^{12}	x_f^{12}	x_f^3
GBPSO-MDDM	x_f^1, x_f^{12}	x_f^3	x_f^3

Finalmente para la fase de monitoreo y prueba 33 nuevas observaciones del subconjunto de variables seleccionadas (x_f^3 en este caso) son analizadas para la detección de fallas. Se encontró que observaciones 18, 22, 23 se encuentran por encima del valor permitido, que en este caso es el valor R^2. Lo cual significa que el sistema ha detectado una falla en dichas corridas de producción (Ver Figura 5.9). Sin embargo se informó por parte de los expertos del proceso que la muestra 18 pertenece a una muestra en operación normal

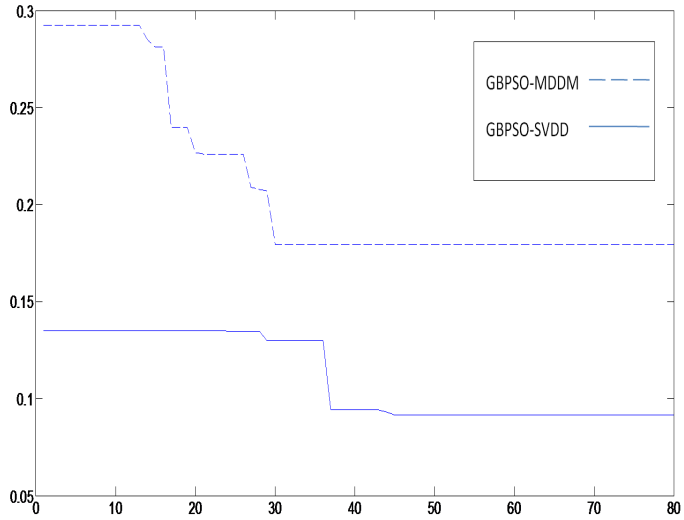

Figura 5.6: Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 1).

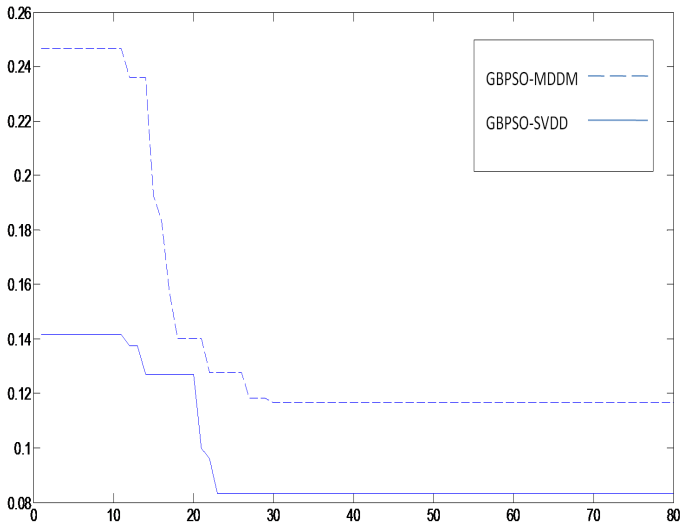

Figura 5.7: Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 2).

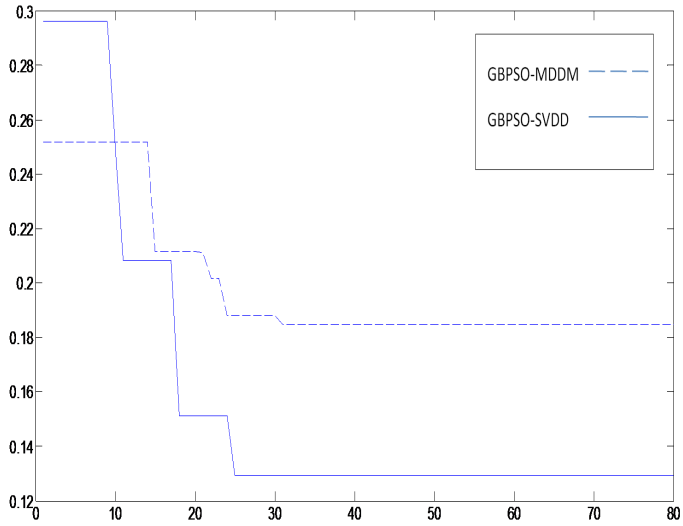

Figura 5.8: Convergencia de GBPSO-SVDD y GBPSO-MDMD para caso 3).

