



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Detección de daño en edificios con técnicas de Minería de Datos

Tesis que presenta

Salvador Abdelaziz Villegas Solís

Para obtener el grado de

Maestro en Ciencias

en Computación

Directores de Tesis

Dra. Xiaou Li Zhang

Dr. Wen Yu Liu

México, D. F.

Noviembre, 2014

Agradecimientos

Primeramente, quiero agradecer a mi familia, quien son el pilar de mi ser y quienes me brindan su apoyo incondicional para seguir adelante; siempre estando en todo momento cuando los necesito. Sin ellos no hubiera sido capaz de ser quien ahora soy, en especial a mis abuelos, a mis padres, mis hermanos y mis tíos, ¡gracias!.

Quiero agradecer a la Dra. Xiaoou Li y al Dr. Wen Yu; quienes me brindaron su tiempo y su asesoramiento para culminar este trabajo de tesis. Así mismo, quiero agradecer al Dr. Amilcar y al Dr. Iván, por ser mis revisores de tesis, y enmarcar las correcciones necesarias en mi trabajo. Además, quiero agradecer al Dr. José, quien me abrió las puertas de su equipo de trabajo para enriquecer mi conocimiento y poder terminar mi trabajo. No olvidando al personal administrativo, quienes me brindaron su amistad y apoyo en los trámites que necesitaba (Sofía, Erika y Felipa).

A mis amigos, con los cuales siempre paso momentos agradables, y que también me han brindado sus consejos para seguir adelante.

Finalmente, agradezco al CONACYT por brindarme una beca de maestría, y al Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV), por brindarme un lugar de encuentro con la ciencia y el conocimiento que tanto me apasionan.

Resumen

El monitoreo del estado estructural (**Structural Health Monitoring** o *SHM* por sus siglas en inglés) es un área de la ingeniería de estructuras, la cual se encarga del estudio e implementación de procedimientos para la detección de daño a través de la medición de las propiedades dinámicas de dichas estructuras. Dicho lo anterior, se pueden utilizar distintas herramientas para el monitoreo; desde inspecciones visuales, hasta llegar a técnicas más avanzadas, v.g. modelos matemáticos o análisis de datos experimentales.

Por tanto, este trabajo de tesis consistió en desarrollar una metodología capaz de detectar daños en un edificio provocados por vibraciones, con técnicas de *Minería de Datos*. El enfoque utilizado para desarrollar la metodología se basó en la obtención de datos experimentales, i.e. debido a que los modelos matemáticos consideran casos ideales, dejando de lado algunos factores que inciden en la estructura, como: viento, presión, humedad, etc. Entre las principales técnicas utilizadas, destacan: las Máquinas de Soporte Vectorial y los Árboles de Decisión, las cuales fueron implementadas y mejoradas, adecuándolas a nuestro problema. Así mismo, con el fin de respaldar los resultados obtenidos de las Máquinas de Soporte Vectorial y los Árboles de Decisión, se realizaron distintos análisis estadísticos.

Para la generación y recolección de datos, se utilizó un prototipo de laboratorio a escala de un edificio, así como un generador de ondas de vibración. Estos dispositivos están equipados con los aditamentos esenciales para llevar a cabo los experimentos requeridos en el proceso del SHM.

Al final de este trabajo de tesis, los resultados demostraron que ambas técnicas propuestas, son capaces de llevar a cabo el proceso del SHM con alta precisión; para la clasificación, localización y severidad de daño del edificio.

Abstract

Structural Health Monitoring (*SHM*), is a research area derived from structural engineering, which deals with the study and implementation of methods for damage detection in structures, using the records of its dynamic properties. Therefore, different tools are necessary for SHM process; ranging from visual inspections to sophisticated techniques, v.g. based on mathematical models or based on experimental data.

Therefore, one of main objectives of this thesis, was to develop a methodology capable of detecting structural damage in a building caused by vibration, by applying *Data Mining* techniques on experimental data. Two main techniques were used: Support Vector Machines and Decision Trees, which were implemented and improved adapting them to our problem. Likewise, to support the results of experiments, statistical analyses were performed, in order to define the modal behavior of the structure (i.e. the comparison of the structure from a state with damage vs. a state without damage).

For generation and collection of experimental data, a building laboratory prototype and a vibration wave generator were used. These devices are equipped with essential hardware and software attachment, to carry out the experiments required by the process of building damage detection.

At the end of this thesis work, results showed that both proposed techniques are capable of carrying out the process of SHM with high precision; for classification, location and severity of damage from building.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Índice de figuras	XIII
Índice de tablas	XVII
1. Introducción	1
1.1. Motivación	3
1.2. Planteamiento del problema	3
1.3. Objetivos de la tesis	4
1.3.1. General	4
1.3.2. Específicos	4
1.4. Organización de la tesis	5
2. Antecedentes y Estado del Arte	7
2.1. Detección de daños en edificios basado en <i>SHM</i>	7
2.1.1. Causas que provocan el daño en edificios	8
2.1.2. Clasificación del daño en estructuras	10
2.1.3. Paradigma de reconocimiento del daño basado en el <i>SHM</i>	11
2.1.4. Utilización de sensores para el monitoreo de estructuras	13
2.1.5. Axiomas dentro de la detección de daño	14
2.2. Minería de Datos	15
2.2.1. División de la Minería de Datos	15
2.2.2. Proceso para la detección del daño con el enfoque <i>KDD</i>	16
2.2.3. Análisis <i>K-NN</i>	17
2.2.4. Análisis de Redes Neuronales	20
2.2.5. Análisis de Clusters	24
2.3. Métodos para la detección de daños en estructuras	25
2.3.1. Detección de daño con Análisis de Clusters	25
2.3.2. Detección de daño con Bosques Aleatorios	26

2.3.3.	Otros métodos basados en Minería de Datos	26
2.3.4.	Métodos basados en el cambio de las frecuencias	27
2.3.5.	Métodos basados en análisis estadístico	27
2.3.6.	Métodos basados en sus características dinámicas	28
2.4.	Conclusión	29
3.	Sistema para la detección de daños	31
3.1.	Descripción del sistema	31
3.1.1.	Edificio	31
3.1.2.	Sistema para la adquisición de los datos	33
3.1.3.	Software	35
3.2.	Pruebas experimentales	35
3.3.	Análisis estadístico	39
3.3.1.	Momentos estadísticos	39
3.3.2.	Distribuciones de Probabilidad	44
3.4.	Pre-procesamiento de los datos	48
3.5.	Conclusión	52
4.	Máquinas de Soporte Vectorial para la detección de daños	53
4.1.	Clasificación del daño	53
4.2.	Máquinas de Soporte Vectorial	55
4.2.1.	Problema de clasificación entre dos clases (Con daño y Sin daño)	56
4.2.2.	Hiperplano óptimo de separación entre clases C_1 y C_2	56
4.2.3.	Condiciones Karush-Kuhn-Tucker	61
4.2.4.	Hiperplanos de margen blando para casos linealmente no separables de C_1 y C_2	62
4.2.5.	Funciones Kernel para el problema de detección de daño	64
4.2.6.	Condición de Mercer para los kernels	67
4.3.	Métricas de evaluación para la clasificación de las clases C_1 y C_2	68
4.4.	Resultados de los experimentos	69
4.5.	Conclusión	79
5.	Árboles de Decisión para la detección de daños	81
5.1.	Árboles de Decisión para la localización y severidad del daño	81
5.1.1.	Componentes del Árbol de Decisión	82
5.1.2.	Construcción de Árboles de Decisión para el proceso de detección de daños	83
5.1.3.	Atributos para la construcción de Árboles de Decisión	87
5.1.4.	Medidas para seleccionar la mejor división	91
5.1.5.	Ganancia de información	94
5.2.	Resultados de los experimentos	95
5.3.	Conclusión	106
6.	Conclusiones y Trabajo futuro	107

ÍNDICE GENERAL

XI

6.1. Conclusiones	107
6.2. Trabajo a futuro	109

Índice de figuras

1.1. Objetivos principales del SHM	2
1.2. Esquema general del trabajo de tesis	6
2.1. Distintas estructuras alrededor del mundo	8
2.2. Diferentes estructuras dañadas, 2.2a daños estructurales de un edificio ¹ , 2.2b sección de un puente dañado ² y 2.2c una turbina de viento con el aspa dañada ³	9
2.3. Clasificación del daño por niveles	11
2.4. Proceso para la detección de daño	16
2.5. Clasificación de un nuevo objeto con K - NN	20
2.6. Neurona biológica	21
2.7. Red Neuronal biológica	21
2.8. Perceptrón (Neurona artificial)	23
2.9. ANN con retro-propagación como algoritmo de entrenamiento	24
2.10. Resultado de uno de los experimentos con Análisis de Clusters	25
3.1. Estructura de prueba para la detección de daños	32
3.2. Dimensiones principales del edificio	32
3.3. Generador de vibraciones y la Base movable	34
3.4. Esquema general del sistema	34
3.5. Aceleraciones para ambos estados de los tres acelerómetros del escenario 1	37
3.6. Aceleraciones para ambos estados de los tres acelerómetros del escenario 2	38
3.7. Medias para A1, A2 y A3 del escenario 1	41
3.8. Desviación estándar de A1, A2 y A3 del escenario 1	41
3.9. Comparación de la media y la desviación estándar del escenario 1, para A2	42
3.10. Medias para A1, A2 y A3 del escenario 2	42
3.11. Desviación estándar de A1, A2 y A3 del escenario 2	42
3.12. Comparación de la media y la desviación estándar del escenario 2, para A2	43
3.13. Función Kernel para A2 del escenario 1	45
3.14. Función de probabilidad para A2 del escenario 1	46
3.15. Probabilidad acumulativa para A2 del escenario 1	46
3.16. Función Kernel para A2 del escenario 2	47
3.17. Función de probabilidad para A2 del escenario 2	47

3.18. Probabilidad acumulativa para A2 del escenario 2	47
3.19. Transformada Rápida de Fourier para A1, A2 y A3, estado 1 del escenario 1	50
3.20. Transformada Rápida de Fourier para A1, A2 y A3, estado 2 del escenario 1	50
3.21. Transformada Rápida de Fourier para A2 con daño y A2 sin daño del escenario 1	51
3.22. Transformada Rápida de Fourier para A1, A2 y A3, estado 1 del escenario 2	51
3.23. Transformada Rápida de Fourier para A1, A2 y A3, estado 2 del escenario 2	51
3.24. Transformada Rápida de Fourier para A2 con daño y A2 sin daño del escenario 2	52
4.1. Enfoque general para la clasificación del daño con Máquinas de Soporte Vectorial	54
4.2. Diagrama de la construcción del modelo de clasificación para la detección de daños	55
4.3. Clasificación del daño y sus posibles hiperplanos	56
4.4. Posibles hiperplanos canónicos	58
4.5. SVM con el hiperplano óptimo para la separación de las clases C_1 y C_2 . .	61
4.6. SVM con hiperplano de margen blando	62
4.7. SVM mapeada en un plano dimensional mayor	66
4.8. Representación de los datos para entrenar y probar los modelos	70
4.9. Agrupación de los datos por escenarios	70
4.10. Valores de la curva ROC de los <i>kernels</i> utilizados	73
4.11. Clasificador construido para aceleración y posición	75
4.12. SVM construida para aceleración y posición	76
4.13. SVM para la clasificación del daño para aceleración y posición	77
4.14. Área bajo la curva del modelo con aceleración, velocidad y posición	78
5.1. Componentes de un Árbol de Decisión	82
5.2. Árbol de Decisión para la detección de daño por niveles	83
5.3. Representación del conjunto de datos para la clasificación del daño	84
5.4. Construcción de un AD para la clasificación del daño	84
5.5. Tipos de variables para la construcción de Árboles de Decisión	87
5.6. Construcción de un Árbol con un atributo y dos salidas.	88
5.7. Árbol de múltiples salidas	88
5.8. División binaria de un Árbol, agrupando los valores de los atributos	89
5.9. Diferentes maneras de agrupar valores de atributos ordinales	90
5.10. Construcción de Árboles con atributos continuos	90
5.11. Probabilidades de clasificación para un nuevo objeto	91
5.12. Coeficiente de Gini	92
5.13. Gráficas para los distintos índices de impureza	94
5.14. Representación de los datos para el entrenamiento de los datos	96
5.15. Árbol construido para la clasificación del daño	98
5.16. Reglas generadas para la clasificación del daño	99
5.17. Reglas generadas para la identificación del daño por niveles	100

5.18. Árbol construido con la modificación de los parámetros	101
5.19. Importancia de las variables para el modelo clasificador	103
5.20. Reglas generadas para la identificación del daño por niveles (variables po- sición y velocidad)	104
5.21. Distinto Árbol modificando parámetros	104

Índice de tablas

3.1. Valores de las aceleraciones con la presencia de daño en A2 para el escenario 1	36
3.2. Valores de las aceleraciones sin la presencia de daño para el escenario 1	36
3.3. Características estadísticas para ambos estados del escenario 1	43
3.4. Características estadísticas para ambos estados del escenario 2	43
5.1. Matriz de confusión para la clasificación de daño	99
5.2. Matriz de confusión para la clasificación de daño por severidad	101
5.3. Matriz de confusión para la clasificación del daño por niveles	103

Capítulo 1

Introducción

En la actualidad las sociedades dependen ampliamente de sistemas tanto estructurales como mecánicos, los cuales son necesarios para llevar a cabo actividades de la vida diaria. Estos sistemas están en todas partes y se pueden ver: en puentes, caminos, centrales eléctricas, plataformas para la extracción de petróleo, edificios, sistemas de defensa, etc. Todos éstos sistemas están expuestos a condiciones internas y/o externas que afectan su usabilidad y por tanto, pueden presentarse daños que perjudican la vida de quienes las utilizan [1]. En este contexto, es necesario implementar mecanismos de detección de daños que solventen la necesidad de brindar la seguridad de sus ocupantes. Cabe mencionar que estos mecanismos de detección, usualmente se relacionan con una o más áreas de conocimiento, en las que se incluyen; el Monitoreo del Estado Estructural (*SHM*¹ por sus siglas en inglés), Monitoreo de Condición (*CM*), Evaluación No Destructiva (*NDE*) – también conocida como Pruebas No Destructivas (*NDT*), Estado y Uso del Sistema de Monitoreo (*HUMS*), Proceso de Control Estadístico (*SPC*) y Pronóstico de Daño (*DP*). Sin embargo, estas áreas son análogas al SHM y estudian otro tipo de estructuras como: máquinas rotativas (campo de estudio del CM, [Worden y Barton, 2004] [1]), localización de daño en contenedores (los cuales trabajan a presión), rieles de ferrocarriles (correspondiente al área del NDE para medir la severidad del daño); o incluso algunas otras áreas como el DP se enfocan en el estudio para predecir el tiempo de vida de un sistema, una vez se ha presentado el daño [1]. Por otro lado, el SHM es el término principal que se utiliza en lo que concierne a la detección de daños en edificios.

Según Farrar et al. [1–7], el término SHM usualmente se refiere al proceso de implementar alguna estrategia de detección de daño ya sea en la ingeniería aeroespacial, civil o ingeniería de las infraestructuras. El proceso consiste en observar (monitorear) a la estructura o sistema mecánico en cuestión sobre el transcurso del tiempo, y ya sea utilizando las mediciones de las respuestas dinámicas, las mediciones de la extracción de sus características sensibles al daño o utilizando un análisis estadístico, es posible determinar el estado

¹A partir de esta sección, usualmente será utilizado el acrónimo en inglés SHM, para referirse al monitoreo del estado estructural.

actual del sistema que se está estudiando.

Como es de esperar, en todos lados existe una preocupación constante sobre el daño que pueden sufrir las estructuras, debido al impacto económico y/o social que esto representa. A manera de ejemplo, se puede pensar en algún tipo de estructura, v.g. una presa, en la cual se ejerce una gran presión en sus paredes principales, debido al agua que golpea hacia el exterior. Por tanto, una consecuencia inmediata puede ser la fractura de alguna de sus paredes; provocando una serie de catástrofes con posibles pérdidas. Otro ejemplo, se aprecia en la industria aeronáutica, en la cual una de sus principales preocupaciones, es brindar integridad en las aeronaves para evitar incidentes aéreos que afecten a los pasajeros y a la tripulación a bordo. Un ejemplo más, se presenta en la ingeniería civil y sus edificios, los cuales constantemente están expuestos a riesgos, tanto; naturales como artificiales, lo que puede provocar el colapso de la estructura. Como estos ejemplos, existen muchos más que se presentan alrededor de todo el mundo. Por lo tanto, es necesario contar con herramientas que ayuden a solventar el problema de la detección del daño para garantizar la seguridad de las personas.

En la Figura 1.1, se muestra de manera resumida los objetivos principales del SHM que se han mencionado.



Figura 1.1: Objetivos principales del SHM

1.1. Motivación

Alrededor del mundo existen estructuras civiles (en el caso particular de esta tesis, se trata de edificios), las cuales son utilizadas diariamente como un medio sostenible de comunicación, vivienda o aquella de la cual podamos obtener algún tipo de beneficio en nuestras vidas cotidianas. Dichas estructuras están expuestas a condiciones de operación así como también condiciones naturales (v.g. sismos, viento y lluvia), por tanto, son susceptibles a daños y eventualmente a fallos que pueden ocasionar pérdidas económicas o incluso la pérdida de vidas humanas. Así mismo, existen otros factores importantes que deben ser tomados en cuenta, lo cual implica que sea objeto de estudio para la obtención de conocimiento. A continuación, se listan algunos de estos factores:

- **Evaluación de la estructura**

Implica el tipo y la importancia de la estructura, las condiciones ambientales y operacionales; las limitaciones en la adquisición de los datos, el equipo tecnológico con el que se cuenta para monitorear a la estructura, etc.

- **Identificación de parámetros**

Al presentarse el daño, surgen algunas preguntas como: ¿cuál es la causa del daño?, ¿qué se puede hacer para prevenir el daño?, ¿qué tan rápido el daño se extenderá a otras partes de la estructura?, etc.

- **Acción**

De acuerdo con los daños presentes en la estructura, se requiere tomar medidas de mantenimiento que solventen el problema, con el fin de minimizar el riesgo potencial de fallas catastróficas.

Por todos estos factores, se justifica que se establezca el paradigma del SHM para determinar y clasificar el daño de una manera temprana y así tomar las medidas necesarias.

1.2. Planteamiento del problema

La importancia de implementar estrategias eficientes para detectar el daño de una manera temprana es uno de los objetivos principales del SHM. Por tanto, es necesario analizar las características de los cambios en la rigidez, masa o energía disipada de nuestro sistema para evaluar la condición en que se encuentra. Durante varias décadas; investigadores, ingenieros y gente especializada en el área, han desarrollado métodos que cumplen con estas premisas. Sin embargo, la mayoría de los trabajos realizados, solo se enfocan en otras técnicas, basadas en frecuencias naturales y curvaturas modales [5,6,8–10]. Del mismo modo, pocos trabajos han sido realizados con técnicas de Minería de Datos, y más aún, los trabajos que han sido presentados, se basan en modelos matemáticos o en modelos experimentales

con otro tipo de variables (v.g. módulos de elasticidad, rigidez, curvaturas, etc.) [11–23], diferentes a las utilizadas en este trabajo (i.e. aceleración, velocidad y posición).

Por lo anterior, se ha decidido abordar el problema de la detección de daños en un contexto experimental, con la virtud de contar con diferentes condiciones (tanto; ambientales como operacionales, que se asemejan a las de un edificio de tamaño real), así como la implementación de algunas de las técnicas de la Minería de Datos como coadyuvante en la toma de decisiones; ya que por el desarrollo de distintos programas de software, algoritmos y maneras de visualizar los resultados, resulta una manera fácil y eficiente de tratar el problema.

1.3. Objetivos de la tesis

1.3.1. General

El objetivo general de este trabajo de tesis es el de desarrollar una metodología basada en modelos de Minería de Datos que permita detectar y localizar los daños presentes en un edificio; así como el de predecir futuros escenarios de daños que pudieran suscitarse por causa de las vibraciones.

1.3.2. Específicos

Los objetivos específicos que se plantean, son los siguientes:

- *Realizar* un análisis estadístico de los datos obtenidos para poder inferir la presencia de daño en el edificio.
- *Implementar* las Máquinas de Soporte Vectorial para la clasificación del daño en el edificio.
- *Implementar* los Árboles de Decisión para la clasificación, localización y determinación de la severidad del daño en el edificio.
- *Comparar* los modelos obtenidos de las Máquinas de Soporte Vectorial y los Árboles de Decisión, con otros modelos (i.e. K-Vecinos más Cercanos y Redes Neuronales) para comprobar su efectividad.
- *Mostrar* los resultados obtenidos de los experimentos para representar el daño en la estructura.

1.4. Organización de la tesis

De igual modo, el resto del documento de la tesis está organizado en capítulos como se describe a continuación:

El **Capítulo 2**, contiene los fundamentos teóricos que son necesarios para entender de manera precisa el contexto de investigación de esta tesis, así como una revisión de algunas técnicas de Minería de Datos que fueron utilizadas como punto de comparación con las Máquinas de Soporte Vectorial y los Árboles de Decisión. Así mismo, se presenta una revisión de algunos trabajos técnicos que tienen relación con este trabajo de tesis.

Después, en el **Capítulo 3**, se describen las herramientas tanto de hardware como de software que fueron necesarias para llevar a cabo el trabajo de tesis, i.e. la composición del sistema. Así mismo, se describe el trabajo de pre-procesamiento de los datos y el desarrollo de las técnicas estadísticas requeridas para el proceso de detección de daños.