Tabla 5.20: Hiperparámetros seleccionados.

Algoritmo		$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
GBPSO-SVDD	C	0.0001	0.001	0.0001
	s	1	200	1
GBPSO-MDDM	C	0.001	0.001	0.0001
	s	1	1	1

Tabla 5.21: Tiempo computacional en fase de entrenamiento (segundos).

Algoritmo	$a_1 = 0{,}05$ $a_2 = 0{,}95$	$a_1 = 0{,}95$ $a_2 = 0{,}05$	$a_1 = 0{,}5$ $a_2 = 0{,}5$
BPSO	1303.10	1322.90	1388.10
GBPSO	1211.00	1290.20	1300.70

mientras que la muestra 21 pertenece a una muestra en falla. Por lo que el sistema tuvo 2 errores. Usando la ecuación 4.2, la clasificación errónea es de 0.1833.

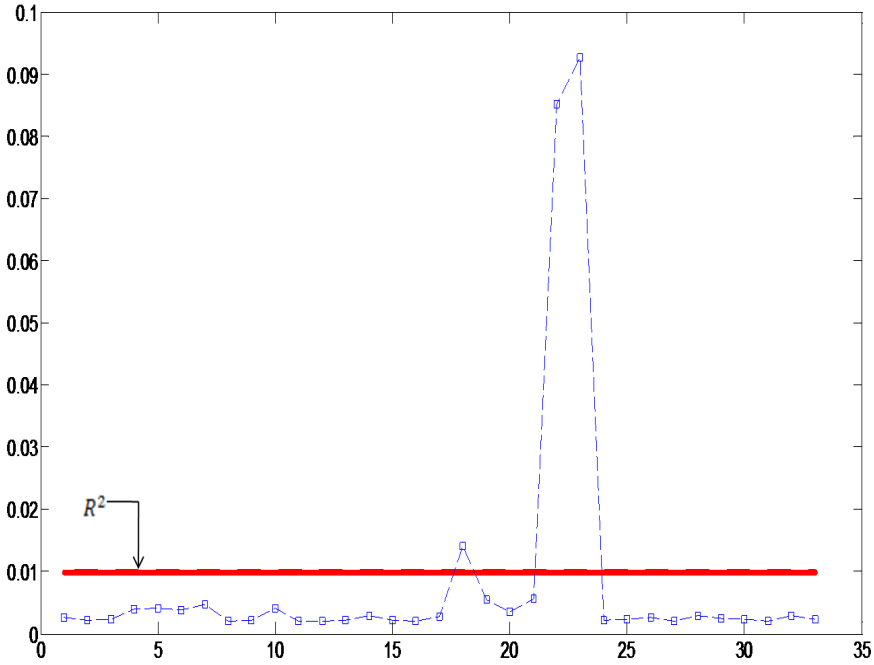

Figura 5.9: Gráfico de detección de fallas (caso 2)

6

Conclusiones y trabajo futuro

En este trabajo de investigación se propuso una metodología para la detección de fallas en procesos multivariados la cual se compone de tres etapas: 1) pre procesamiento, 2) aprendizaje y 3) monitoreo y prueba. Dicha metodología realiza en su primera fase limpieza de señales mediante limpieza multivariada Wavelet en caso de que el proceso así lo requiera. Posteriormente en la etapa de aprendizaje se combina PSO binario Gompertz (GBPSO) y Support vector Data Description (SVDD) para llevar a cabo la optimización del espacio de referencia (ajuste de hiperparámetros para SVDD) y la selección de variables con el fin de mejorar la detección de fallas, es decir reducir la clasificación erronea. Por lo tanto los objetivos son la maximización del rendimiento del modelo de clasificación y la minimización del número de variables utilizadas. Finalmente la etapa de monitoreo y prueba lleva a cabo la detección de falla en muestras nuevas recopiladas del proceso bajo análisis.