En el **Capítulo 4 y Capítulo 5** se presenta la propuesta de implementación de las dos principales técnicas utilizadas (Máquinas de Soporte Vectorial y Árboles de Decisión), las cuales son la parte principal y de contribución de este trabajo de tesis. Al final del **Capítulo 4 y Capítulo 5**, se detallan los resultados obtenidos de las pruebas que se efectuaron para la verificación y validación de los modelos, junto con otras técnicas de Minería de Datos (K-Vecinos más Cercanos y Redes Neuronales) que fueron implementadas.

Finalmente, en el **Capítulo 6** se presentan las conclusiones a las que se llegaron tras implementar la metodología propuesta, así como las ventajas y desventajas de la utilización de una técnica comparada con otra. En la última parte del capítulo, se plantean algunas interrogantes que surgieron y que en un trabajo a futuro pueden ser abordadas para la extensión de este trabajo de tesis.

En la Figura 1.2, se observa de manera general la organización del documento de la tesis.

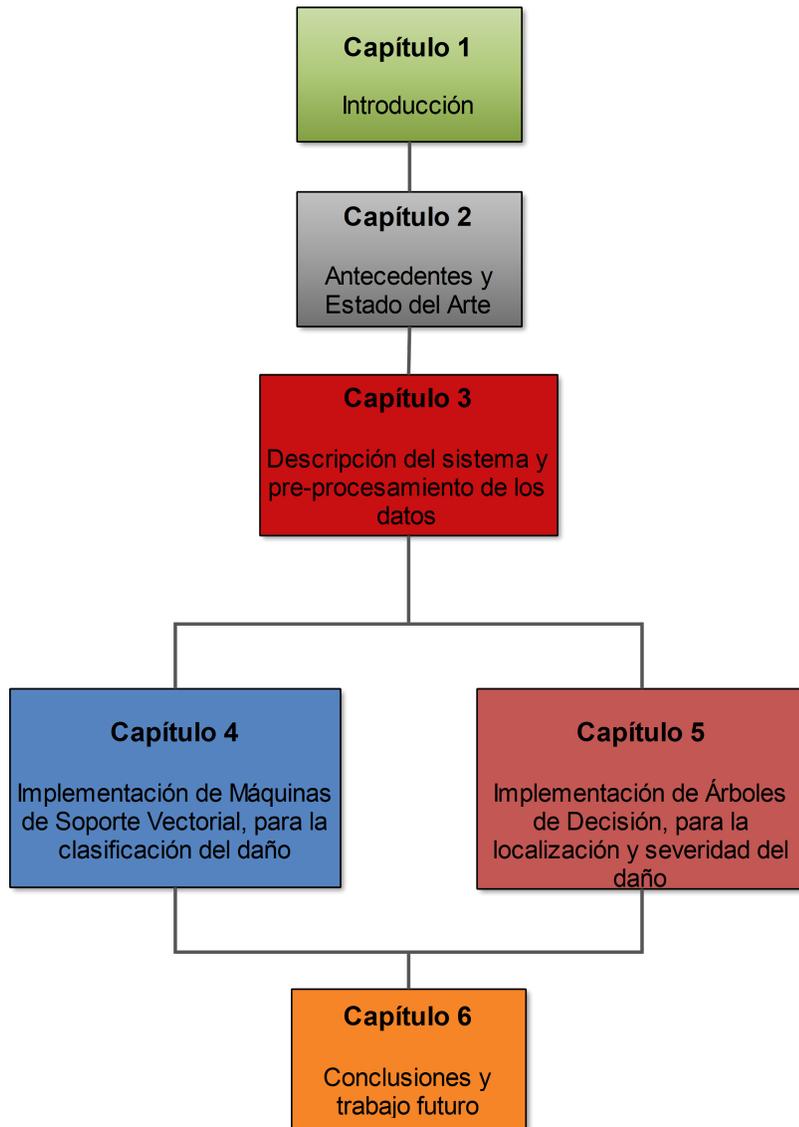


Figura 1.2: Esquema general del trabajo de tesis

Capítulo 2

Antecedentes y Estado del Arte

Para entrar en el contexto de la investigación de este trabajo, es necesario antes conocer los principios y la teoría que hay detrás del estudio del monitoreo del estado de las estructuras. Por tanto, en este capítulo se detallan los conceptos que rigen el estudio del SHM para las estructuras, así como la descripción del K-NN y las Redes Neuronales, los cuales son dos de las técnicas que fueron utilizadas en este trabajo como comparación de resultados, con la propuesta de implementación de las Máquinas de Soporte y los Árboles de Decisión. Además, se hace una revisión de los principales trabajos de investigación relacionados con la detección de daños en estructuras que se han realizado hasta el momento.

2.1. Detección de daños en edificios basado en *SHM*

Para entender el proceso de detección del daño en edificios de acuerdo al enfoque del SHM, primero se tiene que definir el concepto de daño, en específico; aquel que se presenta en las estructuras.

Definición de daño

El daño es definido como un conjunto de factores que inciden en un sistema, ya sea de manera intencional o no intencional y que afecta en un corto o largo plazo la integridad de la estructura [1]. Otra definición, más puntual, nos dice: el daño puede ser visto como el cambio en las propiedades materiales y/o geométricas de la estructura, incluyendo sus condiciones de frontera, conectividad entre elementos, secciones transversales geométricas; cargas, propiedades materiales y cualquier otro factor capaz de provocar un comportamiento inusual en un tiempo presente o futuro de la estructura [7].

Cabe destacar que: en términos de escalas, el daño se puede acumular en largos periodos de tiempo, v.g. cuando se presenta fatiga o corrosión en la estructura. Por otra parte, el

daño puede progresar rápidamente, v.g. cuando se presenta una fractura que se extiende a lo largo de la estructura. Y ya sea que se presente uno o ambos casos; conforme el daño crece, eventualmente alcanza un estado en donde afecta la operación del sistema (también referido como falla), hasta el punto que no es aceptable para su utilización [24].



Figura 2.1: Distintas estructuras alrededor del mundo

2.1.1. Causas que provocan el daño en edificios

Durante el tiempo que duran en pie los edificios, estos tienen una carga dinámica intensiva (ya sea por fenómenos naturales y/o operacionales) que puede provocar el debilitamiento-falla, afectando las características en la masa, rigidez y disipación de energía en el edificio. Si bien es cierto que existen condiciones que son la causa de los daños y que afectan la integridad de los edificios, es posible hacer una clasificación de manera general como sigue:

- **Deformación plástica**

Se refiere a la propiedad mecánica de un material, el cual tiende a deformarse de manera permanente e irreversible cuando está sometido a tensiones por encima de su rango elástico (módulo de Young ¹). Dicho lo anterior, todas las estructuras están construidas con materiales deformables, y por tanto, son susceptibles a rupturas y eventualmente a la falla de las mismas.

- **Pandeo**

¹El módulo de Young es un parámetro, el cual mide la elasticidad de los materiales; y está representado por la razón entre el incremento de esfuerzo aplicado a un material y el cambio correspondiente a la deformación que experimenta.

Se refiere a la deformación lateral de una pieza de la estructura que se curva por la parte central (por lo regular se presenta en las columnas que sostienen el edificio), al estar sometida a fuerzas de compresión en la dirección de su eje.

■ Fatiga

Es el proceso de cambio estructural permanente, progresivo y localizado, que ocurre en un material sujeto a esfuerzos repetitivos en algún punto y que puede llegar a producir fisuras o la fractura total del(os) material(es). Estas fisuras (también conocidas como grietas) producen un cambio en la geometría, así como en sus características de rigidez; y dependiendo del tamaño y la ubicación, pueden alterar el funcionamiento del sistema.

■ Pérdida de peso

La pérdida de peso, es visto como un evento en el que cambia la masa y por tanto la rigidez; pudiendo llegar a perjudicar la estabilidad de la estructura.

■ Aflojamiento

El aflojamiento, es un fenómeno que se presenta cuando se afecta una conexión atornillada, soldada o de cualquier otro tipo de unión en una estructura, y que puede dañar la conectividad entre los elementos. A menudo, esta forma de daño agrega efectos de disipación de energía hacia otras partes de la estructura, que se ve reflejado como un aumento en las propiedades de amortiguamiento.

■ Corrosión

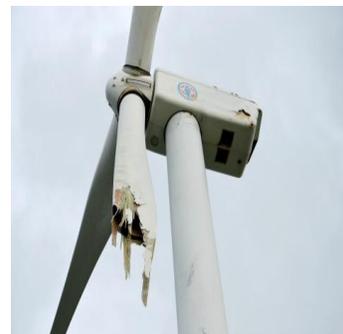
La corrosión, se presenta por la degradación de los materiales (en especial las partes en una estructura que contiene metales), causada por acción de agentes externos (v.g. lluvia ácida o algún tipo de químico) y suele presentarse como una oxidación en el material.



(a)



(b)



(c)

Figura 2.2: Diferentes estructuras dañadas, 2.2a daños estructurales de un edificio², 2.2b sección de un puente dañado³ y 2.2c una turbina de viento con el aspa dañada⁴.

2.1.2. Clasificación del daño en estructuras

El daño en una estructura puede ser clasificado de dos maneras: lineal y no lineal [6] [7].

1. **Daño lineal:** el daño puede clasificarse como lineal; cuando inicialmente la estructura permanece de manera elástica-lineal (i.e. que aún no alcanza su punto de fractura) y que después de un daño, permanece de la misma manera. Estos daños provienen de los cambios en las propiedades modales que se atribuyen a los cambios en la geometría y/o las propiedades materiales de la estructura, pudiendo ser identificados mediante el uso de algún tipo de método.
2. **Daño no lineal:** es definido como el caso en donde inicialmente la estructura se encuentra de manera elástica-lineal y luego de que el daño se hace presente, se comporta de una manera elástica-no lineal (v.g. por causa de una ruptura que se abre y se cierra tras sufrir vibraciones).

Para determinar el tipo de daño (lineal o no lineal) que sufre una estructura, se requiere de la utilización de algún método. Estos métodos pueden ser clasificados en tres categorías:

- **Basados en modelos:** a estos métodos también se les llaman problema inverso, en el que se asume que la estructura monitoreada responde de una manera determinada, de acuerdo a ciertas restricciones, tanto operativas como de entorno.

Para construir el modelo, se utilizan métodos numéricos como el análisis de elementos finitos (*FEM* por sus siglas en inglés), tal como en las teorías de *Euler-Bernoulli* o *Timoshenko* para el modelado de vigas.

- **Basados en datos:** estos métodos, basan su funcionamiento en monitorear a la estructura para obtener datos experimentales sujetos a condiciones, tales como: vibraciones, deformaciones, fatigas, etc., ya sea en tamaño real o en un prototipo a escala.
- **Híbridos:** otro enfoque menos conocido pero que se ha utilizado son los métodos híbridos, los cuales basan su funcionamiento en la combinación de modelos matemáticos, así como métodos basados en datos.

Clasificación del daño por niveles

Así mismo, se puede clasificar el daño en una serie de niveles o etapas, que van desde la detección del daño, hasta la predicción del tiempo de vida de la estructura (ver Figura 2.3) Rytter [25]. Usualmente la mayoría de literatura enfocada en el proceso de detección de daño, puede ser clasificada en el Nivel 1, Nivel 2 o Nivel 3 ya que estos niveles se relacionan con las pruebas dinámicas (v.g. aceleración y velocidad) y problemas de modelado.

²http://commons.wikimedia.org/wiki/File:Kenton_Chambers_Building_79.jpg

³http://commons.wikimedia.org/wiki/File:Taylor,_Nebraska_damaged_bridge_1.JPG

⁴<http://www.taipeitimes.com/News/taiwan/archives/2013/07/08/2003566614>

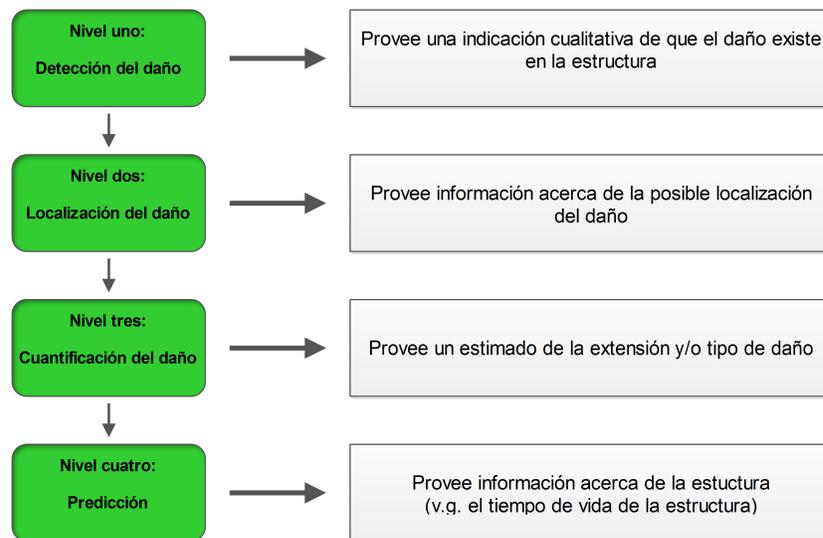


Figura 2.3: Clasificación del daño por niveles

2.1.3. Paradigma de reconocimiento del daño basado en el SHM

El proceso de implementar estrategias para reconocimiento del daño puede ser dividido en cuatro partes [1] [6]:

Evaluación operacional

La evaluación operacional describe los propósitos principales por lo cual la estructura debe ser evaluada. Específicamente, intenta responder las siguientes cuatro preguntas:

1. ¿Cuáles son las razones de seguridad y/o justificación económica para llevar a cabo el monitoreo de la estructura?.
2. ¿Cómo se define el daño para el sistema que está siendo investigado?, y para múltiples posibilidades de daño, ¿cuáles son los casos de mayor preocupación?.
3. ¿Cuáles son las condiciones operacionales y ambientales, bajo las cuales el sistema será monitoreado?.
4. ¿Cuáles son las limitaciones para la adquisición de los datos en el entorno operacional?.

Como vemos, la evaluación operacional comienza a establecer límites en qué, cómo y cuándo será monitoreada la estructura. Además, trata de adaptar el proceso de identificación de daños a las características que son únicas del sistema.

Adquisición de datos

El proceso de adquisición de datos está relacionado con el proceso del SHM que involucra aspectos tales como: los tipos de sensores que serán utilizados, el número y colocación de los sensores, el almacenamiento, los medios de transmisión para el envío de los datos recolectados, etc. Una elección de un sensado activo o pasivo, así como la velocidad de muestreo, son otras consideraciones que deben ser tomadas en cuenta.

Extracción de características

Fundamentalmente, el proceso de extracción de características está enfocado en adecuar algún modelo, ya sea basado en herramientas analíticas o en datos, para medir las respuestas del sistema. Con esto se busca obtener características que provean información de los parámetros y que después se convertirán en las características de daño. Además de los enfoques basados en las formas modales, existen otros enfoques (características) que son usados para este propósito, tales como:

- Frecuencias naturales.
- Formas en la curvatura.
- Funciones de respuesta de frecuencia y transmisibilidades.
- Flexibilidad dinámica.
- Coeficientes de amortiguamiento.

En resumen: la extracción de características, es el proceso en el cual consiste en determinar las propiedades de las respuestas medidas de la estructura, que indiquen la presencia (o no) de daño [3] [6].

Modelos estadísticos para la clasificación de características

El desarrollo de modelos estadísticos tiene que ver con la implementación de algoritmos que operan sobre la extracción de características para determinar el daño en la estructura (principalmente es utilizado en un enfoque basado en datos experimentales). Los algoritmos utilizados en el desarrollo de modelos estadísticos, usualmente caen dentro de tres categorías como sigue:

1. La **primera** categoría: se refiere a cuando los datos de la estructura están disponibles tanto con daño como sin daño, entonces los algoritmos que se utilizan son referidos como *aprendizaje supervisado, clasificación y análisis de regresión*.
2. Por otra parte, la **segunda** categoría: es cuando los datos solo están disponibles para un estado sin daño de la estructura, entonces los algoritmos se denominan *aprendizaje no supervisado*.

3. A veces el aprendizaje no supervisado también puede ser utilizado para construir un modelo dado una sola clase, y posteriormente ser utilizado para comparar nuevos datos con el modelo construido. Por lo tanto, la **tercera** categoría recae en el *análisis de datos atípicos* (*outlier* en inglés) o *detección nueva* (*novelty detection* en inglés).

2.1.4. Utilización de sensores para el monitoreo de estructuras

Una de las principales herramientas utilizadas dentro del SHM para el monitoreo de estructuras, son los sensores. Ya que por los avances de la tecnología, se ha hecho posible reducir su costo y tamaño; así como el aumento del nivel de precisión y sensibilidad, entre otras características. A continuación, se explica de manera breve los conceptos principales que hay detrás de un sensor:

Sensores utilizados para el SHM

Un sensor, es un instrumento de medición diseñado para captar información de una magnitud externa y transformarla en otra magnitud (por lo regular eléctrica), de manera que el usuario sea capaz de cuantificar y manipular. Existen diferentes tipos de sensores que sensan distintos tipos de variables, tales como: temperatura, fuerza, distancia, aceleración, humedad, presión, movimiento, torsión, etc. Los valores devueltos pueden mostrarse directamente (v.g. un termómetro) o conectados a un indicador (v.g. una computadora o una pantalla), ya sea de manera alámbrica o inalámbrica, de tal manera que puedan ser interpretados y usados. Dentro del SHM, suelen utilizarse sensores que entregan valores con relación a la aceleración, masa, rigidez y amortiguamiento de la estructura; resultando valores útiles, ya que son sensibles para determinar la presencia de daño [53]. Estos sensores, usualmente son instalados en puntos específicos de la estructura de manera alámbrica; aunque algunas veces, por las dimensiones y/o accesibilidad de la estructura, son utilizados sensores que envían los datos de manera inalámbrica.

Características necesarias en los sensores

Dentro de las características mínimas necesarias que se requieren para la obtención de valores, podemos destacar las siguientes:

- **Rango de medida:** referente al dominio de la magnitud mínima y máxima que puede aplicarse al sensor.
- **Precisión:** es la cualidad del sensor para dar lecturas próximas al valor real de la magnitud medida (i.e. el error de medida máximo que se puede alcanzar).
- **Rapidez de respuesta:** referente al tiempo que tarda en responder el dispositivo para medir las magnitudes.

- **Sensibilidad:** se refiere a la capacidad de percibir estímulos externos y dar un valor de cambio a la salida.

Además de las características anteriores, es importante la colocación de los sensores; de tal manera que se logre obtener la mayor información del registro de las características dinámicas de la estructura en cuestión [5] [11].

2.1.5. Axiomas dentro de la detección de daño

Ya que las implicaciones tanto de seguridad como económicas tienen una alta importancia, muchos estudios se han realizado con el fin de avanzar en el área del SHM. Sin embargo, es conveniente aclarar que existen axiomas que siempre están presentes en el cualquier estructura, no importando la calidad de construcción, material del que esté compuesto o condiciones operacionales y ambientales a las que esté expuesta [1] [2].

1. *Axioma I.* Todos los materiales tienen fallas o defectos.
2. *Axioma II.* La evaluación de daño requiere una comparación entre dos estados del sistema (i.e. con daño y sin daño).
3. *Axioma III.* La identificación y localización de daño, también puede realizarse a través de métodos de aprendizaje no supervisado, mientras que para determinar la severidad del daño es necesario utilizar métodos de aprendizaje supervisado.
4. *Axioma IV(a).* Los sensores no detectan el daño, sino que es necesario valerse de métodos como: procesamiento de señales y clasificadores estadísticos para convertir los datos en información relevante que lleve hacia la detección de daño.
5. *Axioma IV(b).* Sin características eficientes en los sensores (precisión, rango de medida, *offset*, etc.), los valores obtenidos pueden cambiar los resultados de los análisis.
6. *Axioma V.* El tamaño y las escalas de tiempo asociadas con la iniciación de daño y evolución del mismo, dictan las propiedades requeridas del sistema de detección del SHM.
7. *Axioma VI.* Existe un equilibrio entre la sensibilidad al daño de un algoritmo y su capacidad de rechazo de ruido.
8. *Axioma VII.* El tamaño del daño que puede ser detectado de los cambios dinámicos en el sistema, es inversamente proporcional al rango de frecuencia de excitación.
9. *Axioma VIII.* El daño se incrementa conforme a la complejidad de la estructura.

2.2. Minería de Datos

En esta sección se hace una breve introducción a cerca de la Minería de Datos, su surgimiento e importancia en este trabajo de tesis. De igual modo, se abordan las técnicas de Minería de Datos que fueron utilizadas en este trabajo como punto de comparación con las Máquinas de Soporte y los Árboles de Decisión propuestos en este trabajo.

¿Qué es la Minería de Datos?

La Minería de Datos es una nueva y promisoría tecnología, la cual puede ser definida como el proceso de descubrimiento de nuevas relaciones, patrones o tendencias significativas, a través de un “minado” de grandes volúmenes de datos almacenados propiamente en una base de datos; mediante el uso de técnicas estadísticas, inteligencia artificial (*IA*), *machine learning*, teoría de la información y técnicas de visualización [26]. Otra definición acerca de la Minería de Datos, nos dice: la Minería de Datos es una tecnología que combina los métodos tradicionales de análisis de datos con algoritmos sofisticados, para procesar grandes volúmenes de datos en búsqueda de nuevo conocimiento que sea útil [27].

A últimas fechas, una enorme proliferación de bases de datos en todas las áreas ha surgido, demandando poderosas herramientas capaces de realizar tareas de análisis de los datos obtenidos en búsqueda de nuevo conocimiento. Estas áreas incluyen; los negocios, la medicina, la ciencia y la ingeniería, en las cuales se requiere ofrecer por ejemplo; mejores productos, calidad de servicios de salud, anuncios de nuevos descubrimientos, utilización de nuevas tecnologías, etc., que ayuden al bienestar de las personas [27]. En el afán de satisfacer las crecientes necesidades de las personas, los investigadores han desarrollado nuevos métodos capaces de resolver estos problemas, dando como resultado un nuevo campo, denominado Minería de Datos y descubrimiento del conocimiento (*KDD* por sus siglas en inglés). En donde, por un lado; el *KDD* se enfoca en las relaciones y patrones entre los datos, y por otro lado; la Minería de Datos describe el descubrimiento de etapas del proceso del *KDD*, i.e. que la Minería es una parte integral del descubrimiento de nuevo conocimiento.

2.2.1. División de la Minería de Datos

La Minería de Datos se divide en dos categorías como sigue:

- **Análisis descriptivos**

En este tipo de análisis se trata de encontrar patrones (correlaciones, grupos, trayectorias y anomalías) entre las relaciones de los datos, de tal modo que sean capaces de brindar un panorama del comportamiento de los datos estudiados. Las tareas de análisis descriptivos son frecuentemente de carácter exploratorio (v.g. Análisis de

Clusters) y a veces requieren de otras técnicas de pos-procesamiento para validar los resultados.

■ Análisis Predictivos

El objetivo de este análisis, es el de predecir el comportamiento de nuevos datos de los atributos objetivo o variables dependientes, a partir de otros atributos de prueba (también llamados variables independientes). En este tipo de análisis destacan la clasificación y la regresión (v.g. Análisis Lineal Discriminante).

En este trabajo de tesis se realizó un análisis predictivo, i.e. debido a que fueron utilizados métodos supervisados, en donde los datos fueron etiquetados con las clases; Con Daño y Sin Daño, con el fin de predecir la clase a la podrían pertenecer un nuevo conjunto de datos.

2.2.2. Proceso para la detección del daño con el enfoque KDD

Para llegar a obtener un conocimiento en cuanto a la presencia y localización de daños en el edificio, es necesario seguir un proceso. Este proceso, consiste en una serie de pasos que transforman los datos obtenidos en información.

En la Figura 2.4, se describe el proceso que fue utilizado para desarrollar la propuesta de la detección del daño en el edificio. Como primer paso; se contempla la recolección y almacenamiento de los datos provenientes de los sensores instalados en el edificio, posteriormente; realizar un análisis estadístico básico, con el propósito de entender el comportamiento de los datos; luego en la etapa del pre-procesamiento, se filtran y se transforman los datos del dominio del tiempo al dominio de la frecuencia (ver Capítulo 3 para referencias de los pasos anteriores). En las siguientes etapas, se hace uso de la Minería de Datos para la construcción de modelos y la generación del conocimiento para la representación de los resultados(ver Capítulo 4 y Capítulo 5).

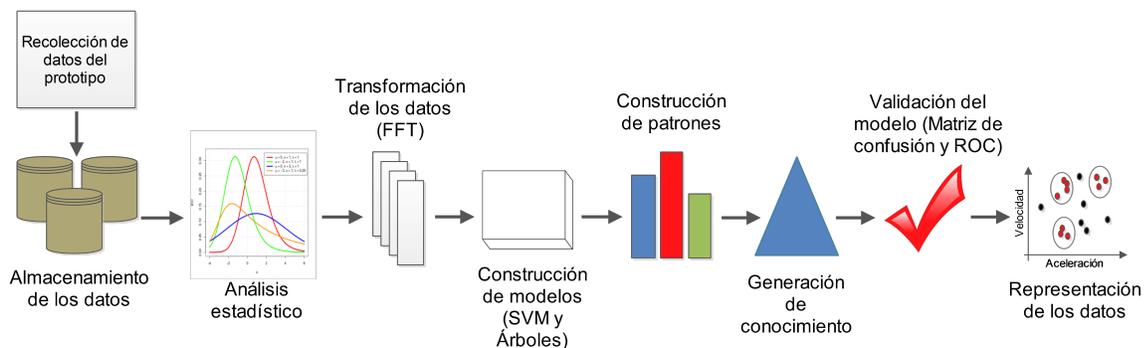


Figura 2.4: Proceso para la detección de daño

A continuación, en las sub-secciones siguientes; se describe de manera breve la teoría que existe detrás del K-NN y las Redes Neuronales, las cuales fueron adecuadas a la resolución de nuestro problema del daño. Estas técnicas se compararon con los resultados las Máquinas de Soporte y los Árboles de Decisión. De igual modo, se presenta una breve discusión del Análisis de Clusters, el cual fue otro de los métodos utilizados; sin embargo, debido a la entrega de los malos resultados, no fue incluido en esta comparación.

2.2.3. Análisis K-NN

Dentro de los métodos más utilizados en Minería de Datos, destaca el *K-NN*⁵ (*K-Nearest Neighbors* por sus siglas en inglés, [Fix y Hodges, 1951]) [27] [28], el cual es un método de clasificación de aprendizaje supervisado, utilizado para estimar la función de densidad $f(x|C_i)$ de las variables predictivas x 's por cada clase C_i . Dicho de otra manera; estima la probabilidad de que un elemento nuevo y sin etiqueta x , de un conjunto de datos pertenezca a cualquier clase C_i con mayor frecuencia a la que pertenecen sus K vecinos más cercanos, de acuerdo con una métrica de distancia (v.g. distancia *Euclidea*).

Estimación de densidad no paramétrica para K-NN

Dado que K-NN es un método de estimación no paramétrico n -dimensional, entonces el espacio de la función de densidad de probabilidades puede ser dividido en hiper-cubos con un tamaño h y un volumen V , dado $x_i = 1, 2, \dots, N$, como los vectores que contienen a los elementos del conjunto de datos. Entonces la probabilidad de que un objeto x' caiga dentro de un volumen definido V , centrado en un punto x , está dada por:

$$\theta = \int_{V(x)} p(x) dx, \quad (2.1)$$

donde la integral calcula el volumen V como sigue:

$$\theta \sim p(x)V. \quad (2.2)$$

La probabilidad de θ , puede ser aproximada por la proporción de objetos que caen dentro del volumen V como:

$$\theta \sim \frac{k}{N}, \quad (2.3)$$

⁵Comúnmente es utilizado el término K-NN para referirse al método de los K-Vecinos más Cercanos.

donde las Ecuaciones 2.2 y 2.3 calculan la densidad de k número de vecinos de un total de N , que están dentro de V (siendo k una función de x) como:

$$\widehat{p}(x) = \frac{k}{NV}. \quad (2.4)$$

En resumen: el enfoque K-NN trata de encontrar la probabilidad de k/N y un volumen determinado V , el cual contendrá los k vecinos próximos a un objeto x .

Regla de decisión para K-NN

Una vez calculado la estimación de densidad para K-NN, la regla de decisión K-NN, es equivalente a aplicar reglas de *Bayes* para encontrar una decisión de la clase C_i a la que pertenecerá un nuevo objeto x , de acuerdo a su vecino más cercano.

Dado un conjunto de datos de entrenamiento con sus respectivas clases tales: (x_i, C_i) , con $i = 1, 2, \dots, N$, donde x_i es el i -ésimo vector con los atributos de entrenamiento y C_i es la clase a la cual será clasificado un nuevo objeto x (tal que: $C_i = j$, si el i -ésimo vector de entrenamiento es un objeto de la clase D_j). Suponiendo que; dados los primeros k vecinos, existen k_m en una clase dada C_m de N_m (tal que; $\sum_{m=1}^C N_m = N$). Entonces se puede estimar la probabilidad $p(x|C : m)$, como:

$$\widehat{p}(x|C_m) = \frac{k_m}{N_m V}, \quad (2.5)$$

y la probabilidad $\widehat{p}(C_m)$, está dada como:

$$\widehat{p}(C_m) = \frac{N_m}{N}, \quad (2.6)$$

entonces la regla de decisión es asignar un objeto x a una clase C_m , de acuerdo con el teorema de *Bayes*:

$$\frac{k_m}{N_m V} \frac{N_m}{N} \geq \frac{k_i}{N_i V} \frac{N_i}{N}, \quad \forall i, \quad (2.7)$$

i.e. asignar x a C_m , si:

$$k_m \geq k_i, \quad \forall i. \quad (2.8)$$

En resumidas palabras: la regla de decisión consiste en asignar x a la clase que recibe mayor número de k vecinos próximos (frecuentes).

K-NN para la detección de daños

De acuerdo a los conceptos vistos anteriormente, ahora es posible adecuar un algoritmo *K-NN* al problema de detección de daño en nuestro edificio:

Algoritmo, K-NN para el SHM

- 1: **Entrada** de los vectores: (a_i, v_i, p_i) , con C_m , conjunto de entrenamiento.
 $x = (x_1, x_2, \dots, x_N)$, nuevo(s) objeto(s) a clasificar.
 $k = (1, 2, \dots, N)$, elección de k .
 - 2: **Para** todo $((a_i, v_i, p_i), C_m)$.
 calcular $d_i = d(x, (a_i, v_i, p_i))$, distancia Euclideana.
 - 3: **Ordenar** $d_i = (1, 2, \dots, N)$.
 - 4: **Obtener** o número de objetos de $((a_i, v_i, p_i), C_m)$ de los k vecinos más próximos.
 - 5: **Asignar** x a la clase con mayor frecuencia de $((a_i, v_i, p_i), C_m)$.
 - 6: **Fin.**
-

Dado el algoritmo propuesto; se tienen como entrada a los datos de entrenamiento (pre-procesados) obtenidos de los acelerómetros; representados como en la Ecuación 2.9:

$$((a_i, v_i, p_i), C_m), \text{ con } i = 1, 2, \dots, N \in \text{Datos de entrenamiento}, \quad (2.9)$$

donde a_i, v_i, p_i son los vectores que contienen los atributos de aceleración, velocidad y posición; y además cada uno contiene una etiqueta de la clase $C_m \equiv \{C_1 = \text{Con Daño}$ o bien, $C_2 = \text{Sin Daño}$, que representa la presencia o ausencia de daño (ver Capítulo 3). Así mismo, se tiene el vector x y k , que representan a los nuevos objetos y el k número de elementos próximos a x respectivamente. Continuando con el algoritmo: un nuevo objeto será asignado a la clase C_m (Con Daño o Sin Daño) si es la clase con más objetos cercanos de entrenamiento entre los k vecinos próximos, i.e. de acuerdo con la métrica de distancia Euclideana como sigue:

$$d(x, (a_i, v_i, p_i)) = \sqrt{\sum_{i=1}^n (t_n(x) - t_n(a_i, v_i, p_i))^2}. \quad (2.10)$$

En la Figura 2.5, se propone un ejemplo de clasificación hipotético con respecto a nuestro problema. El problema consta de decidir en qué lugar pertenece un nuevo objeto (estrella), dada la *Clase 1* (Con Daño) y *Clase 2* (Sin Daño), con $k = 3$. De acuerdo a la figura, el nuevo objeto sería clasificado en la Clase 2, ya que existen dos círculos y solo un cuadrado dentro del radio que los contiene. Sin embargo, si el valor de k se cambia por 5 ($k = 5$), el objeto se clasifica dentro de la Clase 1, ya que por el aumento del vecindario y la distancia, ahora existen más objetos próximos de la Clase 1 que de la Clase 2.

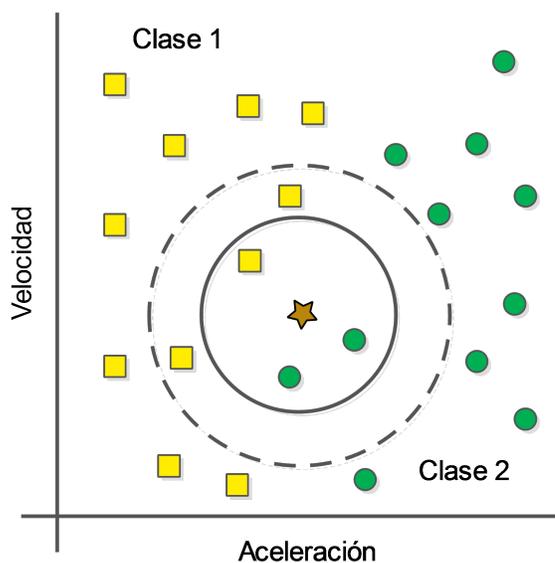


Figura 2.5: Clasificación de un nuevo objeto con K - NN

2.2.4. Análisis de Redes Neuronales

Otro de los métodos que fueron utilizados para realizar la comparación con las Máquinas de Soporte y los Árboles de Decisión, son las *Redes Neuronales Artificiales*; también conocidas como “*Redes Neurales*” o simplemente *ANN* [29] [30].

Podemos enfocarnos en las Redes Neuronales ya que pueden aprender de relaciones altamente no lineales, ayudando a resolver distintos problemas de clasificación dentro de distintos ámbitos. Las Redes Neuronales Artificiales (*ANN*'s ⁶ por sus siglas en inglés), como su nombre lo sugiere; son modelos computacionales abstractos inspirados por el cerebro humano, los cuales intentan simular de groso modo a las neuronas de un sistema biológico nervioso central (humano o animal), neurona por neurona.

Naturaleza biológica de las Redes Neuronales

Las Redes Neuronales Artificiales tienen su base matemática y computacional en las células nerviosas biológicas, mejor conocidas como neuronas (ver Figura 2.6), las cuales están inter-conectadas unas con otras para formar una Red Neuronal (ver Figura 2.7). Una neurona se compone de diferentes elementos; el primero es el *cuerpo de la célula*, el cual contiene un *núcleo*, en donde se controla la información que dirige la actividad de la neurona. Otros elementos de los que se compone una neurona son: las *dendritas*, las cuales son la parte por la cual se reciben los impulsos nerviosos de otras neuronas; el *axon*, el cual se encarga de distribuir los impulsos eléctricos recibidos por las dendritas de la misma neurona hacia la *región sináptica* (o botones terminales) y propiamente los *botones terminales*,

⁶A partir de ahora, una Red Neuronal Artificial, también podrá ser nombrada como *ANN*.

en los cuales se reciben o envían los impulsos eléctricos que se transmiten de una neurona a otra, mediante una conexión llamada sinapsis.

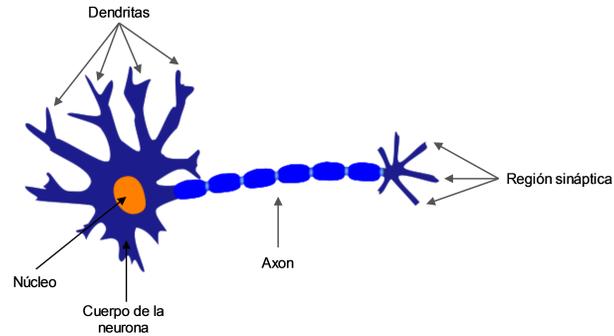


Figura 2.6: Neurona biológica

Una neurona puede tener cientos de sinapsis (i.e. que puede conectarse y comunicarse con otras neuronas mediante señales eléctricas), y dado que una neurona posee muchas dendritas, una sola neurona puede recibir mensajes de otras neuronas formando una red de neuronas. Sin embargo, no todas las inter-conexiones en una neurona tienen el mismo peso en cuanto a relación de transmisión de señales se refiere, ya que algunas tienen mayor prioridad (mayor peso) sobre otras neuronas.

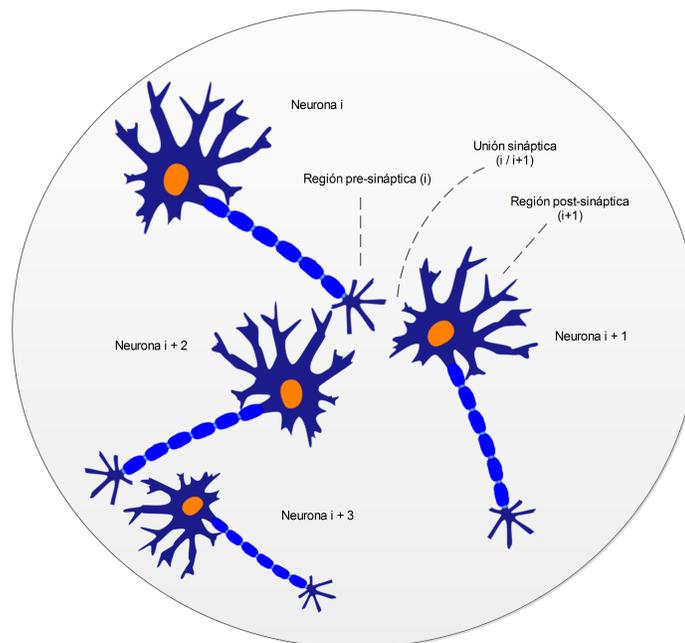


Figura 2.7: Red Neuronal biológica

Principios de Entradas/Salidas del perceptrón

Tal y como lo propuso Rosenblatt (1958), el perceptrón es la estructura básica para todas las ANN's. Éstos están representados básicamente como relaciones de entrada/salida. Entonces: dado nuestro problema; podemos representar nuestras entradas de las aceleraciones, velocidades y posiciones tales que:

$$C = \sum_i w_i(a_i, v_i, p_i) \quad (2.11)$$

$$y = f_N(c),$$

donde w_i es el peso en las entradas (a_i, v_i, p_i) , con $i = 1, 2, \dots, N$, donde c es el nodo de salida (nodo sumatorio de los pesos) y f_N es un operador no lineal que entrega como salida un valor y , tal y como se ve en la Figura 2.8. Por otro lado, se puede hacer uso de una red de perceptrones (denominada, *Red Neuronal de Perceptrones*). Donde se denota la suma de la salida del i -ésimo perceptrón como $c_i \equiv \{C_1 = \text{Con Daño o bien}, C_2 = \text{Sin Daño}\}$, dadas sus entradas $(a_{1i}, v_{1i}, p_{1i}), \dots, (a_{Ni}, v_{Ni}, p_{Ni})$, tales:

$$c_i = \sum_{j=1}^m w_{ij}(a_{ij}, v_{ij}, p_{ij}), \quad (2.12)$$

siendo w_{ij} el peso (puede ser ajustado) de la j -ésima entrada de la i -ésima célula. Por tanto, la Ecuación 2.12 puede ser escrita como un vector de la forma:

$$c_i = w_i^T(a_i, v_i, p_i), \quad (2.13)$$

donde:

$$w_i = [w_{i1}, w_{i2} \dots, w_{iN}]^T \quad (2.14)$$

$$(a_i, v_i, p_i) = [(a_{i1}, v_{i1}, p_{i1}), \dots, (a_{iN}, v_{iN}, p_{iN})]^T.$$

Una representación análoga de un perceptrón, se muestra en la Figura 2.8, en la cual dadas nuestras entradas (a_i, v_i, p_i) provenientes de otra neurona, cada una con diferentes pesos w_i , entran al bloque que representa a la neurona; así mismo se tiene una función $f_N(c)$ de activación y sus respectivas salidas y .

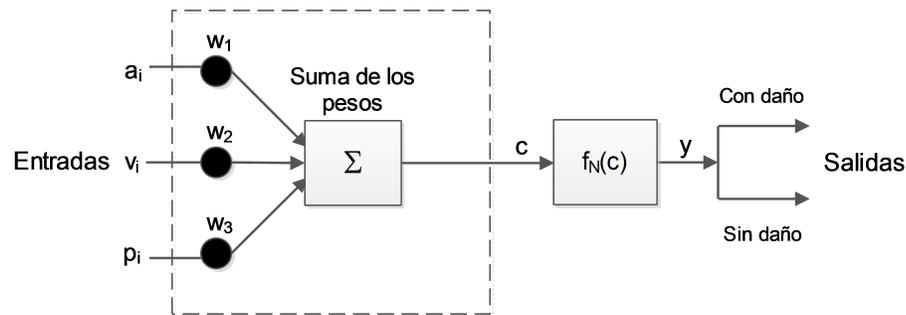


Figura 2.8: Perceptrón (Neurona artificial)

Funciones de activación para un perceptrón

La operación de activación mantiene la salida de la neurona artificial entre ciertos límites como en una neurona biológica y se encuentra en términos de una *función de activación* $f(c_i)$ la cual es una función no lineal que produce la i -ésima salida y_i que satisface:

$$y_i = f(c_i). \quad (2.15)$$

Diferentes funciones $f(c_i)$ son utilizadas para la activación, sin embargo, una de las funciones más comunes y la que fue utilizada para esta técnica, es la *función sigmoide* como sigue:

$$y_i = \frac{1}{1 + \exp(-c_i)} = f(c_i), \quad (2.16)$$

la cual es una función de activación unipolar que va de 0 a 1 en el eje ordenado. Usualmente, otras funciones de activación pueden ser utilizadas, tal y como funciones binarias o bipolares.

Algoritmos de entrenamiento para la detección de daños con ANN's

Uno de los principales algoritmos utilizado para entrenar una ANN, es el denominado: retro-propagación (*Back-Propagation* en inglés), el cual es un algoritmo basado en multi-capas de perceptrones, dando una solución a las capas ocultas de aprendizaje (Rumelhart et al., 1986) [29]. El algoritmo pretende ajustar los valores de las conexiones entre las capas ocultas y la capa de salida de una ANN para obtener un valor de convergencia deseado, mediante una propagación del error hacia atrás. Siendo E el error en la capa de salida, tal que:

$$E = \frac{1}{2} \sum_k (d_k - y_k)^2 = \frac{1}{2} \sum_k E_k^2, \quad (2.17)$$

en donde, $k = 1, 2, \dots, N$; siendo N el número de neuronas en la capa de salida, y_k la salida obtenida y d_k es la salida deseada (conf. [29] [30]). En la Figura 2.9 se muestra la estructura de una sola capa para una ANN de tipo retro-propagación con sus entradas (X) y salidas (O), así como la capa oculta de neuronas.