La metodología propuesta en este trabajo se implementó en cuatro distintos casos, de los cuales los dos primeros son problemas *benchmarks*. En el primero, conocido como conjunto de datos iris de Fisher, se encontró mediante análisis de correlación que la variable largo de pétalo esta altamente correlacionada con otras dos de las tres variables restantes. En la etapa de aprendizaje se uso la combinación de GBPSO con los clasificadores SVDD y MDDM (modelo de SVDD construido en base a la distancia Mahalanobis). Tanto GBPSO-SVDD y GBPSO-MDDM seleccionaron la variable largo de pétalo como la variable relevante. Es decir, dicha variable permite reducir la clasificación errónea la cual fue de 0.2188 y de 0.1250 para SVDD y MDDM respectivamente. Debido a la reducida

dimensionalidad de este conjunto de datos es fácil observar que para este caso en particular la variable seleccionada por la metodología propuesta es la que tiene una fuerte correlación con las otras restantes, por lo tanto dicha característica es considerada como la de mayor relevancia para llevar a cabo con éxito el reconocimiento de la especie iris setosa.

En el segundo de los problemas *benchmarks* el objetivo es el diagnosticar un cáncer como maligno o benigno a partir de un conjunto de datos conocido como: base de datos cáncer de mama. Los resultados obtenidos por la metodología propuesta se comparan con el estudio realizado por (Glander, 2017) donde se muestra que las variables x_f^2 y x_f^3 (uniformidad del tamaño de la célula y uniformidad de la forma celular respectivamente) presentan alta correlación con el resto de las variables. Posteriormente llevan a cabo un análisis de importancia de variables el cual da como resultado que las variables x_f^2 y x_f^3 son las de mayor importancia. Al implementar la metodología propuesta se obtiene como resultado que GBPSO-SVDD selecciona la variable x_f^2 con un valor de clasificación errónea de 0.0993, mientras que GBPSO-MDDM selecciona la variable x_f^3 con un valor de clasificación errónea de 0.0556. Se observa que los resultados de selección de características obtenidos por la metodología propuesta coinciden en gran medida con el estudio de referencia dando validez a este trabajo de investigación. De acuerdo a las conclusiones presentadas por (Glander, 2017), algoritmos genéticos (GA) es el algoritmo que le permitió en conjunto con random forest alcanzar los mejores resultados en clasificación. Por tal motivo en el siguiente caso de estudio GA es considerado para resolver el problema de la selección de variables.

La metodología propuesta es implementada en dos procesos multivariados reales. El primero corresponde a un proceso de moldeo por inyección de plástico de pedales de coche en una compañía automotriz de la región el cual consta de 36 características. Debido a que gran parte de estas características son de naturaleza eléctrica, se lleva a cabo una limpieza de señales wavelet multivariada dando como resultado señales mas suaves lo cual facilita el proceso de clasificación de la metodología. Por lo tanto la limpieza de señales usando wavelets es idónea para aplicarse en la etapa de pre procesamiento, respondiendo así la primera pregunta de investigación planteada en este trabajo. En la etapa de aprendizaje se implementaron distintos enfoques como GA-SVDD, BPSO-SVDD, GBPSO-SVDD y GBPSO-MDDM. Así como SVDD estándar, es decir, realizar la detección de fallas sin llevar a cabo selección de características con el fin de validar la ventaja que ofrece dicho

procedimiento. Los resultados muestran que cualquiera de los enfoques que realizan selección de características presentan una menor clasificación errónea en relación con SVDD estandar. Mientras que GA y BPSO presentan un menor desempeño que GBPSO. Los enfoques SVDD y MDDM para la clasificación presentan un desempeño muy similar siendo ligeramente mejor SVDD. En los proceso reales estudiados se llevo a cabo la etapa de monitoreo de nuevas observaciones, dando como resultado un valor de 0 clasificación errónea por parte del sistema para un nuevo conjunto de datos probados.

El segundo caso real analizado se trata de un proceso de maquinado de una cabeza de motor. Dicho proceso involucra 12 características, y el conjunto de datos consta de 189 observaciones en operación normal y 16 observaciones en modo de falla. Debido al desempeño superior de GBPSO sobre otros algoritmos como BPSO y GA mostrado en el caso anterior, para este caso de estudio se utiliza únicamente GBPSO para resolver el problema de optimización combinatoria para la selección de características. Al contar con variables altamente correlacionadas se usara SVDD y MDDM para llevar a cabo la tarea de clasificación. Donde SVDD obtuvo una clasificación errónea menor que MDDM, ambos seleccionando la variable x_f^3. En la fase de monitoreo y prueba 33 nuevas observaciones del subconjunto de variables seleccionadas (x_f^3 en este caso) son analizadas para la detección de fallas donde el sistema presentó una clasificación errónea de 0.1833 el cual es un valor cercano al obtenido en la fase de aprendizaje.

Se encontró que en tres de los cuatro casos de estudio donde se implementó la metodología propuesta, SVDD presentó un mejor desempeño que el modelo construido con la distancia Mahalanobis (MDDM) para la clasificación independientemente de las características propias de los distintos conjuntos de datos. Esto comprueba que los modelos basados en vectores de soporte no necesitan de supuestos acerca de los datos para resolver problemas como se afirma en distintos trabajos de investigación, por ejemplo en (Campos-Espinoza, 2012). Donde además se establece que los problemas causados por la colinealidad entre las variables y la falta de normalidad en los datos, no tienen sentido en este tipo de técnica, respondiendo así la segunda pregunta de investigación. Por otro lado se encontró que el algoritmo GBPSO fue más eficiente para resolver el problema de optimización planteado en este trabajo por encima de otros como BPSO y GA, respondiendo así la tercera pregunta de investigación. Además, como se mostró anteriormente, los valores de clasificación errónea obtenidos en la fase de entrenamiento usando validación cruzada de 10 iteraciones, son cercanos a los valores que se obtienen al monitorear nuevas observaciones.

Por lo tanto la validación cruzada de 10 iteraciones es un método adecuado para evaluar el desempeño de la metodología propuesta para la detección de fallas. Respondiendo así la cuarta pregunta de investigación planteada en este trabajo.

Debido a los valores de clasificación errónea obtenidos por parte de SVDD en las fases de entrenamiento y monitoreo además del comportamiento en la convergencia por parte de GBPSO se demuestra que la metodología propuesta es eficiente para la detección de fallos y la selección de variables relevantes con el fin de llevar a cabo una mejora efectiva en procesos de manufactura. Con lo cual se resuelve de manera satisfactoria el problema general y los problemas específicos establecidos, además de verificar la hipótesis formulada en este trabajo de investigación.

Este trabajo de investigación abordó problemas de clasificación conocidos como clasificación de una clase y clasificación de dos clases. Por lo tanto, en trabajos futuros la metodología se extenderá para lidiar con problemas de multi clasificación y de esta manera detectar diferentes tipos específicos de fallas presentes en procesos industriales. Por otro lado, debido al desempeño similar en la clasificación mostrado por MDDM con respecto al algoritmo SVDD utilizado en este trabajo, el cual incorpora los llamados ejemplos negativos para su etapa de entrenamiento. Es de interés desarrollar en trabajos futuros una versión de MDDM que incorpore ejemplos negativos para la construcción del modelo de predicción.

Otra área de investigación para hacer más robusta esta metodología es disminuir el tiempo de entrenamiento para los clasificadores mediante la sustitución del algoritmo de punto interior para resolver el problema de optimización cuadrática por un método analítico como optimización secuencial mínima (SMO) propuesto por **John Platt (1998)** en (Platt, 1998) para el algoritmo estándar SVM y adaptarlo para los modelos de SVDD y MDDM.

Referencias

Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O., y Ferreira-Santiago, A. (2013). Feature selection using associative memory paradigm and parallel computing. *Computación y Sistemas*, *17*, 41-52.

Al-jonid, K., Jiayang, W., y Mohammed, N. (2013, May). A new fault classification model for prognosis and diagnosis in cnc machine. En *2013 25th chinese control and decision conference (ccdc)* (p. 3538-3543). doi: 10.1109/CCDC.2013.6561561

Altman, A., y Gondzio, J. (1999). Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization.

Aminghafari, M., Cheze, N., y Poggi, J.-M. (2006). Multivariate denoising using wavelets and principal component analysis. *Computational Statistics & Data Analysis*.

Andre, A. B., Beltrame, E., y Wainer, J. (2013). A combination of support vector machine and k-nearest neighbors for machine fault detection. *Applied Artificial Intelligence*, *Vol. 27*, 36-49.