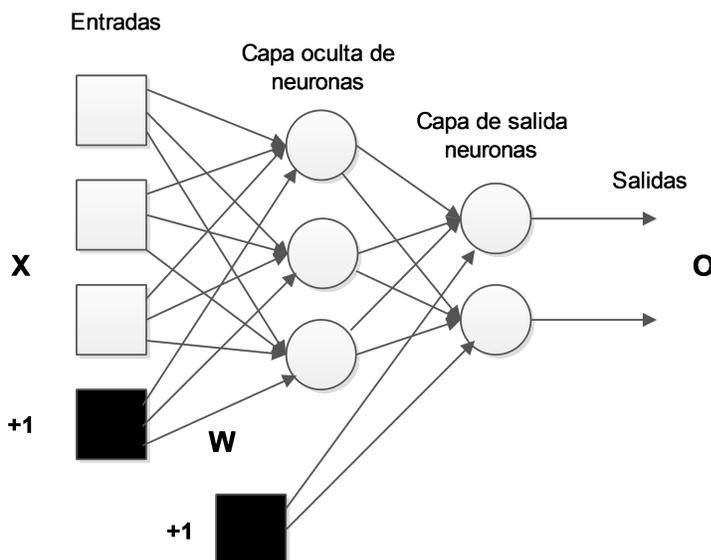


Figura 2.9: ANN con retro-propagación como algoritmo de entrenamiento

Este algoritmo de entrenamiento puede ser utilizado para el problema de clasificación de daño en edificios, ya que es capaz de resolver problemas altamente no lineales, tomando como entradas nuestros valores de aceleración, velocidad y posición; para al final entregar una clase de salida C_i (Con Daño o Sin Daño) capaz de detectar la presencia del daño.

2.2.5. Análisis de Clusters

La última de las técnicas que fueron utilizadas en este trabajo, es el Análisis de Cluster. Esta técnica consiste en dividir los datos en grupos o *clusters*⁷, de tal manera que sean significativos, usables o ambos. El objetivo es que los objetos dentro de un cluster cumplan con un criterio, ya sea de distancia o similitud uno con otro (conocida como distancia *intra-cluster*), y al mismo tiempo los grupos sean distintos con respecto a otros grupos (conocida como distancia *inter-cluster*) [27].

Sin embargo, esta técnica no fue comparada con otras de las técnicas, debido al mal desempeño que mostró en un principio; i.e. porque los datos se encuentran de manera linealmente no separable y su alta correlación entre variables, y por tanto; fue requerido otro

⁷La palabra cluster se refiere a un grupo de datos agrupados por alguna medida de distancia o de similitud.

tipo de técnicas más potentes para realizar la clasificación, tal es el caso de las técnicas mencionadas en sub-secciones anteriores, así como las Máquinas de Soporte y los Árboles de Decisión. Cabe mencionar que algunos autores han logrado tener éxito al utilizar esta técnica, y con otro tipo de variables [11–15, 18]. Sin embargo, los autores hacen uso de la combinación de otros métodos como el *PCA* o el *SVD* para lograr reducir la dimensionalidad y mejorar la visualización de los resultados. Aunque al aplicar esta técnica no se garantiza la clasificación que se desea, debido a la complejidad del problema que se está tratando. En la Figura 2.10, se muestra uno de los resultados de aplicar esta técnica.

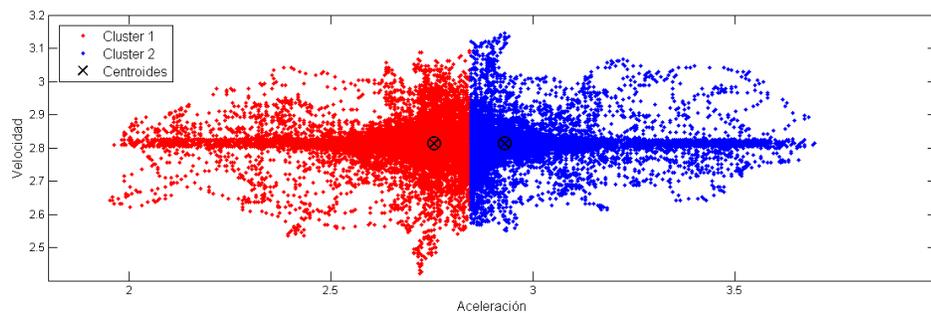


Figura 2.10: Resultado de uno de los experimentos con Análisis de Clusters

2.3. Métodos para la detección de daños en estructuras

Dentro de la biografía técnica, se encuentran distintos métodos que han sido aplicados en la detección del daño en estructuras. Algunos de ellos están basados en el estudio de las «frecuencias y resonancias», formas modales, curvaturas modales y otras más, basadas en vibraciones.

2.3.1. Detección de daño con Análisis de Clusters

En 2010, Lazarevic et al. [18], tratan de localizar el daño en una estructura con múltiples variables objetivo, y para predecir el valor de dichas variables se basan en la obtención de datos de las frecuencias naturales (medidos de la estructura, d_1, d_2, \dots, d_n), los cuales están compuestos por el par c, O , donde c son los atributos de la frecuencia medida, $c = c_1, c_2, \dots, c_m$ y $O = o_1, o_2, \dots, o_p$, son las variables objetivo. Para la determinación de daño, proponen un particionamiento jerárquico de clusters, el cual consiste de tres partes básicas que se aplican recursivamente: (1) particionar el grupo de datos, (2) localizar los grupos de interés en las variables objetivo y (3) predecir las variables objetivo. También incluyen el uso de K-NN basado en densidades, y tomando los valores de las frecuencias más bajas, logran determinar en qué lugares se puede presentar daño.

En otro trabajo [12], los autores utilizan el algoritmo de agrupamiento K-medias para hacer las agrupaciones de los datos que son similares, de acuerdo a una métrica de similaridad (distancia Euclidiana). El aspecto importante en este trabajo es que utilizan el criterio de entropía dada una función objetivo para ser utilizada como un criterio de parada y encontrar un número de clusters óptimo para realizar la clasificación del daño.

2.3.2. Detección de daño con Bosques Aleatorios

De igual manera que las técnicas de clusters, los Bosques Aleatorios, han sido utilizados para clasificar el daño en estructuras [15] [31]. En uno de los trabajos [15], los autores detectan y clasifican el daño para estructuras un tanto complejas, esto lo hacen con ayuda de un modelo de análisis de elementos finitos. Los autores, obtienen los valores del módulo de elasticidad (módulo de Young) de cada elemento (columnas principales) en la estructura, que posteriormente utilizan para entrenar un Bosque Aleatorio y predecir el daño. De acuerdo con los autores, este método es capaz de detectar el daño en estructuras de escalas mayores y complejas, a través de la simplificación de la estructura en sub-estructuras más pequeñas y de esa manera analizar los elementos de una manera individual, permitiendo detectar el daño por sectores y aumentando la escalabilidad y la identificación de múltiples elementos de falla con mayor exactitud.

2.3.3. Otros métodos basados en Minería de Datos

Otros enfoques basados en Minería de Datos han sido también aplicados en la detección de daño; v.g. el que proponen algunos autores [22], basado en Mapas Cognitivos Difusos (*FCM* siglas en inglés) con aprendizaje Hebbiano no supervisado para la detección de daño en vigas, con el análisis de elementos finitos. Los autores construyen una matriz de pesos de acuerdo a las frecuencias naturales de la viga, relacionando los valores de las frecuencias con conceptos (asignan mayor peso a las relaciones de las secciones que presentan el daño), posteriormente entrenan una Red Neuronal y logran determinar en qué nivel de la estructura se presenta el daño. En otro trabajo realizado [24], los autores logran identificar el daño presente en un puente, mediante la clasificación de valores atípicos obtenidos de las resonancias de dicha estructura; y mediante el uso del algoritmo K-medias son capaces de indicar la presencia de daño. Un trabajo más es el de Lieven et al. [32], el cual basa su funcionamiento en la combinación de componentes independientes y Redes Neuronales para la clasificación del daño en estructuras (i.e. vigas). Así mismo, en otros trabajos; Rodellar et al. [33] [34], logran hacer la detección del daño en placas de aluminio con el uso de sensores piezoeléctricos, mediante la utilización del PCA y un enfoque estadístico; Mujica et al., (2014) [35], de igual modo hacen uso del PCA combinado con la inferencia estadística multivariable para la detección de daños en una placa, logrando realizar la detección del daño con éxito. Así mismo. Li et al. [48] han logrado tener éxito con el uso de PCA.

2.3.4. Métodos basados en el cambio de las frecuencias

La presencia de daño o deterioro en una estructura conlleva a la modificación de las frecuencias naturales de la misma, por tanto, existen métodos para determinar esos cambios que afectan a dicha estructura y con ello inferir la presencia del daño.

Dado los múltiples cambios de frecuencia que pueden ser obtenidos de las estructuras, estos pueden proveer de información espacial acerca del daño estructural, ya que los cambios en la estructura en diferentes localizaciones causarán diferentes combinaciones en las frecuencias modales. En varios estudios se muestran los resultados en el que intentan identificar el daño en un catálogo de posibles escenarios [10] [36]. Los autores, calculan la frecuencia de los cambios de distintos escenarios (con daño y sin daño). Después las proporciones de todos los cambios de frecuencia son calculados mediante el uso de un análisis de correlación modal (*MAC*), obteniendo las matrices de las frecuencias. Por último, comparan las matrices y de acuerdo a un rango de valores que varían según el daño (los valores están en el intervalo de $[0, 1]$ según su correlación), son capaces de detectar la presencia de daño. En otro trabajo, Gudmunson [8], utiliza un enfoque de detección de daño basado en las perturbaciones de la energía, derivada de las frecuencias resonantes, provocadas por el daño presente en la estructura.

Métodos basados en ondas Wavelets

En este tipo de métodos, los datos de las formas modales son tratados como una señal en el dominio espacial. Las señales son cambiadas, usando una transformación espacial *Wavelet* para detectar la irregularidad de las señales causadas por el daño. En general, se puede decir que cuando una estructura pierde rigidez en alguna parte, los valores sensados (en la parte afectada o cerca de ella) tienden a tener amplitudes de onda mayor (reflejado en los picos de las ondas) en donde se localiza el daño. Muchos autores [9] [37] [38], se basan en este método para analizar las señales de los dispositivos instalados en las estructuras y en base en los valores obtenidos, son capaces de detectar determinado daño en una estructura. Liew y Wang [8], utilizan los coeficientes Wavelet para la detección del daño, basados en una solución numérica para las deflexión de una viga bajo oscilaciones. Zhong y Oyadadiji [8], se basan en este método para detectar las rupturas en una viga simétrica, con el uso de la transformada Wavelet de los datos obtenidos de las formas modales. Chang y Chen [8] utilizan las Wavelets de Gabor para la detección múltiple de daño en una viga, dadas las frecuencias naturales y las posiciones de las rupturas. Así mismo, son capaces de predecir la profundidad de las rupturas mediante la utilización de técnicas de optimización.

2.3.5. Métodos basados en análisis estadístico

Muchos han sido los autores [1, 2] [6] [18] [39] [40] [41] [42], que han utilizado técnicas estadísticas para poder localizar el daño en estructuras. Los autores sostienen que un para-

digma estadístico de reconocimiento puede ser descrito en dos categorías: (Aprendizaje no supervisado y Aprendizaje supervisado).

Métodos de Series de tiempo

Las técnicas de series temporales originalmente fueron desarrolladas para analizar grandes secuencias de datos en las que se toman los datos de prototipos o modelos matemáticos y con base en relaciones se puede llegar a predecir el comportamiento de una estructura. Han sido numerosos estudios en donde modelos de series de tiempo fueron utilizados como herramientas auxiliares de identificación de características dinámicas y posteriormente para la detección de daño [43].

Métodos Auto-Regresivos

Algunos autores [42] [44], se valen de la utilización de modelos auto regresivos (*AR*) de series de tiempo junto con otras herramientas (*PCA*, para la reducción de dimensionalidad de los datos y Redes Neuronales para clasificar y estimar el daño), para describir el historial de aceleración de los sensores instalados en las estructuras; i.e. con el fin de predecir comportamientos que podrían sufrir las estructuras tras un daño.

2.3.6. Métodos basados en sus características dinámicas

Al día de hoy, existen numerosos análisis que permiten estimar respuestas dinámicas de estructuras modeladas tanto; analíticamente como experimentalmente. Todos estos análisis tienen en común que tratan de identificar el daño sin afectar la integridad de la estructura. Desde el año de 1984, West muestra que es posible utilizar la información de las formas modales para la localización de los daños estructurales, sin la necesidad de utilizar antes un método de elementos finitos (*FEM*) [7]. El autor utiliza el criterio de correlación modal (*MAC*) para determinar el nivel de correlación entre modos de prueba de una estructura, antes y después de ser sometida a daño. Las formas modales son particionadas junto con los cambios en los valores *MAC*, para conocer los daños que se presentan en dicha estructura.

En 1997, Ratcliffe, presenta una técnica para localizar el daño en una viga, utilizando una aproximación de diferencia finita de un operador Laplaciano en las formas modales de los datos [7]. El mismo año Cobb y Liebst, presentan un método para priorizar la localización de sensores para la identificación de daño en una estructura, basado en un análisis de sensibilidad de los eigen-valores de los datos de cada sensor [7]. Stubbs, et al., (1992) [7], implementan un método que calcula el cambio en la energía de deformación almacenada en una estructura, basado en el hecho de que cuando se presenta un daño, la distribución de energía se verá afectada, disminuyendo así la deformación almacenada.

2.4. Conclusión

En este capítulo se presentó en términos generales la premisa del SHM, incluyendo los conceptos y herramientas necesarias para lograr los objetivos planteados en el Capítulo 1. Por otra parte, se introdujo al lector en los conceptos principales de la Minería de Datos, así como las técnicas que fueron utilizadas para la comparación con las Máquinas de Soporte y los Árboles de Decisión, tales como: los K-Vecinos más Cercanos y las Redes Neuronales. Estas dos últimas técnicas, han sido utilizadas en otros trabajos para la clasificación del daño en distintas estructuras. Sin embargo, como ya se mencionó en secciones anteriores, los autores utilizan otro tipo de variables (v.g. módulos de elasticidad, flexiones, energías, etc.) para entrenar sus modelos. En nuestra propuesta, se plantea utilizar los valores obtenidos de los acelerómetros instalados en cada nivel del edificio, con el afán de utilizar otros parámetros que no han sido utilizados (por lo menos, no con las técnicas propuestas) para realizar las detección del daño.

Dentro de las *contribuciones* que se tienen en este trabajo destacan:

- La realización de un análisis estadístico, con el afán de inferir la presencia del daño en el edificio.
- El desarrollo de las Máquinas de Soporte Vectorial y los Árboles de Decisión; para la detección, localización y severidad del daño en un edificio, con los valores obtenidos de los acelerómetros. Así mismo, una comparación de los resultados de las técnicas anteriores, con otras de las técnicas de aprendizaje (K-NN y Redes Neuronales).

Capítulo 3

Sistema para la detección de daños

En este capítulo se hace una descripción de los componentes del sistema; tanto de *hardware* como de *software*, que fueron utilizados para realizar el proceso de la detección del daño. De igual modo, se detalla el proceso de adquisición de los datos provenientes de la estructura, la descripción del análisis estadístico y el pre-procesamiento que se realizó a los conjuntos de datos obtenidos de las pruebas.

3.1. Descripción del sistema

En esta sección, se describen los componentes (i.e. *hardware* y *software*) utilizados para el proceso de detección del daño.

3.1.1. Edificio

Para la realización de las pruebas, fue utilizada una estructura de laboratorio a escala de un edificio de tres niveles (ver Figura 3.1); la estructura está equipada con los requerimientos necesarios para obtener los datos que fueron utilizados para la detección del daño (ver sección 3.1.2).

El edificio está conformado por platos y columnas de aluminio ensamblados con pernos; los cuales mantienen unidos a estos dos elementos. Además, la estructura está montada en una base móvil en donde son verificadas; la velocidad y la posición, durante la excitación de las vibraciones (en nuestro caso; la simulación de sismos). Cabe mencionar que; la dirección del movimiento de la estructura es unidireccional en un eje X . La Figura 3.2 muestra la representación esquemática de los componentes principales y las medidas del edificio.

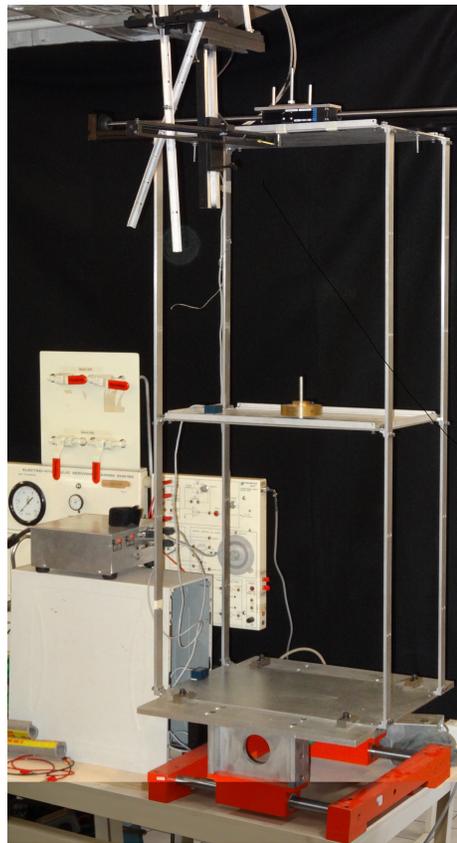


Figura 3.1: Estructura de prueba para la detección de daños

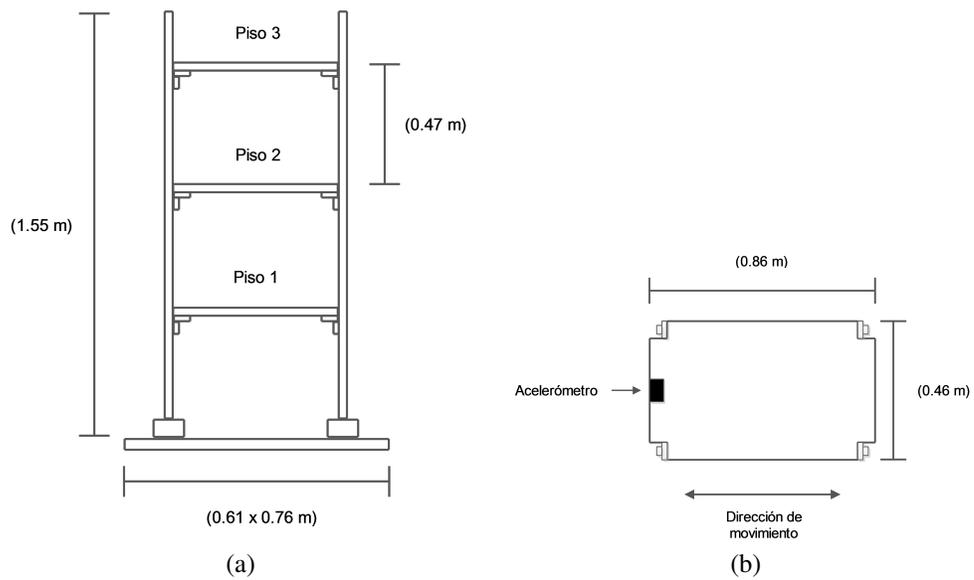


Figura 3.2: Dimensiones principales del edificio

3.1.2. Sistema para la adquisición de los datos

Las variables que se obtienen a partir de las mediciones, son de importancia para el análisis estadístico, de tal modo que brindan una visión del estado real de nuestra estructura y de cómo afectan los cambios vibratorios a dicha estructura. A continuación, se listan los elementos (sensores y actuadores) que están instalados en el edificio:

- **Acelerómetro**

Aunque existen diferentes tipos de sensores para diferentes aplicaciones, los más comunes y de fácil acceso para nuestro propósito, son los dispositivos que miden la aceleración. Esto se debe a que son fáciles de instalar y configurar, además de que existen en todos lugares (v.g. celulares, juguetes, medios de transporte, etc.). Por tanto, son más baratos que otros sensores utilizados en la detección de daños como: las galgas extensiométricas, amortiguadores, piezoeléctricos, etc.

Para nuestros propósitos, fueron utilizados tres acelerómetros modelo *13203B* (fabricados por *Summit Instruments, Inc.*) de un grado de libertad (*SDOF* por sus siglas en inglés), de tal manera que el eje que mide las aceleraciones queda paralelamente a la dirección del movimiento del actuador que genera las vibraciones (ver Figura 3.3). Además, cada uno de los acelerómetros está instalado en el centro de cada piso de la estructura debidamente calibrado.

- **Encoder magnético**

Otro de los dispositivos instalados en la estructura del edificio y que es utilizado para medir la posición de la misma; es un *encoder* magnético lineal (*LM15*), con una resolución de $50 \mu\text{m}$ (Figura 3.3, *Position Sensor*).

- **Tarjeta de adquisición de datos**

Para la adquisición de datos, se tiene una tarjeta *ServoToGo Modelo II*, la cual es utilizada para adquirir y procesar las señales de aceleración. La tarjeta utiliza un convertidor analógico-digital de 13 *bits*.

- **Sensor activo**

Además, en el último nivel de la estructura; se encuentra instalado un sensor activo *ATMD*, cuya función principal es la de tratar de atenuar las vibraciones lo máximo posible, con el uso de algoritmos de teoría de control (este sensor no fue utilizado para los experimentos, sin embargo, forma parte de la estructura).

- **Actuador para la generación de vibraciones**

Para la generación de vibraciones, un *shaker* electrodinámico modelo *FEEDBACK EHS 160*, es utilizado para proveer la excitación lateral al centro de la base de la estructura, simulando el movimiento provocado por sismos de diferentes magnitudes.