Badii, M., Guillen, A., Lugo-Serrato, O., y Aguilar-Garnica, J. (2014). Non-parametric correlation and its application in scientific research.

Barrera, G., y Carreón, G. (2005). Control de calidad de piezas cerámicas por medio de análisis de vibraciones y redes neuronales artificiales. *Computación y sistemas*, *8*, 187-195.

Benkedjouh, T., Medjaher, K., Zerhouni, N., y Rechak, S. (2012). Fault prognostic of bearings by using support vector data description. *In Prognostics and Health Management (PHM), IEEE Conference*, 1-7.

Berenji, H. R., y Wang, Y. (2006). Wavelet neural networks for fault diagnosis and prognosis. *IEEE International Conference on Fuzzy Systems*, 1334-1339.

Bhajantri, L. B., y Nalni, N. (2014). Adaptive in natural and artificial systems. *I.J. Computer Network and Information Security*, *12*, 37-46.

Bolón-Canedo, V., Sánchez-Maroño, N., y Alonso-Betanzos, A. (2015). *Feature selection for high-dimensional data*.

Bounsiar, A., y Madden, M. G. (2014). Kernels for one-class support vector machines. *Information Science and Applications (ICISA). IEEE*, 1-4.

Campos-Espinoza, R. A. (2012). *Técnicas de sistemas automaticos de soporte vectorial en la réplica del rating crediticio* (Tesis Doctoral no publicada). ESADE.

Chandrashekar, G., y Sahin, F. (2014). A survey on feature selection methods. *Computers*

and Electrical Engineering, *40*, 16-28.

Chauhan, P., Pant, M., y Deep, K. (2012). Novel binary pso for continuous global optimization problems. *Procedings of the international conference of soft computing for problem solving*, 169-176.

Chiang, L. H., Russell, E. L., y Braatz, R. D. (2001). *Fault detection and diagnosis in industrial systems* (P. M. J. Grimble y P. M. A. Johnson, Eds.). Springer-Verlag London Ltd.

Chun-Chin, H., Mu-Chen, C., y Long-Sheng, C. (2010). Intelligent ICA¨CSVM fault detector for non-gaussian multivariate process monitoring. *Elsevier*, *37*, 3264-3273.

Das, S., Abraham, A., y Konar, A. (2008). Particle swarm optimization and differential evolution algorithms. technical analysis, applications and hybridization perspectives. *Studies in Computational Intelligence*.

Daubechies, I. (1992). *Ten lectures on Wavelets*. Society for Industrial & Applied Mathematics.

Deng, N., Tian, Y., y Zhang, C. (2013). *Support vector machines: Optimization based theory, algorithms, and extensions*. Chapman.

Domínguez-Moreno, J. J. (2004). *Un algoritmo de puntos interiores para programación cuadrática* (Tesis de Master no publicada).

Elhariri, E., El-Bendary, N., Fouad, M. M. M., Platoš, J., Hassanien, A. E., y Hussein, A. M. M. (2014). Multi-class svm based classification approach for tomato ripeness. En A. Abraham, P. Krömer, y V. Snášel (Eds.), *Innovations in bio-inspired computing and applications: Proceedings of the 4th international conference on innovations in bio-inspired computing and applications, ibica 2013, august 22 -24, 2013 - ostrava, czech republic* (pp. 175–186). Cham: Springer International Publishing. Descargado de `https://doi.org/10.1007/978-3-319-01781-5_17` doi: 10.1007/978-3-319-01781-5_17

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems.

Frazier, M. W. (1999). *An introduction to wavelets through linear algebra* (K. R. S. Axler F.W. Gehring, Ed.). Springer.

Gaied, K. S., Ping, H. W., Uddin, M. N., Khalid, M., y Saghafina, A. (2012). Wavelet based prognosis for fault tolerant control of induction motor with stator and speed sensor faults. *IEEE*, 1-8.

Gamarra, M., y Quintero, C. (2013). Using genetic algorithm feature selection in neural classification systems for image pattern recognition. *Ingeniería e Investigación.*, *33*,

52.58.

Ganesh, M., Sutherland, H. A., y Worden, B. D. (2013). *Prognostics system and method for fault detection in electrical insulation* (n.º US20140049264A1).

García-Díaz, E. E. (2005). *Boosting support vector machines* (Tesis de Master no publicada). Universidad de los andes facultad de ingeniería, Bogotá Colombia.

Ghasemi, E., Aaghaie, A., y Cudney, E. A. (2015). Mahalanobis taguchi system: a review. *International Journal of Quality & Reliability Management*, *Vol. 32*.

Giaouris, D., Finch, J., Student, O. F., Kennel, R., y El-Murr, G. (2008). Wavelet denoising for electric drives. *IEEE Transactions on Industrial Electronics*.

Gill, P. E., y Wong, E. (2015). Methods for convex and general quadratic programming. *Math. Prog. Comp.*.

Glander, S. (2017). *Feature selection in machine learning (breast cancer datasets).* Descargado de `https://shiring.github.io/machine_learning/2017/01/15/rfe_ga_post`

Haixiang, G., Yijing, L., Yanan, L., Xiao, L., y Jinling, L. (2016). BPSO-adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*.

Hauser, K. (2012). *Lecture 10: Convex optimization and interior-point methods.*

Hernández, E. (1996). *A first course on wavelets.* CRC PRESS.

Holland, J. H. (1975). *Adaptive in natural and artificial systems.*

Hubbard, B. B. (1996). *The world according to wavelets.* A K Peters.

Iquebal, A. S., Pal, A., Ceglarek, D., y Tiwari, M. K. (2014). Enhancement of Mahalanobis-Taguchi system via rough sets based feature selection. *Expert Systems with Applications*, *vol. 41*, 8003-8015.

Jiang, H., Chen, J., Dong, G., Liu, T., y Chen, G. (2014). Study on hankel matrix-based svd and its application in rolling element bearing fault diagnosis. *Mechanical Systems and Signal Processing, ELSEVIER*, *52-53*, 338-359.

Jiang, S., y Benlian, X. (2010). Multivariate statistical process monitoring of propylene polymerization with principal component analysis and support vector data description. *Information Science and Engineering (ICISE), IEEE*, 1-4.

Karlis, D., Saporta, G., y Spinakis, A. (2003). A simple rule for the selection of principal components. *Comm. Statist. Theory Methods*.

Kennedy, J., y Eberhart, R. (1995). Particle swarm optimization. *International conference on neural networks, IEEE*.

Keskes, H., Brahama, A., y Lachiri, Z. (2013). Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet svm. *Electric Power Systems Research, Elsevier*, 151-157.

Kobayashi, M. (1998). *Wavelets and their applications.* Society for industrial and applied mathematics.

Kutylowska, M. (2014). Neural network approach for failure rate prediction. *Engineering Failure Analysis, ELSEVIER*, *47*, 41-48.

K. Yang, J. T. (2004). *Multivariate statistical methods in quality management.* McGraw-Hill Professional.

Le, T. M. (2012). *Multi-modal approach to support vector machine and support vector data description* (phdthesis).

Lee, G. M., Tam, N. N., y Yen, N. D. (2005). *Quadratic programming and affine variational inequalities a qualitative study.*

Lei, L., Xiao-dan, W., Xi, L., y Ya-fei, S. (2016). Hierarchical error-correcting output codes based on svdd. *Pattern Analysis and Applications*, *19*(1), 163–171. Descargado de `http://dx.doi.org/10.1007/s10044-015-0455-5` doi: 10.1007/s10044-015-0455-5

Lichman, M. (2013). *UCI machine learning repository.* Descargado de `http://archive.ics.uci.edu/ml`

Liu, M., y Zhang, D. (2016). Feature selection with effective distance. *Neurocomputing*, *215*, 100-109.

Luke, S. (2013). *Essentials of metaheuristics* (second ed.). Lulu. (Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/)

Ma, J., Wang, Q., y Dong, A. (2012). PCA-SVM based fault prognosis for flue gas turbine. *Second International Conference on Instrumentation & Measurement, Computer, Communication and Control*, 1304-1308.

Martis, R. J., y Acharya, U. R. (2013). ECG beat classification using PCA, LDA, ICA and discrete wavelet transform. *Biomedical Signal Processing and Control, Elsevier*, *Vol. 8*, 437-448.