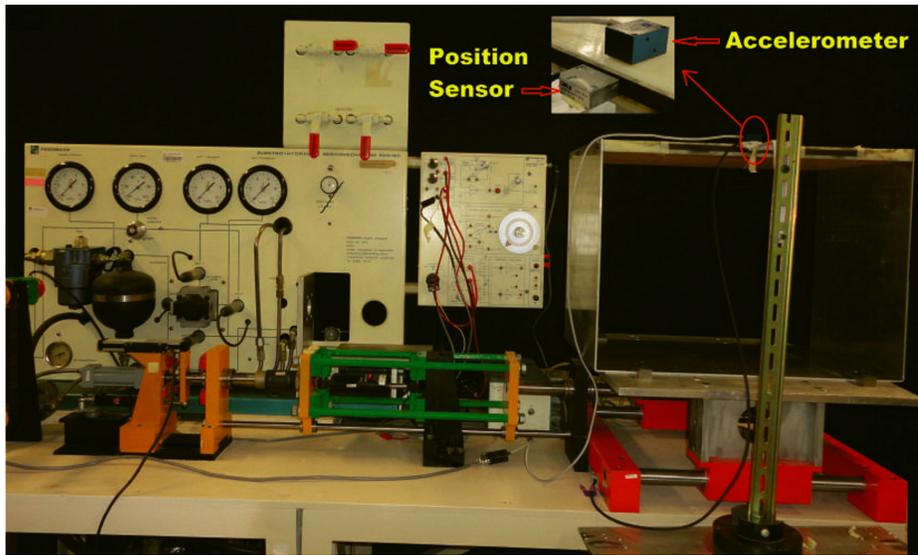


Figura 3.3: Generador de vibraciones y la Base movable

En la Figura 3.4, se muestra el diagrama general del sistema completo con sus componentes principales (para mayores especificaciones de la estructura y el sistema de adquisición de datos, cfr. [45] [46]).

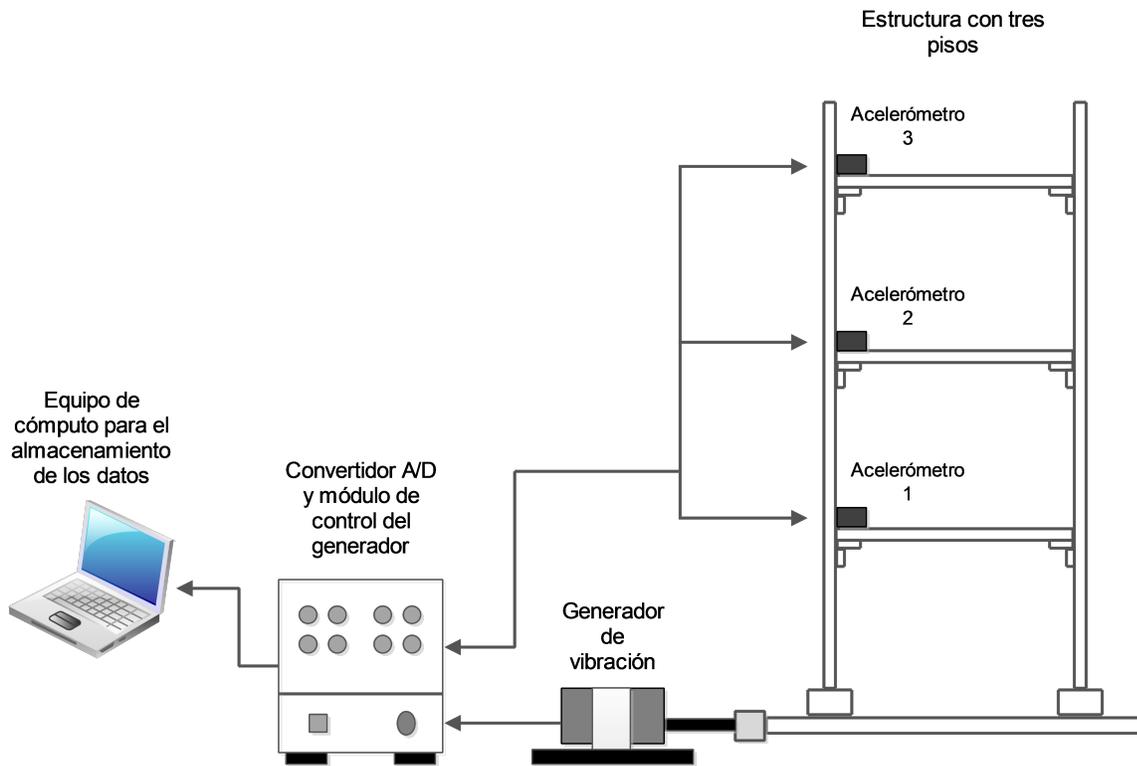


Figura 3.4: Esquema general del sistema

3.1.3. Software

Además de contar con herramientas de *hardware*, también es necesario valernos de herramientas de *software* para el estudio de los datos adquiridos. El *software* que se utilizó para todo el proceso de detección de daños, que va desde; el pre-procesamiento, hasta la validación de los modelos, son: *MATLAB*, así como el *Lenguaje R*. Estas dos herramientas brindan una serie de utilidades (v.g. bibliotecas) que resultan fáciles de utilizar a la hora de entrenar los modelos requeridos para la determinación del daño.

- **Lenguaje R**

R¹ es un lenguaje y entorno de programación para análisis estadístico y gráfico, que proporciona buenas herramientas para el estudio de modelos lineales y no lineales, análisis de series temporales, algoritmos de clasificación, algoritmos de agrupamiento, etc.

- **MATLAB**

MATLAB (*MATrix LABoratory* en inglés) es un lenguaje de alto nivel y de entorno interactivo para cálculo numérico, visualización, y programación. MATLAB cuenta con una amplia gama de herramientas, incluyendo; comunicaciones, procesamiento de vídeo e imagen, sistemas de control, cálculo financiero, etc.

Dentro de estas herramientas, se utilizaron algunas bibliotecas tales como: la *e1071*² para R, la cual contiene funciones para análisis de clases, algoritmos difusos, clasificadores de Bayes, funciones para Máquinas de Soporte Vectorial, etc. Por otro lado, en el ámbito de MATLAB, se utilizaron algunas funciones incluidas en la versión *R2013*³; para el pre-procesamiento de los datos, análisis estadísticos, graficación, entre otras cosas más.

3.2. Pruebas experimentales

Para facilitar la identificación de los experimentos, el edificio de prueba fue dividido en tres pisos, asignándole una etiqueta a cada piso (ver Figura 4.8a), así por ejemplo (de abajo hacia arriba); el primer piso tiene la etiqueta *P1*, el segundo piso tiene la etiqueta *P2* y el último piso tiene la etiqueta *P3*. De igual manera, cada acelerómetro colocado en cada piso, tiene su propia etiqueta; *A1*, *A2* y *A3* respectivamente (ver Figura 3.4).

Para las pruebas experimentales, se diseñaron dos escenarios (cada escenario con dos experimentos); i.e. en el escenario 1, se introdujo el daño en *P2* para el primer experimento,

¹<http://www.r-project.org/>

²<http://cran.r-project.org/web/packages/e1071/index.html>

³<http://www.mathworks.com/products/matlab/>

dejando así el segundo experimento sin la presencia de daño (sin daño en P1, P2 y P3). Para el escenario 2, se repitieron los experimentos 1 y 2, con el fin de respaldar los experimentos y obtener mayor número de datos para el análisis.

Para simular el daño físico en la estructura, se les disminuyó el torque a los pernos que sostienen el plato en el piso central (P2), ya que es en donde usualmente comienzan a sufrir daños los edificios de tamaño real. Cada prueba tuvo una duración de *40 segundos* con un muestreo de datos de *1 mili-segundo, i.e. 1 KHz de frecuencia*. Además, la señal de entrada fue ajustada a la del sismo de *Loma Prieta Costa-Este del 18 Octubre de 1989*, con el objetivo de obtener escenarios realistas de las vibraciones. Finalmente, al término de las pruebas, los datos son almacenados en una base de datos que guarda los registros de los acelerómetros en cuestión. En la Tabla 3.1 y Tabla 3.2, se muestran parte de los datos obtenidos para el escenario 1, con los datos con daño y sin daño en P2 respectivamente.

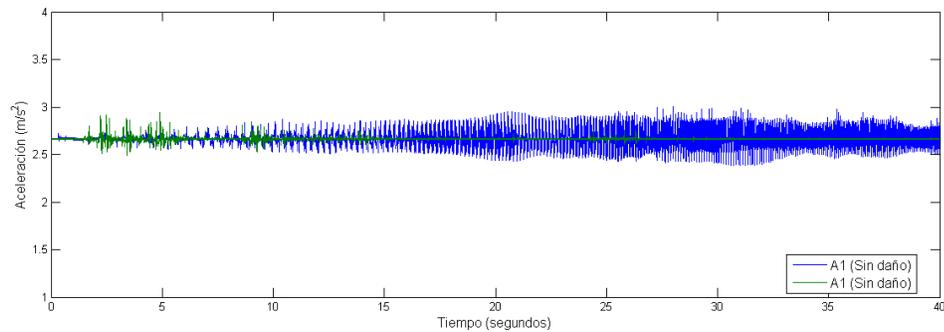
A1	A2	A3
2.66667	2.81807	2.72039
2.66667	2.81319	2.72039
2.66178	2.81807	2.71551
2.66178	2.81319	2.71551
2.66667	2.81319	2.72039
2.67155	2.81807	2.72039
2.66178	2.81807	2.72039
2.66667	2.81319	2.71551
2.66667	2.81807	2.71551
2.67155	2.81807	2.72039
2.66178	2.81319	2.72039
2.66178	2.81319	2.71551
2.66178	2.81319	2.71551
2.66667	2.81319	2.72527
2.66667	2.81807	2.71551
⋮	⋮	⋮

Tabla 3.1: Valores de las aceleraciones con la presencia de daño en A2 para el escenario 1

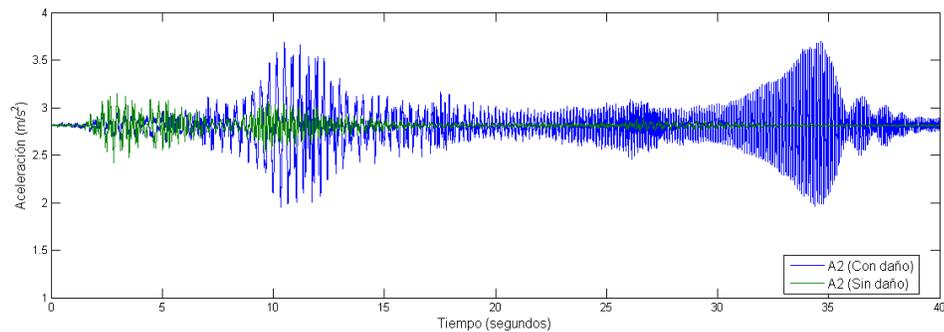
A1	A2	A3
2.66178	2.80830	2.70085
2.66667	2.81319	2.70574
2.66667	2.81807	2.71551
2.66667	2.81319	2.71062
2.66178	2.80830	2.70574
2.66667	2.81319	2.70574
2.66667	2.81319	2.71551
2.66178	2.81319	2.71551
2.66178	2.80830	2.71062
2.66667	2.80830	2.70574
2.67155	2.81319	2.71551
2.66178	2.81319	2.71551
2.66178	2.81319	2.71062
2.66178	2.81319	2.70574
2.66178	2.81319	2.70574
⋮	⋮	⋮

Tabla 3.2: Valores de las aceleraciones sin la presencia de daño para el escenario 1

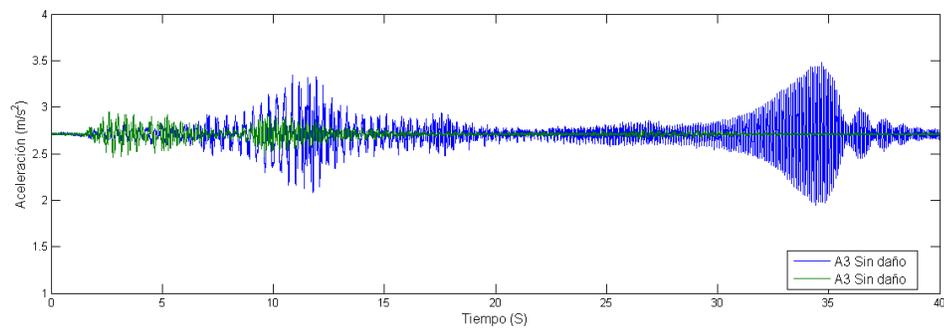
En la Figura 3.5, se muestran los resultados obtenidos de las aceleraciones⁴ del escenario 1 para cada uno de los acelerómetros, con daño y sin daño en P2. De igual manera, en la Figura 3.6, se muestra la otra comparación del escenario 2 de la estructura. N.b. que a simple vista, cuando existe el daño en P2, las aceleraciones comienzan en un nivel mayor, con mayores amplitudes en la señal; sin embargo, no es prueba fehaciente de la presencia del mismo, ya que pueden influir muchos factores, tales como: el ruido, la colocación de los acelerómetros, el *offset*, etc.



(a)



(b)



(c)

Figura 3.5: Aceleraciones para ambos estados de los tres acelerómetros del escenario 1

⁴Las aceleraciones están medidas en m/s^2 .

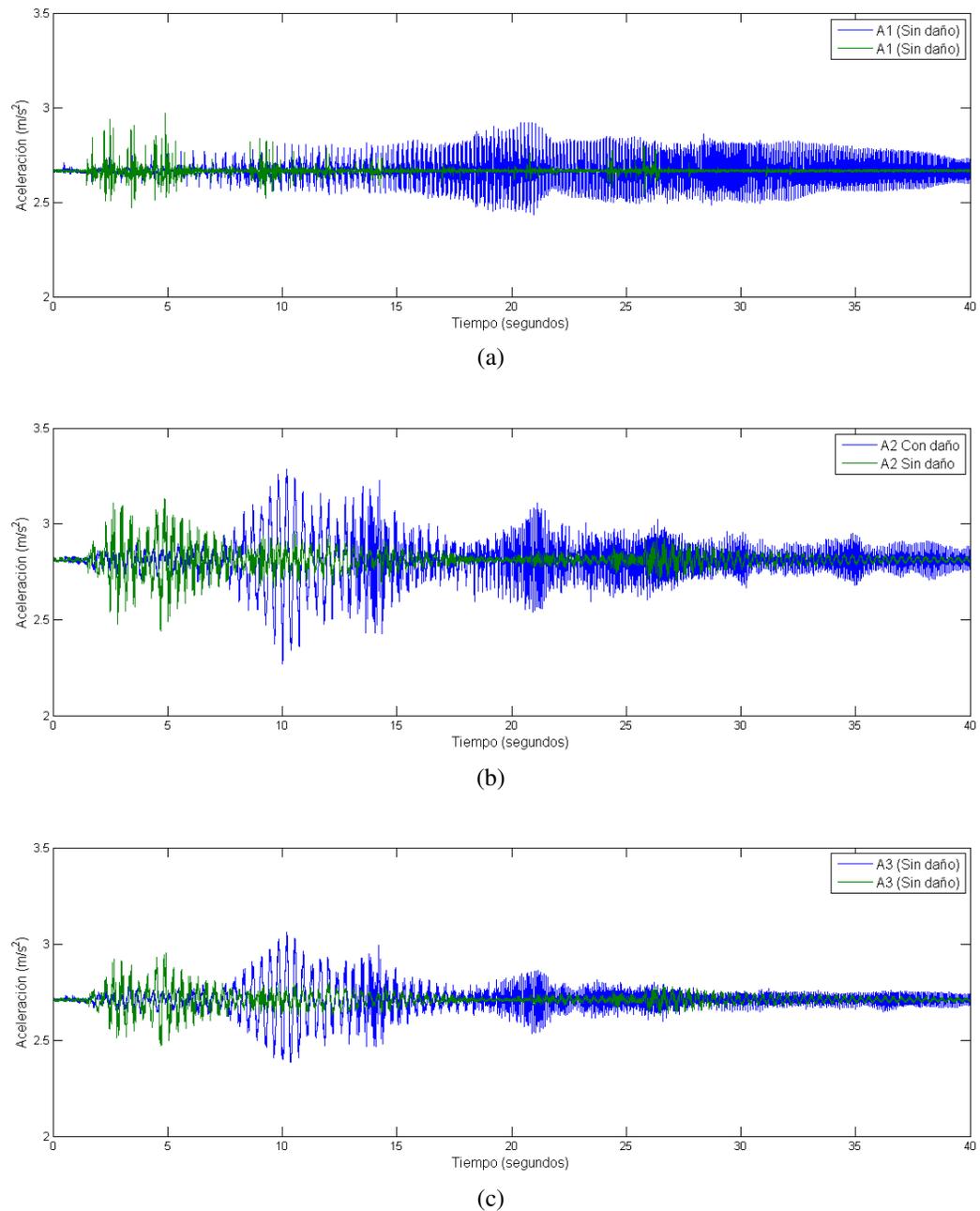


Figura 3.6: Aceleraciones para ambos estados de los tres acelerómetros del escenario 2

Una vez obtenidos los valores de las aceleraciones, se obtuvieron los datos para las velocidades y las posiciones de ambos escenarios.

3.3. Análisis estadístico

Comúnmente las características dinámicas de un sistema proveen información que es estudiada para construir modelos de detección de daño. La aceleración, velocidad, y posición; son los parámetros que fueron utilizados en el proceso de este análisis. Por tanto, un análisis estadístico básico de estos parámetros puede arrojar valiosa información a cerca del comportamiento de los datos y brindar un indicio de valores atípicos que podrían representar la presencia del daño. Algunos de los conceptos estadísticos a los que se hace referencia son: la *media aritmética*, la *desviación estándar*, el *sesgo* y la *curtosis*.

Para este análisis estadístico, se tomó en cuenta el teorema del límite central, el cual en resumidas palabras enuncia que: cuando se analizan datos medidos por una variable cuantitativa continua, las pruebas estadísticas de estimación y contraste empleadas, se basan en suponer que se ha obtenido una muestra aleatoria de una distribución de probabilidad de tipo Normal o de *Gauss*. Y dado que la señal de entrada del *shaker* se asume como normal para condiciones normales (sin daño), entonces podemos hacer una comparativa con condiciones en donde el daño está presente.

3.3.1. Momentos estadísticos

Como ya se mencionó, las respuestas provenientes del sistema pueden representar una comparativa de reconocimiento de daño entre estados. Los primeros cuatro momentos estadísticos (media, desviación estándar, sesgo, y curtosis) son frecuentemente calculados, examinando los datos sin procesar, i.e. con el fin de explorar para poder sugerir indicios de patrones propios de los datos (v.g. valores atípicos y valores nulos).

Media y Varianza

Uno de los parámetros más populares es la media, (también llamada promedio) la cual representa el promedio aritmético de todos los datos de una variable. La media puede ser representada como en la Ecuación 3.1:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n a_i, \quad (3.1)$$

donde a_i es la respuesta de los datos, con $i = 1, 2, \dots, m$, y n es el número de ejemplos del conjunto de valores. Entonces; la varianza mide la media de las desviaciones cuadráticas de una variable aleatoria. En simples palabras; mide la variabilidad de la variable, y puede

ser formulada como en la Ecuación 3.2:

$$s^2 = \frac{1}{n} \sum_{i=1}^n (a_i - \bar{x})^2. \quad (3.2)$$

Desviación estándar

La desviación estándar (o desviación típica) brinda una idea de cuán cercano está el conjunto de datos al valor promedio de la muestra. En otras palabras; brinda un promedio de fluctuación de los datos respecto al valor central, y se representa como la raíz cuadrada positiva de s , tal y como en la Ecuación 3.3:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - \bar{x})^2} = \sqrt{s^2}. \quad (3.3)$$

Sesgo

El tercer momento estadístico es el sesgo, el cual describe la asimetría de la distribución de probabilidad basado en muestras aleatorias, donde el sesgo de una distribución Normal es cero. Por otro lado, si el valor es negativo, la curva de probabilidades tiende a desviarse a la izquierda, caso contrario, cuando el valor es positivo. La Ecuación que representa a este momento estadístico está dado por:

$$\gamma = \frac{\frac{1}{n} \sum_{i=1}^n (a_i - \bar{x})^3}{s^3}. \quad (3.4)$$

Curtosis

Finalmente, el cuarto momento estadístico es la curtosis k , y es definida en la Ecuación 3.5:

$$k = \frac{\frac{1}{n} \sum_{i=1}^n (a_i - \bar{x})^4}{s^4}. \quad (3.5)$$

Para una distribución Normal, la curtosis es igual a tres. Cuando los valores para la curtosis son mayores que tres, significa que: existe una alta concentración de datos cerca de la media, y la curva presenta un mayor apuntalamiento con colas más anchas, en contraposición que si el valor es menor que tres.

Un indicio del cual la estructura podría presentar daño, se puede observar con la media y la desviación estándar. En las Figuras 3.7 y 3.8, se muestran las medias y las desviaciones estándar de cada uno de los acelerómetros para el escenario 1. Se puede notar que los valores de la media son muy pequeños entre sí, por tanto, en la Figura 3.9a y Figura 3.9b se realiza un *zoom* para hacer una comparación adecuada. En las figuras, se nota que la media del estado donde se sufre daño es mayor (ver Figura 3.9a). Sin embargo, n.b. que los valores entre un estado y otro son mínimos como para obtener una conclusión concreta.