Merino-Maestre, M. (2013). *Técnicas cálsicas de optimización. parte I: Programación lineal y no lineal.*

Misiti, M., Misiti, Y., Oppenheim, G., y Poggi, J.-M. (2007). *Wavelets and their applications.* ISTE Ltd.

Munakata, T. (2008). *Fundamentals of the new artificial intelligence* (D. Gries y F. B. Sh-

neider, Eds.). Springer.

M. Y. Abu, K. R. J., y Ramlie, F. (2014). Pattern recognition using mahalanobis-taguchi system on connecting rod through remanufacturing process: A case study. *Advanced Materials Research*, *Vol. 845*, 584-589.

Nason, G. P. (2008). *Wavelet methods in statistics whit R.* Springer.

Nemirovski, A. S., y Todd, M. J. (2009). Interior-point methods for optimization. *Cambridge University Press.*

Nguyen, T., Khosravi, A., Creighton, D., y Nahavandi, S. (2015). Classification of healthcare data using genetic fuzzy logic system and wavelets. *Expert Systems with Applications, ELSEVIER*, *42*, 2184-2197.

Nunes, I., Hernane, D., Andrade, R., Bartocci, L., y dos Reis, S. (2017). *Artificial neural networks: a practical course.*

Onar, S. Ç., Öztayşi, B., Kahraman, C., Yanık, S., y Şenvar, Ö. (2016). A literature survey on metaheuristics in production systems. En E.-G. Talbi, F. Yalaoui, y L. Amodeo (Eds.), *Metaheuristics for production systems* (pp. 1–24). Cham: Springer International Publishing. Descargado de `http://dx.doi.org/10.1007/978-3-319-23350-5_1` doi: 10.1007/978-3-319-23350-5_1

Osman, I., y Laporte, G. (1996). Metaheuristics: A bibliography. *Ann Oper Res.*

Palencia, G., Hing, R., y Rojas, M. (2008). Métodos de punto interior para optimización cuadrática convexa con matrices no definidas positivas.

p. Gu, Y., s. Cheng, L., y l. Chen, X. (2014, Aug). Optimization on reference space of mahalanobis-taguchi system based on hybrid encoding genetic algorithms. En *2014 international conference on management science engineering 21th annual conference proceedings* (p. 62-68). doi: 10.1109/ICMSE.2014.6930209

Platt, J. C. (1998). *Sequential minimal optimization: A fast algorithm for training support vector machines* (Inf. Téc.). ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING.

Qian, X., Liao, L.-Z., y Sun, J. (2017). Analysis of some interior point continuous trajectories for convex programming.

Radunoviç, D. (2009). *Wavelets from math to practice.* Springer.

Ren, J. (2012). ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Strathprints Institutional Repository*, *26*, 144-153.

Ren, J., Cai, Y., Xing, X., y Chen, J. (2011). A method of multi-class faults classi-

fication based-on mahalanobis-taguchi system using vibration signals. *Reliability, Maintainability and Safety (ICRMS),*, 1015-1020.

Reséndiz, E., Moncayo-Martínez, L., y is, G. S. (2013). Binary ant colony optimization applied to variable screening in the mahalanobis¨ctaguchi system. *Expert Systems with Applications, ELSEVIER*, *40*, 634-637.

Reséndiz-Flores, E., y López-Quintero, M. (2016). Optimal identification of impact variables in a welding process for automobile seats mechanism by mts-gbpso approach. *Int J Adv Manuf Technol*.

Reséndiz-Flores, E. O. (2001). *Interpolación, Waveletes y B-splines.*

Reséndiz-Flores, E. O., y Rull-Flores, C. A. (2013). Mahalanobis-Taguchi system applied to variable selection in automotive pedals components using gompertz binary particle swarm optimization. *Expert Systems with Applications*.

Richard, J., y Dean, W. (2007). *Applied multivariate statistical analysis.* Prentice Hall.

Rosas-Snell, A., y Zúñiga-Contreras, J. (2004). *Estadística descriptiva e inferencial I.*

Rull-FLores, C. A. (2013). *Aplicación de la metodología Mahalanobis-Taguchi a un proceso industrial utilizando optimización con enjambre de partículas.* (Tesis de Master no publicada).