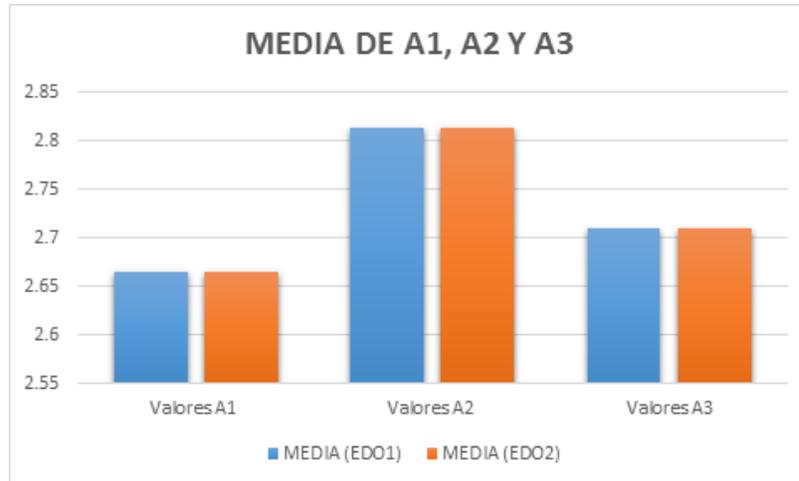


Figura 3.7: Medias para A1, A2 y A3 del escenario 1

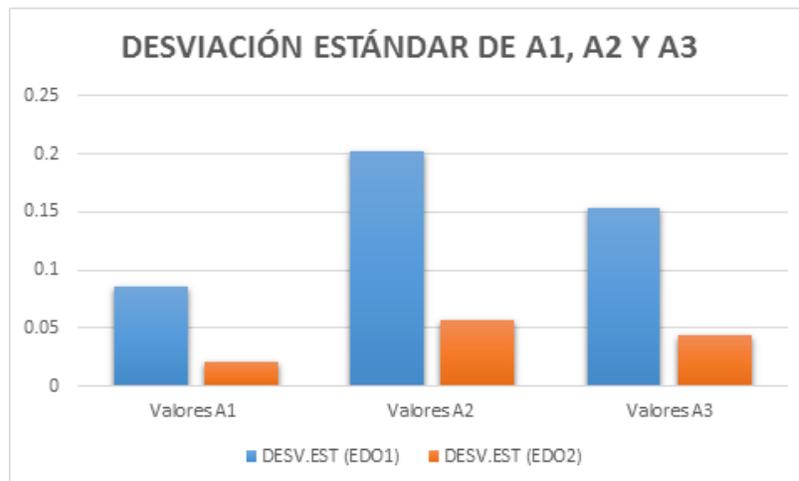


Figura 3.8: Desviación estándar de A1, A2 y A3 del escenario 1

Así mismo, se muestran las medidas centrales para el escenario 2 (ver Figura 3.10 y Figura 3.11), n.b. que en las Figuras 3.12a y 3.12b, en este caso la media decrece, mientras que la desviación estándar aumenta cuando se presenta el daño (en el EDO1), esto se debe a que los valores tienden a dispersarse de la media cuando las aceleraciones son irregulares, al mismo tiempo que aumenta su desviación estándar.

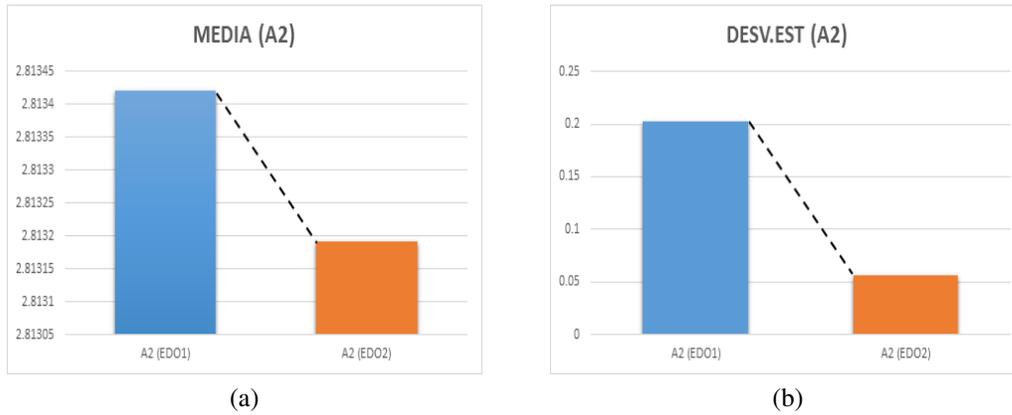


Figura 3.9: Comparación de la media y la desviación estándar del escenario 1, para A2

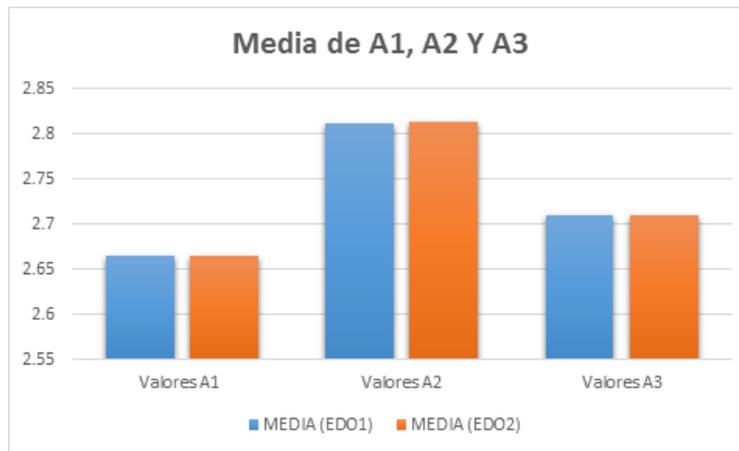


Figura 3.10: Medias para A1, A2 y A3 del escenario 2

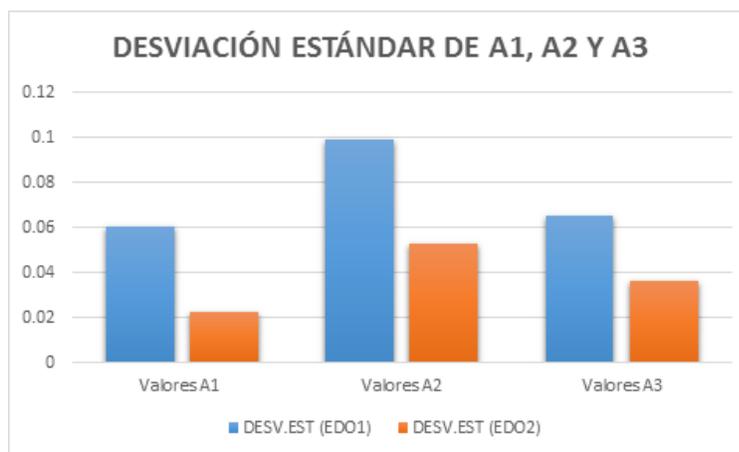


Figura 3.11: Desviación estándar de A1, A2 y A3 del escenario 2

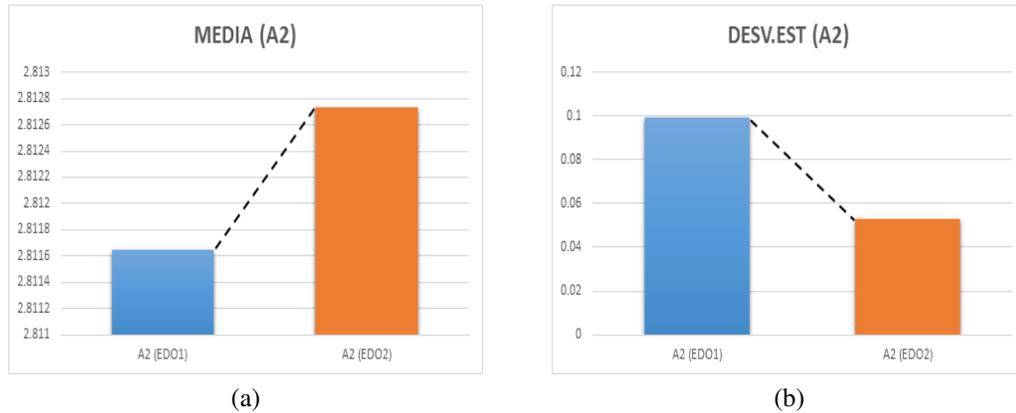


Figura 3.12: Comparación de la media y la desviación estándar del escenario 2, para A2

Las Tablas 3.4 y 3.3, muestran los diferentes valores de las aceleraciones para A2, así como los momentos estadísticos para el escenario 1 y el escenario 2 (cada escenario tiene dos estados para A1, A2 y A3).

Estado	Media	Varianza	Desv.Std	Sesgo	Curtosis
A1 - Sin daño	2.6654	0.0004	0.0208	2.6386	4.4674
A1 - Sin daño	2.6653	0.0074	0.0860	0.1996	0.2284
A2 - Con daño	2.8131	0.0032	0.0568	-0.0079	7.1907
A2 - Sin daño	2.8134	0.0410	0.2026	0.0315	3.6476
A3 - Sin daño	2.7103	0.0018	0.0434	-0.0685	6.3047
A3 - Sin daño	2.7098	0.0233	0.1527	-0.0116	4.9384

Tabla 3.3: Características estadísticas para ambos estados del escenario 1

Estado	Media	Varianza	Desv.Std	Sesgo	Curtosis
A1 - Sin daño	2.6648	0.0005	0.0224	2.2862	0.4110
A1 - Sin daño	2.6648	0.0036	0.0607	0.1196	0.8614
A2 - Con daño	2.8127	0.0027	0.0526	-0.1673	7.5884
A2 - Sin daño	2.8116	0.0098	0.0990	-0.0700	4.2407
A3 - Sin daño	2.7099	0.0013	0.0362	-0.1214	4.8448
A3 - Sin daño	2.7101	0.0042	0.0654	-0.0283	5.53771

Tabla 3.4: Características estadísticas para ambos estados del escenario 2

Como ya se pudo observar, la media y la desviación estándar no siempre ofrecen valores confiables, esto se debe a que son susceptibles a valores atípicos. Sin embargo, el sesgo y la curtosis son medidas más confiables que también pueden ser indicadores de daño en la estructura cuando los estados son comparados. Se puede notar que cuando el daño ocurre, los valores del sesgo cambian de cero, sin embargo, son valores muy pequeños para hacer una comparación. De igual modo, cuando el daño ocurre, los valores de la curtosis son mayores a tres, lo que indica una desviación de los parámetros normales (estado sin daño).

3.3.2. Distribuciones de Probabilidad

Como ya mencionamos, el estado del escenario sin daño se supone como una distribución Normal. Basado en esa aseveración, las gráficas de distribución de probabilidad son utilizadas para verificar entre una distribución Normal (valores sin daño) y cualquier otro estado en los que puede existir el daño; verificando si se ajusta a una distribución Normal o no. Una gráfica de probabilidad de tipo Normal cumple con dos condiciones: que la media sea cero y la desviación estándar uno (para valores normalizados). Si los datos se desvían demasiado de estos dos parámetros, entonces ha de suponerse que provienen de otro tipo de distribución.

Distribución *Kernel*

La distribución *Kernel* es una representación no paramétrica de la Función de Densidad de Probabilidad (*PDF* por sus siglas en inglés) de una variable aleatoria (en nuestro caso se habla de aceleraciones). Esta distribución tiene una gran ventaja, la cual sirve para utilizarla cuando no se sabe el tipo de distribución proveniente de la muestra (v.g. *Normal*, *Weibull*, *Student*, etc). Esta distribución se define por una función de suavizado y un valor de ancho de banda (ventana) que controla la suavidad de la curva de densidad resultante, en donde la función de suavizado define la forma de la curva usada para generar la *PDF*. Esta distribución *Kernel* es parecida a un histograma, en cuanto a la representación de la distribución de probabilidad del conjunto de datos analizado.

El estimador de la distribución *Kernel* se formula como en la Ecuación 3.6;

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right); \quad -\infty < x < \infty, \quad (3.6)$$

donde n es el tamaño de la muestra, $K(\cdot)$ es la función de suavizado (*kernel*) y h es el ancho de la ventana.

Función de Distribución Acumulada

Una Función de Distribución Acumulada (*FDA*) describe la probabilidad de que una variable aleatoria real X sujeta a cierta ley de distribución de probabilidad, se sitúe en la zona de valores menores o iguales a x . La función se define como en la Ecuación 3.7:

$$F(x) = P\{w : -\infty \leq X(w) \leq x\} = P(X \leq x) = \int_{-\infty}^x f(x) dx. \quad (3.7)$$

Función de Distribución de Probabilidad

Para una variable continua hay infinitos valores posibles de la variable, y entre cada dos de ellos se pueden definir infinitos valores más. En estas condiciones no es posible deducir la probabilidad de un valor puntual de la variable; tal y como se puede hacer en el caso de variables discretas, sin embargo, es posible calcular la probabilidad acumulada hasta un cierto valor (Función de Distribución de Probabilidad). Además, es posible analizar cómo cambia la probabilidad acumulada en cada punto. Para el caso de valores continuos, la distribución de probabilidad es la integral de la función de densidad (curva de densidad). Por tanto, se tiene definida en la Ecuación 3.8:

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(x)dx, \quad (3.8)$$

donde X es una variable continua, donde una distribución de probabilidad (FDP , Función de Densidad de Probabilidad) de X es una $F(x)$ tal que cualesquiera valores desde a hasta b siendo $a \leq b$, tenemos que:

$$P(a \leq X \leq b) = \int_a^b f(x) dx. \quad (3.9)$$

La distribución *Kernel*, la Densidad de Distribución Acumulada y la Función de Probabilidad del escenario 1 es mostrado en las Figuras 3.13, 3.14 y 3.15. De igual modo, se muestran las Figuras 3.16, 3.17 y 3.18 para el escenario 2. Las figuras muestran las comparaciones entre estados, uno sin daño y el otro con la presencia de daño en P2 de los datos normalizados. Se puede observar por ejemplo; que cuando existe daño en la estructura en cierto nivel, las tendencias centrales pueden desajustarse debido al incremento de las aceleraciones disperejas.

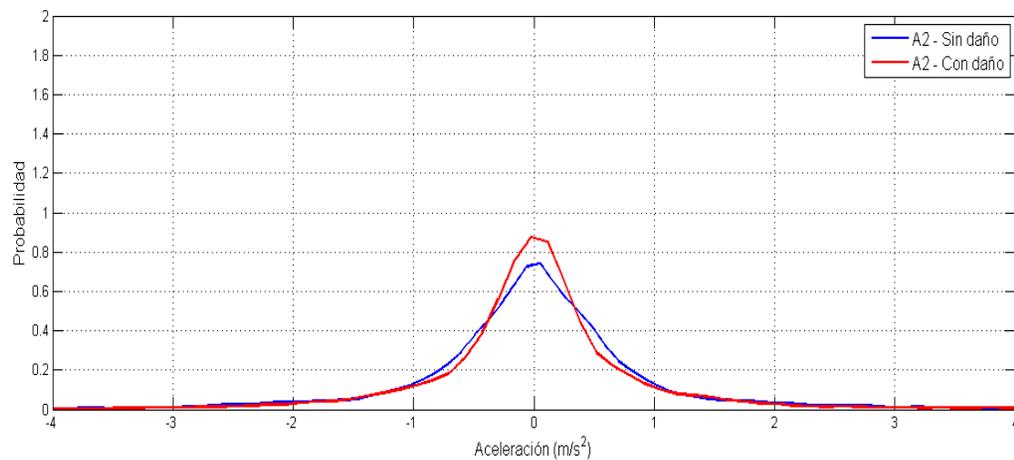


Figura 3.13: Función Kernel para A2 del escenario 1

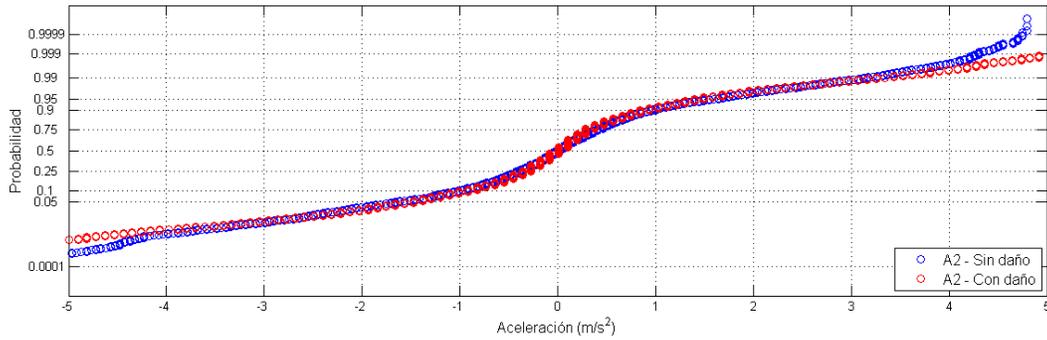


Figura 3.14: Función de probabilidad para A2 del escenario 1

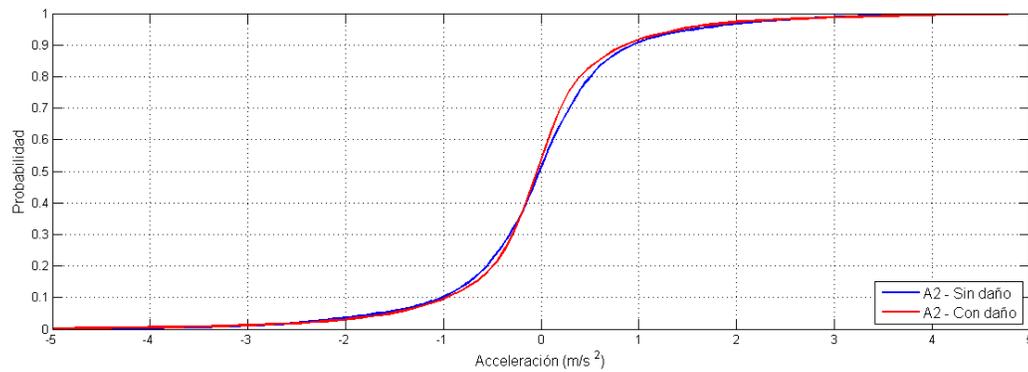


Figura 3.15: Probabilidad acumulativa para A2 del escenario 1

Como ya se habrá notado, las respuestas estructurales (aceleración) y sus características estadísticas involucran algunas diferencias cuando ocurre el daño. Sin embargo, la información del daño es difícil de discernir cuando no existe el daño. Por tanto, es necesario utilizar métodos más robustos que indiquen verdaderamente la existencia de daño de manera precisa. Como veremos en el Capítulo 4 y Capítulo 5, las técnicas aplicadas logran con éxito determinar y localizar la presencia de daño con mucha exactitud.

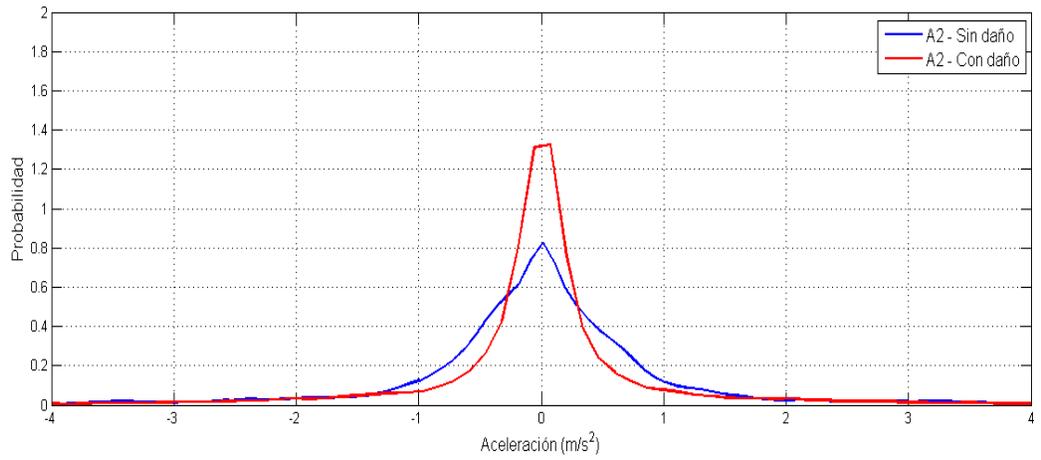


Figura 3.16: Función Kernel para A2 del escenario 2

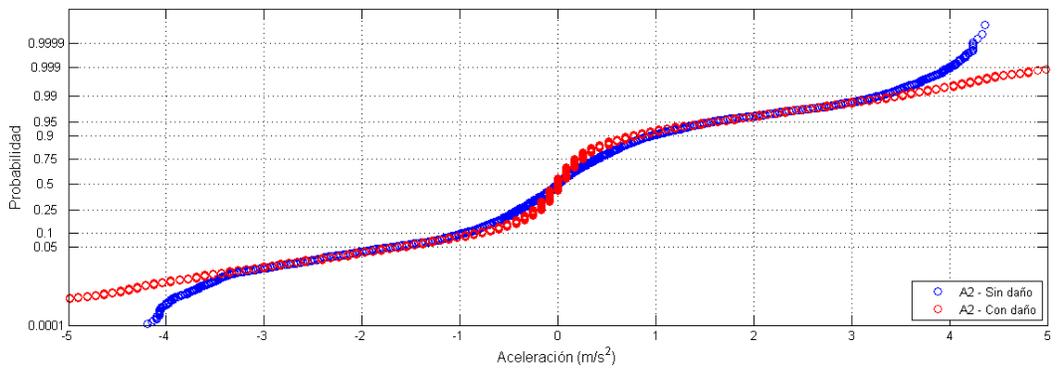


Figura 3.17: Función de probabilidad para A2 del escenario 2

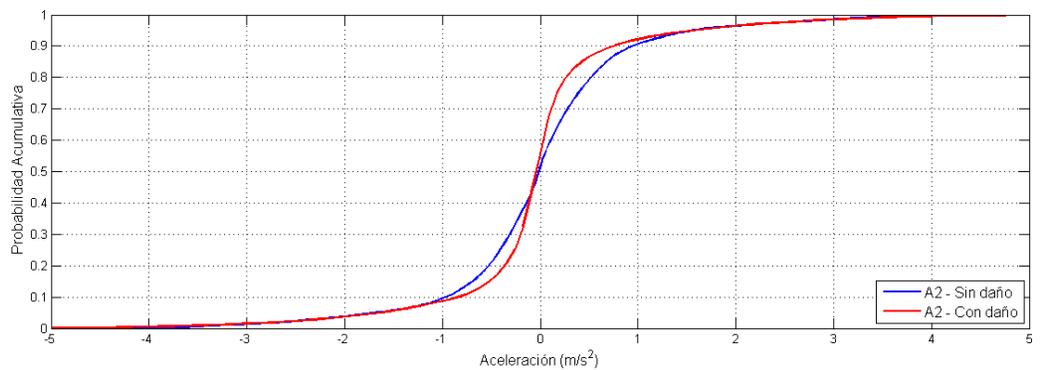


Figura 3.18: Probabilidad acumulativa para A2 del escenario 2

3.4. Pre-procesamiento de los datos

Una vez almacenados los datos de los experimentos y realizado una “exploración” de los mismos, se prosigue a pre-procesar los datos. Para este propósito se utilizó una transformación y un filtro, como se detalla a continuación:

Transformación y filtración de los datos en el dominio de la frecuencia

Aunque en las pruebas realizadas, los valores de los acelerómetros fueron medidos en el dominio del tiempo, es posible transformarlos para tratarlos de diferente manera. Por tanto, una de las etapas de pre-procesamiento en este trabajo de tesis, consiste en transformar los datos obtenidos al dominio de la frecuencia. Esto se debe a que es más fácil trabajar en escalas de frecuencia (en contraposición de escalas de tiempo o espacio), ya que una de sus principales ventajas es que no se tienen valores negativos, por el contrario; todos los valores comienzan de una escala de 0. Para lograr dicho objetivo se utilizó una técnica muy conocida en el procesamiento de señales, denominada: Transformada Rápida de *Fourier* o también conocida como *FFT (Fast Fourier Transform)* [47].

El análisis de *Fourier* de una función periódica se refiere a la extracción de series de senos y cosenos, que cuando se superponen, reproducen la función original; y se puede expresar como series de *Fourier*. Dicho lo anterior, la Transformada Rápida de *Fourier (TRF)* es un método matemático para la transformación de una función del tiempo, en una función de la frecuencia. Cabe mencionar que; la TRF es un algoritmo que agiliza el cálculo de la Transformada Discreta de *Fourier (TDF)*, y es fundamentalmente importante, ya que elimina gran parte de los cálculos repetitivos de la TDF.

Entonces; la TDF transforma una señal en el espectro de la frecuencia y es la versión discreta de la Transformada de *Fourier*, representada por la Ecuación 3.10, dada una señal analógica.

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt, \quad (3.10)$$

en donde, dada una variable continua t , ha sido transformada en una variable discreta nT_s , siendo T_s el periodo de muestreo. Dicho lo anterior, se tiene: la Transformada Discreta de *Fourier* (ver Ecuación 3.11) y la Transformada Inversa de *Fourier* (ver Ecuación 3.12):

$$X(k) = \sum_{i=0}^{N-1} x(n) W^{nk}, \quad (3.11)$$

$$x(n) = \frac{1}{N} \sum_{i=0}^{N-1} X(k) W^{-nk}, \quad (3.12)$$

en donde, $k, n = 1, 2, \dots, N - 1$, $X(k)$ es la señal; y $x(n)$ es su densidad espectral. Así mismo, las constantes W son definidas como en la Ecuación 3.13:

$$W = e^{-j2\pi/N}. \quad (3.13)$$

La evaluación directa de la Ecuación 3.11, requiere $O(n^2)$ operaciones aritméticas. Mediante un algoritmo FFT se puede obtener el mismo resultado con sólo $O(n \log n)$ operaciones. La FFT aprovecha la periodicidad y simetría del factor W para el cálculo de la TDF de N puntos en transformadas más pequeñas. Dicha periodicidad implica:

$$W^k = W^{k+N}, \quad (3.14)$$

y su simétrica:

$$W^k = -W^{k+N/2}, \quad (3.15)$$

en donde se hace una descomposición de la transformada en otras más simples, hasta llegar a transformadas de dos elementos, donde k puede tomar los valores 0 y 1 para obtener los valores de X , los cuales son la secuencia de salida de la TDF.

Filtro

Es evidente que a la hora de sensar, existen interferencias que contaminan los datos como: la temperatura, la humedad, el propio error de medición que tienen los sensores (*offset*), la calibración, etc. Por tanto, es necesario utilizar filtros para atenuar o amplificar algunas frecuencias.

Podemos definir un filtro como aquel elemento que discrimina una determinada frecuencia o gama de frecuencias de una señal eléctrica que pasa a través de él, pudiendo modificar tanto su amplitud como su fase. Cabe mencionar que éstos filtros pueden implementarse tanto de manera física en los sensores, así como de manera digital en la computadora con funciones matemáticas.

Resultados del FFT y el Filtro

Para nuestro propósito; los datos fueron transformados al dominio de la frecuencia con la Transformada Rápida de *Fourier*. El proceso anterior, se realizó con *MATLAB*, en donde los datos de las aceleraciones, velocidades y posiciones fueron transformados con la función *fft*⁵, en la cual es necesario ajustar algunos parámetros como: el ángulo de fase, el desfase, el tamaño de la fase, etc.

⁵<http://www.mathworks.com/help/matlab/ref/fft.html>

Después, para la aplicación del filtro, fue necesario utilizar otra de las funciones propias de *MATLAB*. Esta función se denomina: (*detrend*⁶), la cual es utilizada para quitar el ruido de la señal. El resultado de realizar el FFT y el filtro, se puede observar en las Figuras 3.19, 3.20, en donde se muestran las transformaciones para A1, A2 y A3 de las aceleraciones, tanto para el estado 1 como para el estado 2, propios del escenario 1. Después, se hace una comparación en la Figura 3.21, en donde se muestran las frecuencias obtenidas tanto del estado 1 como del estado 2, ambos para A2, donde se compara el estado con daño y el estado sin daño. Las transformaciones y la aplicación del filtro también fueron realizados para las velocidades y las posiciones en cada uno de los escenarios.

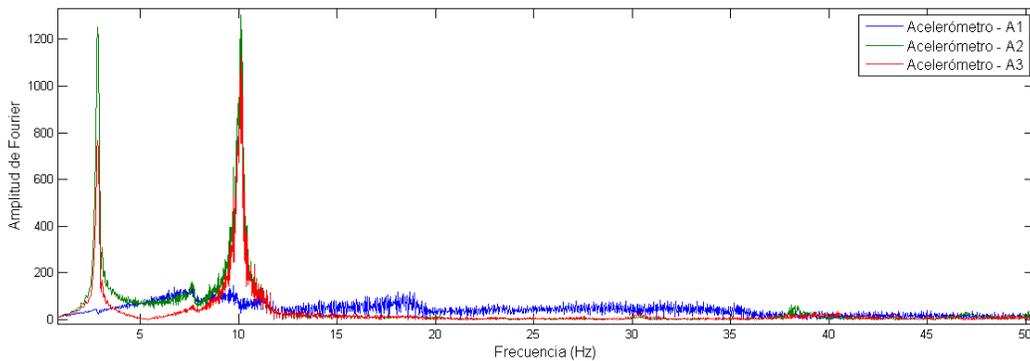


Figura 3.19: Transformada Rápida de Fourier para A1, A2 y A3, estado 1 del escenario 1

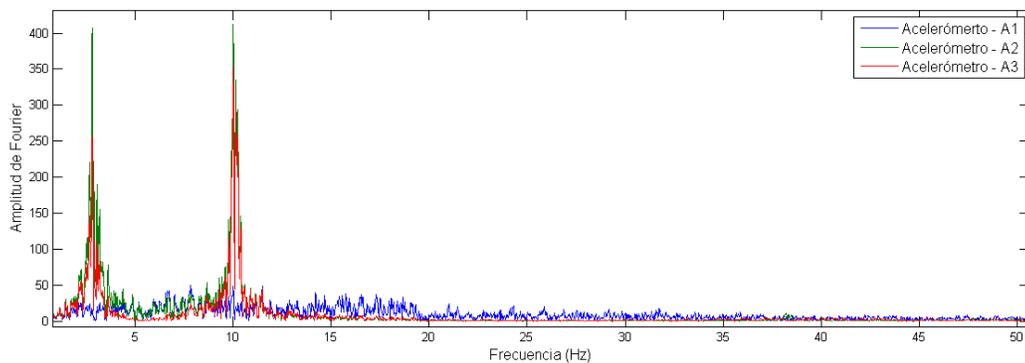


Figura 3.20: Transformada Rápida de Fourier para A1, A2 y A3, estado 2 del escenario 1

Así mismo, se obtuvieron las transformaciones para el segundo escenario, obteniendo los siguientes resultados, como se ve en las Figuras 3.22, 3.23 y 3.24:

⁶<http://www.mathworks.com/help/matlab/ref/detrend.html>

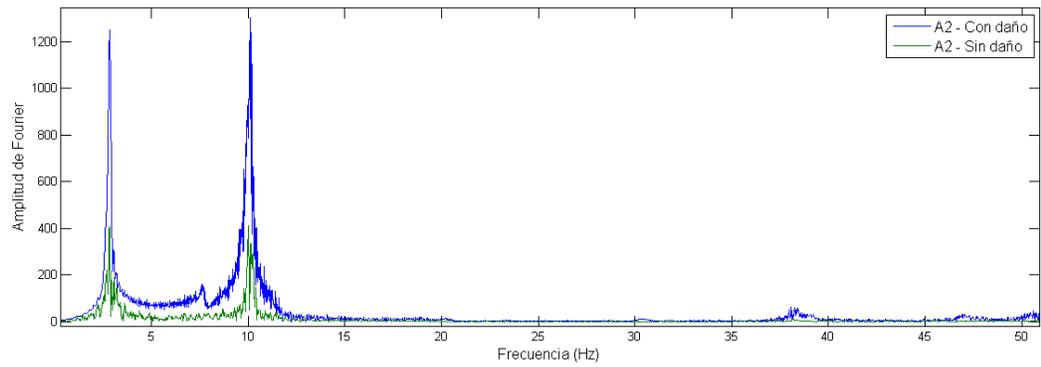


Figura 3.21: Transformada Rápida de Fourier para A2 con daño y A2 sin daño del escenario 1

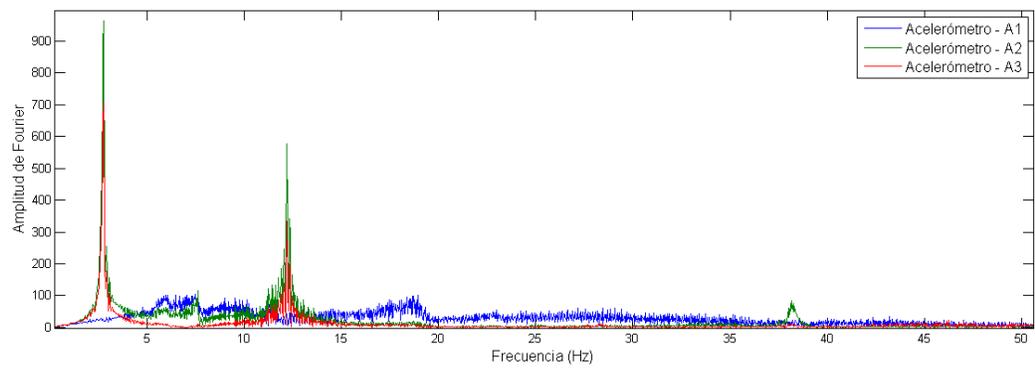


Figura 3.22: Transformada Rápida de Fourier para A1, A2 y A3, estado 1 del escenario 2

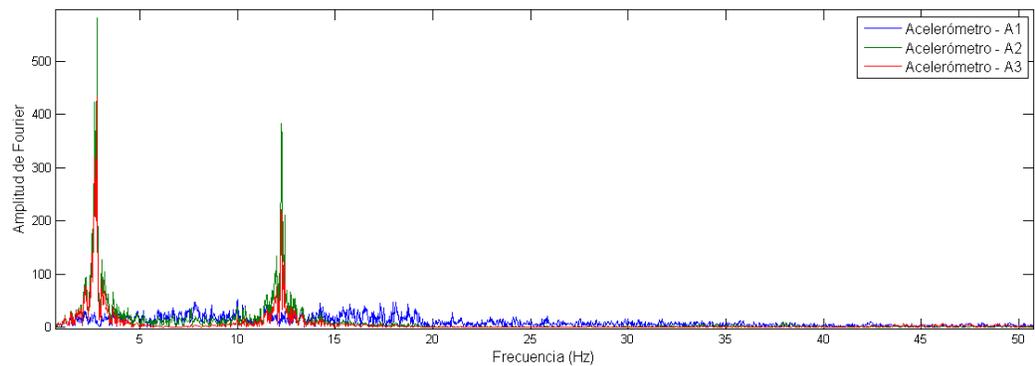


Figura 3.23: Transformada Rápida de Fourier para A1, A2 y A3, estado 2 del escenario 2

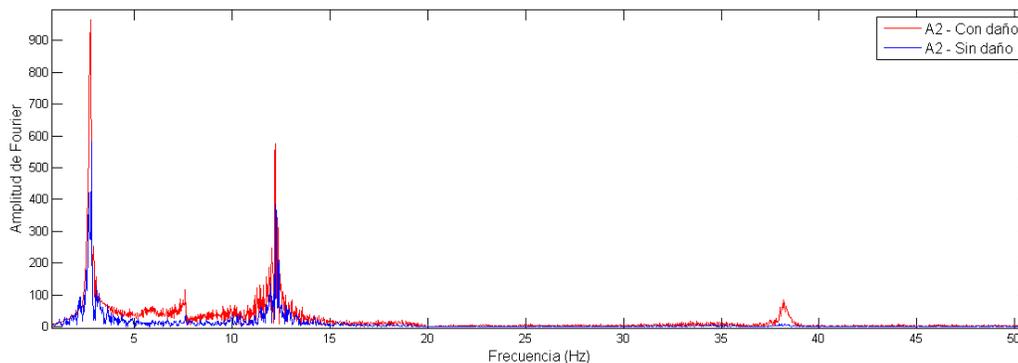


Figura 3.24: Transformada Rápida de Fourier para A2 con daño y A2 sin daño del escenario 2

3.5. Conclusión

En este capítulo se describió el sistema para la detección de daños, el cual está conformado por elementos de *hardware* y *software*. De igual modo, se describieron las características de los experimentos realizados, en donde se explica cómo fueron divididos los datos, la frecuencia de muestreo, la señal de entrada, etc.

En otra sección del capítulo, se describió el análisis estadístico básico que fue realizado, con el fin de encontrar ciertos patrones en los datos. Esta etapa de la Minería de datos es la que comprende la parte del *análisis de las propiedades de los datos*, en especial aquellos que se construyen con momentos estadísticos, distribuciones de probabilidad, histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos). Sin embargo, es necesario tomar en cuenta que este tipo de análisis únicamente es una etapa básica que ayuda a entender ciertos patrones que son de ayuda para aplicarlos en las siguientes etapas de la Minería de Datos. Una de las limitaciones que es ya conocida, fue la media aritmética, la cual se trata de una medida muy sensible a los valores extremos; ya que valores muy grandes tienden a aumentarla mientras que valores muy pequeños tienden a reducirla, lo que implica que puede dejar de ser representativa de la población. De los resultados obtenidos, se nota que los valores son ínfimos para realmente aseverar que existe daño en la estructura. Por otra parte, la *curtosis* parece ser una medida más confiable ya que no se ve afectada por valores muy grandes o muy pequeños.

Luego, en el análisis de las distribuciones de probabilidad, las distribuciones fueron comparadas entre sí (con la presencia de daño y sin la presencia de daño), en donde de los resultados obtenidos se observaron las tendencias de desajuste con respecto a la Normal.

Por último, se realizó la transformación del conjunto de datos de entrada con la FFT y con la aplicación de un filtro; con el objetivo de preparar los datos para aplicar las técnicas de Minería de Datos. Así mismo, en las gráficas al igual que las distribuciones de probabilidad, se observan valores con altas frecuencias, cuando se presenta el daño.

Capítulo 4

Máquinas de Soporte Vectorial para la detección de daños

En el Capítulo 3, se obtuvieron las características estadísticas de los conjuntos de datos provenientes de las aceleraciones, para tratar de inferir la presencia de daño. Así mismo, se pre-procesaron los datos de acuerdo con el paradigma de la Minería de Datos, con el fin de utilizar las técnicas de aprendizaje supervisado propuestos en el Capítulo 1. Ahora, en este capítulo, se describe la teoría y el proceso de clasificación para la detección de daños en el edificio, utilizando las *Máquinas de Soporte Vectorial*.

4.1. Clasificación del daño

Antes de comenzar a definir la propuesta para la clasificación del daño con el uso de las Máquinas de Soporte Vectorial, comenzaremos a definir el concepto de clasificación.

Clasificación

La clasificación es la tarea de asignar objetos a una o más categorías, de acuerdo a sus características cualitativas y/o cuantitativas propias de cada objeto [27]. Otra definición dice: la clasificación es la tarea de encontrar una función f que mapee cada atributo del conjunto x de los datos, a una de las clases predefinidas C_i [30].

Formalmente, tenemos que; el objetivo principal de la clasificación, es tomar un vector de entrada x y asignar cada elemento a una de las N clases discretas C_i , donde: $i = 1, 2, \dots, N$.

La clasificación de los datos es una tarea común dentro del aprendizaje automático, la cual tiene como fin decidir a qué clase pertenecerá un nuevo objeto, mediante un modelo

de entrenamiento. Esta necesidad de clasificar algún evento, objeto o persona, es de gran importancia, ya que ayuda a simplificar procesos de conocimiento para resolver problemas que se presentan en distintas áreas. En nuestro caso en particular, como ya se ha mencionado en capítulos anteriores, se necesita de la clasificación entre dos posibles estados del edificio (con daño y sin daño).

Clasificador para la detección de daños

Un clasificador o técnica de clasificación, es un enfoque sistemático para construir modelos de un conjunto de datos de entrenamiento de entrada. Cada técnica emplea un algoritmo de aprendizaje para construir un modelo que se ajuste mejor a la relación entre el conjunto de atributos y la etiqueta de la clase de los datos de entrada. Además, un objetivo clave del algoritmo de aprendizaje es construir modelos con buena capacidad de generalización; es decir, modelos que predigan con exactitud las etiquetas de la clase a la que pertenecen los objetos hasta ahora desconocidos. Dicho lo anterior, en la Figura 4.1, se muestra el enfoque general propuesto, que fue utilizado para resolver nuestro problema de clasificación de daño en el edificio, con las Máquinas de Soporte Vectorial.

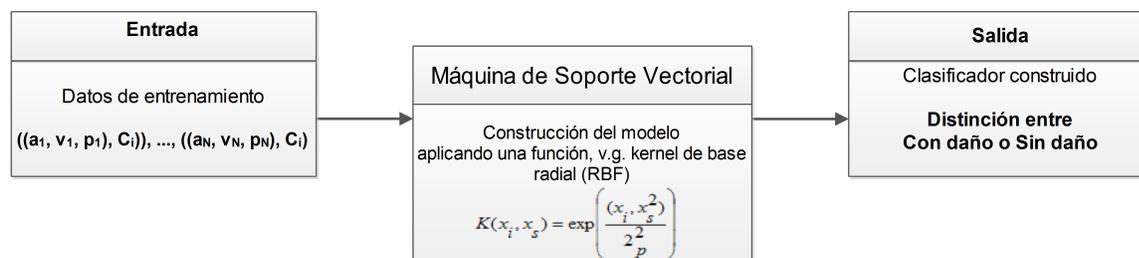


Figura 4.1: Enfoque general para la clasificación del daño con Máquinas de Soporte Vectorial

De acuerdo con la Figura 4.1, se tiene como datos de entrenamiento: las **aceleraciones**, **velocidades** y **posiciones**, que se representan como: $a_1, v_1, p_1, \dots, a_N, v_N, p_N$, y sus clases a las que pertenecen: **Con daño** y **Sin daño**, como C_i , donde: i representa *Con daño* o *Sin daño*. Dentro de la implementación de las Máquinas de Soporte Vectorial, se pueden aplicar distintos *kernels*, v.g. el de base radial, el lineal y el polinomial, con el objetivo de elegir el *kernel* que mapee mejor a los datos para su clasificación. Una vez aplicado una Máquina de Soporte, se espera obtener un clasificador, capaz de distinguir si existe la presencia o no de daño en el edificio. En la Figura 4.2, se muestra el diagrama del proceso seguido para la construcción del modelo de clasificación.