Saha, S., y Rajasekaran, S. (2015). Novel randomized feature selection algorithms. *International Journal of Foundations of Computer Science*, *26*, 321-341.

Sajjad, S., y Zaidi, H. (2013). Diagnosis and prognosis of in-service electric machine in the absence of historic data related to faults and faults progression. *IEEE Xplore Digital Library*, 615-619.

Seebach, C. (2006). *Condiciones de karush kuhn tucker.*

Shen, C., Wang, D., Kong, F., y Tse, P. W. (2013). Fault diagnosis of rotating machinery based on the statistical parameters of wavelet packet paving and a generic support vector regressive classifier. *ELSEVIER*, *46*, 1551-1564.

Siarry, P. (2016). Introduction. En *Metaheuristics.*

Stańczyk, U., y Jain, L. C. (2015). *Feature selection for data and pattern recognition.*

Stewart, A. X., Nuthmannb, A., y Sanguinettic, G. (2014). Single-trial classification of eeg in a visual object task using ica and machine learning. *Journal of Neuroscience Methods*, *228*, 1-14.

Su, C.-T., y Hsiao, Y.-H. (2009). Multiclass mts for simultaneous feature selection and classification. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, *Vol. 21*, 192-205.

Taha, H. A. (2012). *Investigación de operaciones.*

Tax, D. M., y Duin, R. P. (2004). Support vector data description. *Machine Learning, Kluwer Academic Publishers..*

Tingting Mu, A. K. N. (2009). Multiclass classification based on extended support vector data description. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, *39*, 1206-1216.

Torrence, C., y Compo, G. P. (1998). A practical guide to Wavelet analysis. *American Meteorological Society*, *79*, 61-78.

Vanderbei, R. J., y Carpenter, T. J. (1993). Symmetric indefinite systems for interior point methods.

Vapnik, V. (2000). *The nature of statistical learning theory.* Springer.

Walker, J. S. (2008). *A primer on wavelets and their scientific applications.* Chapman & Hall.

Wang, D., y Tan, X. (2013). *Neural information processing* (Z.-G. H. R. M. K. Minho Lee Akira Hirose, Ed.). Springer.

Wei, X., Li, Y., Liu, D., y Zhan, L. (2010). Mahalanobis support vector machines made fast and robust. En *New advances in machine learning.*

Wenliao, D., Zhiqiang, G., Liangwen, W., Ansheng, L., y Zhiyang, W. (2014, June). Intelligent fault diagnosis of plunger pump in truck crane based on a hybrid fault diagnosis scheme. En *Proceeding of the 11th world congress on intelligent control and automation* (p. 5361-5365). doi: 10.1109/WCICA.2014.7053629

Xiaogang, D., y Xuemin, T. (2013). Nonlinear process monitoring using dynamic kernel slow feature analysis and support vector data description. *Control and Decision Conference (CCDC).*

Xie, G., Jiang, Y., y Chen, N. (2013). A multi-class support vector data description approach for classification of medical image. *Ninth International Conference on Computational Intelligence and Security*, 115-119.

Yacout, S., Salamanca, D., y Mortada, M.-A. (2013). *Tool and method for fault detection of devices by condition based maintenance* (n.º US20130132001A1).

Yang, L., Ma, W.-M., y Tian, B. (2011). *Advances in neural networks ¨c isnn 2011* (H. H. Derong Liu Huaguang Zhang Marios Polycarpou CesareAlippi, Ed.). Springer.

Yan Xuesong, Z. Z., Liu Hanmin, y Qinghua, W. (2016). Hybrid genetic algorithm for engineering design problems. *Journal of Computational and Theoretical Nanoscience*,

13, 6312-6319.

Yin, G., Zhang, Y.-T., Li, Z.-N., Ren, G.-Q., y Fan, H.-B. (2014). Online fault diagnosis method based on incremental supportvector data description and extreme learning machine with incremental output structure. *Neurocomputing, ELSEVIER*, *128*, 224-231.

Zhao, Q., Okada, K., Rosenbaum, K., Kehoe, L., Zand, D. J., Sze, R., ... Linguraru, M. G. (2014). Digital facial dysmorphology for genetic screening: Hierarchical constrained local model using ICA. *Medical Image Analysis, Elsevier*, *18*, 699-710.