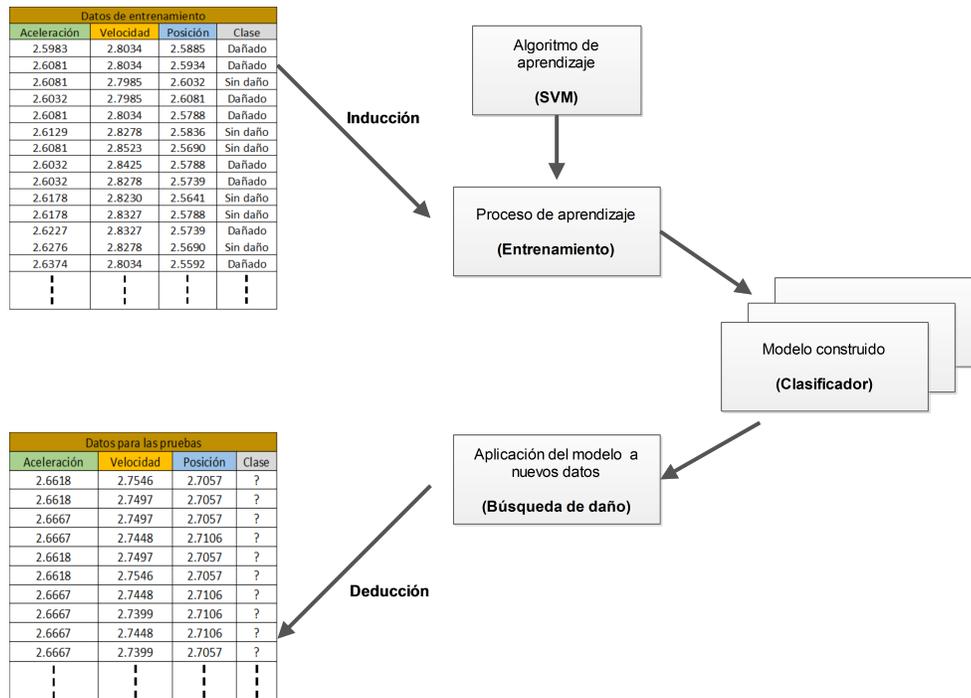


Figura 4.2: Diagrama de la construcción del modelo de clasificación para la detección de daños

4.2. Máquinas de Soporte Vectorial

Las Máquinas de Soporte Vectorial, Máquinas de Vectores de Soporte o simplemente *SVM*¹ por sus siglas en inglés; son un conjunto de algoritmos de aprendizaje supervisado, desarrollados por Vladimir N. Vapnik et al., (1993-1995) [17] [27] [30]. Las SVM's residen dentro de la inteligencia artificial (*IA*), para el reconocimiento de patrones. Estos métodos están propiamente relacionados con problemas de clasificación y regresión, cuyo objetivo es asignar a cada elemento de un conjunto de datos una de las clases existentes.

A últimas fechas, las SVM's, han demostrado ser un método muy bueno en distintas áreas como la medicina, los negocios, la investigación, etc., debido a las sólidas bases matemáticas que hay detrás y los buenos resultados obtenidos [1] [27]. Las SVM's incluso han llegado a desplazar a otros métodos tradicionales como el teorema de *Bayes* o el Análisis Lineal Discriminante (*LDA* por sus siglas en inglés). Dicho lo anterior, se ha decidido utilizar este método, con el objetivo de clasificar la existencia de daño en nuestro edificio que ha sido sometido a vibraciones. A continuación, en las siguientes sub-secciones se describe la teoría que existe detrás de las SVM's, adecuada a nuestro problema, así como los resultados que se han obtenido tras aplicar esta técnica.

¹A partir de ahora se podrá utilizar el término SVM para definir a las Máquinas de Soporte Vectorial.

4.2.1. Problema de clasificación entre dos clases (Con daño y Sin daño)

El objetivo del problema de clasificación para las SVM's, es encontrar un clasificador que logre separar dos o más clases de un conjunto de datos, mapeando los vectores de entrada en una dimensión mayor (en nuestro caso los valores a_i, v_i, p_i), así como encontrar un hiperplano óptimo de separación. Así por ejemplo, dado nuestro conjunto de datos de entrenamiento, podemos etiquetar las clases existentes C_i , de tipo binario como: $+1$ y -1 , con el objetivo de entrenar una SVM, la cual pueda construir un modelo que prediga la clase a la que pertenece un nuevo objeto. Ahora consideremos la Figura 4.3, la cual hace referencia a dos de los atributos de nuestro conjunto de datos, hipotéticamente divididos en dos clases por algunos de los posibles hiperplanos capaces de separar a dichas clases.

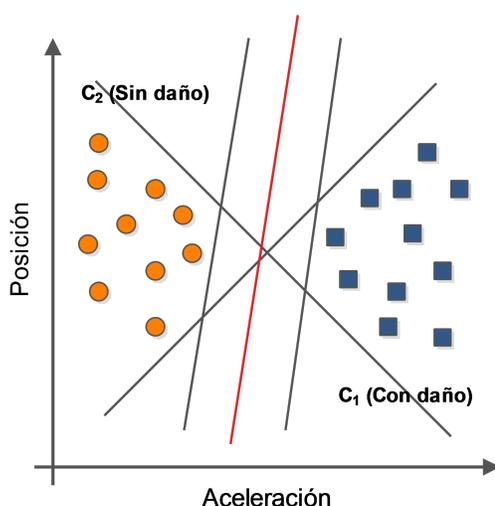


Figura 4.3: Clasificación del daño y sus posibles hiperplanos

De esta división, sólo existe un hiperplano que maximiza el margen entre una clase y otra, i.e. que maximiza la distancia entre el margen y los puntos de datos más cercanos de cada clase (también llamados vectores soporte), denominado hiperplano óptimo de separación (en la Figura 4.3, siendo el hiperplano de color rojo el óptimo).

4.2.2. Hiperplano óptimo de separación entre clases C_1 y C_2

Como vimos en la Figura 4.3, existe un sin fin de hiperplanos que dividen a las clases del conjunto de datos; y encontrar el hiperplano óptimo que maximice la distancia entre las clases, es uno de los objetivos principales dentro del proceso de clasificación. Por ejemplo;

considerando nuestros datos como sigue:

$$\begin{aligned} \text{aceleración} &= a_i, \\ \text{velocidad} &= v_i, \\ \text{posición} &= p_i. \end{aligned} \quad (4.1)$$

El problema de separar el conjunto de vectores de entrenamiento que pertenecen a las dos clases separadas C_i , con:

$$\begin{aligned} C_1 &= \text{Con daño}, \\ C_2 &= \text{Sin daño}, \end{aligned} \quad (4.2)$$

donde las clases C_i , también pueden ser etiquetadas como $Y_i = \pm 1$, está dado por:

$$X = \{((a_1, v_1, p_1), Y_1), \dots, ((a_N, v_N, p_N), Y_N)\}, \text{ donde } : a, v, p \in \mathfrak{R}^N, Y_i \in \{+1, -1\}, \quad (4.3)$$

donde X representa a nuestro conjunto de datos que existe en un espacio N -dimensional, con una función lineal discriminante (hiperplano) $g(x)$, tal que:

$$g(x) = \langle w, x \rangle + b = 0, \quad (4.4)$$

donde x puede representar a cada uno de nuestros vectores a, v y p ; que puede pertenecer a una clase o a otra, dada la regla de decisión:

$$\langle w, x \rangle + b = 0 \begin{cases} \Rightarrow x \in C_1 \text{ para } Y_i = +1, & \text{si } > 0 \\ \Rightarrow x \in C_2 \text{ para } Y_i = -1, & \text{si } < 0. \end{cases} \quad (4.5)$$

Podemos decir que el conjunto de vectores es separado de manera óptima por el hiperplano si se separa sin error, y la distancia entre el vector más cercano al hiperplano es máxima. En la Ecuación 4.4 existe una cierta redundancia, donde es apropiado considerar un hiperplano canónico (Vapnik, 1995), en donde los parámetros w, b están restringidos por:

$$\min_i |\langle w, (a_i, v_i, p_i) \rangle + b| = 1. \quad (4.6)$$

Esta restricción de parametrización es preferible para simplificar la formulación del problema. En pocas palabras: la distancia entre cada uno de los dos hiperplanos adyacentes del hiperplano $g(x) = 0$, es $\frac{1}{\|w\|}$, el cual la norma del vector de soporte, debe ser igual a la inversa de la distancia del punto más cercano en el conjunto de datos hacia el hiperplano $g(x) = 0$. La idea se ilustra en la Figura 4.4, donde se muestra la distancia desde el punto más cercano a cada hiperplano.

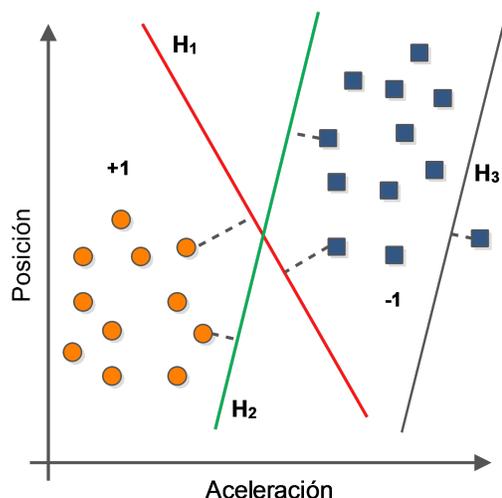


Figura 4.4: Posibles hiperplanos canónicos

Un hiperplano en su forma canónica debe satisfacer las siguientes restricciones:

$$Y_i[\langle w, (a_i, v_i, p_i) \rangle + b] \geq 1, \quad i = 1, 2, \dots, N. \quad (4.7)$$

Entonces, la distancia $d(w, b; (a_i, v_i, p_i))$ de un punto x hacia el hiperplano w, b es equivalente a:

$$d(w, b; x) = \frac{|\langle w, (a_i, v_i, p_i) \rangle + b|}{\|w\|} \equiv \frac{1}{\|w\|}. \quad (4.8)$$

Por tanto, maximizar el margen m para encontrar el hiperplano óptimo, significa encontrar una solución que minimice $\|w\|$, sujeto a las restricciones de la Ecuación 4.7 (ver Figura 4.5). Dado un margen definido por:

$$\begin{aligned} m(w, b) &= \min_{(a_i, v_i, p_i): Y_i = -1} d(w, b; (a_i, v_i, p_i)) + \min_{(a_i, v_i, p_i): Y_i = +1} d(w, b; (a_i, v_i, p_i)) \\ &= \min_{(a_i, v_i, p_i): Y_i = -1} \frac{|\langle w, (a_i, v_i, p_i) \rangle + b|}{\|w\|} + \min_{(a_i, v_i, p_i): Y_i = +1} \frac{|\langle w, (a_i, v_i, p_i) \rangle + b|}{\|w\|} \\ &= \frac{1}{\|w\|} (\min_{(a_i, v_i, p_i): Y_i = -1} |\langle w, (a_i, v_i, p_i) \rangle + b| + \min_{(a_i, v_i, p_i): Y_i = +1} |\langle w, (a_i, v_i, p_i) \rangle + b|) \\ &= \frac{2}{\|w\|}. \end{aligned} \quad (4.9)$$

La Ecuación 4.9, se reduce a minimizar, ya que es más fácil encontrar soluciones en un espacio de búsqueda de orden cuadrado (i.e. debido a que para una ecuación $f(x)$ de orden

cuadrático existe un solo mínimo global), por lo tanto, el hiperplano que separa de manera óptima los datos es el que minimiza:

$$\Phi(w) = \frac{1}{2} \|w\|^2. \quad (4.10)$$

La Ecuación 4.10, es independiente de b de la Ecuación 4.7, ya que se separa el hiperplano. En consecuencia, el margen se mantiene sin cambios, empero el hiperplano ya no es óptimo, debido a que estará más cerca de una clase con respecto de otra. Para considerar cómo minimizar la Ecuación 4.10, es equivalente a implementar el principio SRM (*Structural Risk Minimization*) y la utilización de un problema de optimización, sujeto a las restricciones de la Ecuación 4.7, que se resuelven mediante la utilización de los *multiplicadores de Lagrange* (Langrangianos), tal que:

$$\Phi(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i^n \alpha_i (Y_i [\langle w, (a_i, v_i, p_i) \rangle + b] - 1), \quad (4.11)$$

donde α son los *multiplicadores de Lagrange*. Por tanto, el Lagrangiano tiene que ser minimizado con respecto a: w, b y maximizado con respecto a: $\alpha \geq 0$. El problema de *Dualidad de Lagrangiano* permite transformar la Ecuación 4.11 primal a su dual, el cual es más fácil de resolver, como:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} (\min_{w, b} \Phi(w, b, \alpha)). \quad (4.12)$$

El mínimo con respecto a w y b del Lagrangiano Φ , es dado por:

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 &\Rightarrow \sum_{i=1}^n \alpha_i Y_i = 0 \\ \frac{\partial \Phi}{\partial w} = 0 &\Rightarrow w = \sum_{i=1}^n \alpha_i Y_i (a_i, v_i, p_i). \end{aligned} \quad (4.13)$$

Por lo tanto, a partir de las Ecuaciones 4.11, 4.12 y 4.13 el problema dual es:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Y_i Y_j \langle (a_i, v_i, p_i), (a_j, v_j, p_j) \rangle + \sum_{k=1}^n \alpha_k, \quad (4.14)$$

y la solución para el problema está dada por:

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Y_i Y_j \langle (a_i, v_i, p_i), (a_j, v_j, p_j) \rangle - \sum_{k=1}^n \alpha_k, \quad (4.15)$$

sujeto a restricciones tales:

$$\begin{aligned} \alpha_i &\geq 0 \quad i = 1, 2, \dots, N \\ \sum_j^n \alpha_j Y_j &= 0. \end{aligned} \quad (4.16)$$

Donde resolviendo la Ecuación 4.15 sujeta a restricciones, la Ecuación 4.16 determina los multiplicadores de Lagrange. Así mismo, el hiperplano óptimo de separación está dado por:

$$\begin{aligned} w^* &= \sum_{i=1}^n \alpha_i Y_i(a_i, v_i, p_i) \\ b^* &= -\frac{1}{2} \langle w^*, x_r + x_s \rangle, \end{aligned} \quad (4.17)$$

donde x_r y x_s son los vectores soporte de cada clase.

$$\alpha_r, \alpha_s > 0, \quad Y_r = -1, \quad Y_s = +1. \quad (4.18)$$

Para el caso de un clasificador de márgenes duros se define como:

$$f(x) = \text{sgn}(\langle w^*, (a_i, v_i, p_i) \rangle + b). \quad (4.19)$$

Alternativamente, un clasificador de márgenes blandos puede ser utilizado interpolando linealmente el margen:

$$\begin{aligned} f(x) = h(\langle w^*, (a_i, v_i, p_i) \rangle + b) \quad \text{donde} \quad h(g) = & \begin{aligned} & -1 : g < -1 \\ & g : -1 \leq g \leq 1 \\ & +1 : g > 1 \end{aligned} \\ \text{donde} \quad g = \langle w^*, (a_i, v_i, p_i) \rangle + b = 0 & \end{aligned} \quad (4.20)$$

Esto puede ser más apropiado que el clasificador de margen duro de la Ecuación 4.19, ya que produce una salida de valor entero entre -1 y $+1$, cuando el clasificador se mantiene al margen donde no hay datos de entrenamiento. Los conceptos expuestos anteriormente, pueden ser resumidos gráficamente en la Figura 4.5, en donde se muestran las ecuaciones para el hiperplano que divide a las clases en grupos, C_1 y C_2 . Así mismo, el margen (m) es la distancia entre el hiperplano ($g(x) = 0$) y los hiperplanos delimitados por los vectores soporte (x^+ , x^-).

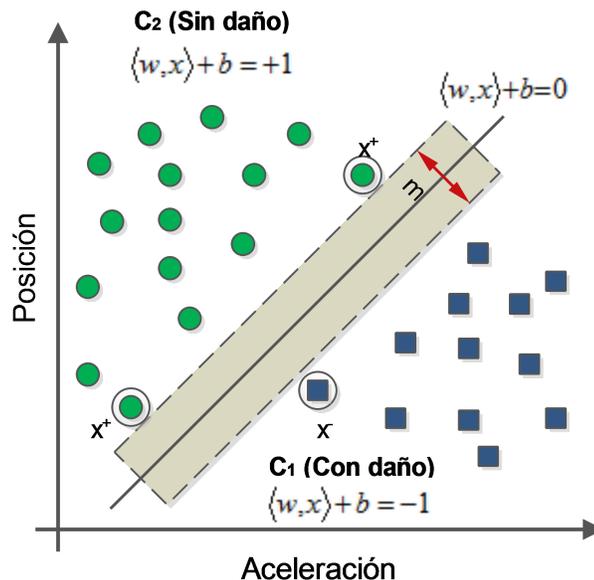


Figura 4.5: SVM con el hiperplano óptimo para la separación de las clases C_1 y C_2

4.2.3. Condiciones Karush-Kuhn-Tucker

Una vez obtenidos los multiplicadores de Lagrange α_i , el valor de w_0 puede ser encontrado a través de las condiciones de *Karush-Kuhn-Tucker* (también conocidas como condiciones KKT), las cuales son utilizadas para resolver el problema de optimización general para encontrar el hiperplano de separación óptimo, siendo una generalización de los multiplicadores de Lagrange.

De las condiciones KKT se tiene:

$$\alpha_i(Y_i[\{w, (a_i, v_i, p_i)\} + b] - 1) = 0, \quad i = 1, 2, \dots, N, \quad (4.21)$$

y por tanto, solo los puntos a_i, v_i y p_i satisfacen a la ecuación siguiente:

$$Y_i[\{w, (a_i, v_i, p_i)\} + b] = +1, \quad (4.22)$$

en donde los multiplicadores de Lagrange son diferentes de 0, llamándose así mismo Vectores de Soporte (VS). Si los datos son linealmente separables, todos los VS se encontrarán entre los límites del margen, y por lo tanto, el número de VS puede ser muy pequeño. Consecuentemente, el hiperplano será determinado por un pequeño subconjunto de puntos del conjunto de entrenamiento, dada la siguiente igualdad:

$$\|w\|^2 = \sum_{i=1}^n \alpha_i = \sum_{i \in VS} \alpha_i = \sum_{i \in VS} \sum_{j \in VS} \alpha_i \alpha_j Y_i Y_j \langle (a_i, v_i, p_i), (a_j, v_j, p_j) \rangle. \quad (4.23)$$

Por tanto, dada la dimensión *Vapnik-Chernovenkis* el clasificador está limitado por:

$$h \leq \min[R^2 \sum_{i \in VS}, n] + 1, \quad (4.24)$$

donde R es el radio de una hiper-esfera que encierra a todos los puntos del conjunto de datos. Así mismo, si los datos de entrenamiento, a , v y p son normalizados, entonces:

$$h \leq 1 + \min[\sum_{i \in VS}, n]. \quad (4.25)$$

4.2.4. Hiperplanos de margen blando para casos linealmente no separables de C_1 y C_2

Las condiciones vistas anteriormente, cumplen cuando los datos de entrenamiento son linealmente separables (ver Figura 4.5), sin embargo, en general esto no sucede en casos de aplicaciones reales. Consideremos por ejemplo, la situación en que los datos están mezclados y que la condición de la Ecuación 4.19 no se cumpla para $Y_i(\langle w, (a_i, v_i, p_i) \rangle + b) \geq 1, \forall i$ (ver Figura 4.6). Esto se debe a que los puntos de intersección no pueden ser correctamente clasificados y para cualquier otro objeto mal clasificado, el correspondiente α_i de los vectores de soporte tenderán al infinito.

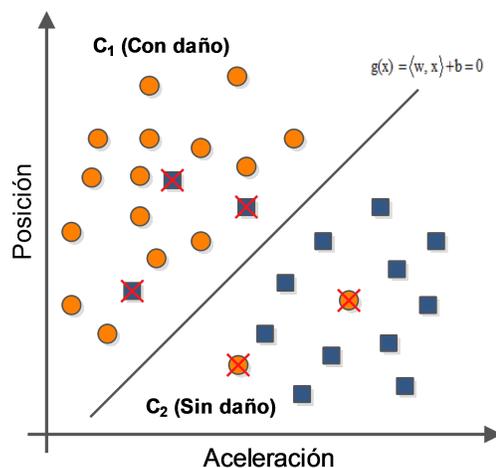


Figura 4.6: SVM con hiperplano de margen blando

En el caso en que se espera que un hiperplano pueda separar correctamente los datos cuando éstos están en intersección, una función adicional de costo debe ser implementada;

permitiendo al margen clasificar ambas clases. Por lo anterior, es necesario introducir variables no negativas, $\xi_i \geq 0$ y una función de penalidad (Cortés y Vapnik, 1995) [17], tal que:

$$F_\sigma(\xi) = \sum_i \xi_i^\sigma; \quad \sigma > 0, \quad (4.26)$$

donde ξ_i , es una medida de los errores de los no clasificados. Entonces; el problema de optimización se posa tanto en minimizar el error de clasificación, así como el minimizar el límite sobre la dimensión Vapnik-Chernovenkis del clasificador. Las restricciones de la Ecuación 4.26 son modificadas para el caso no separable, como sigue:

$$Y_i[\langle w, (a_i, v_i, p_i) \rangle + b] \geq 1 - \xi_i, \quad i = 1, \dots, N. \quad (4.27)$$

donde $\xi_i \geq 0$, y el nuevo hiperplano óptimo es determinado por el vector w , el cual minimiza la función a:

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i, \quad (4.28)$$

donde C es un valor dado, sujeto a las restricciones de la Ecuación 4.27. Con lo anterior, la solución al problema de optimización de la Ecuación 4.28, está dada por el punto de la función del Lagrangiano:

$$\Phi(w, b, \alpha, \xi, \beta) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_{i=1}^n \alpha_i (Y_i[w^T, (a_i, v_i, p_i) + b] - 1 + \xi_i) - \sum_{j=1}^n \beta_j \xi_j, \quad (4.29)$$

donde α y β son los multiplicadores de Lagrange. Tomando en cuenta que el Lagrangiano tiene que ser minimizado con respecto a: $w, b, (a_i, v_i, p_i)$ y maximizado con respecto a: α, β . Al igual que antes, la dualidad del Lagrangiano permite el problema primal. El problema dual viene dado por:

$$\max_\alpha W(\alpha, \beta) = \max_{\alpha, \beta} (\min_{w, b, \xi} \Phi(w, b, \alpha, \xi, \beta)), \quad (4.30)$$

donde el mínimo con respecto a w , b y ξ del Lagrangiano Φ , es dado por:

$$\begin{aligned}\frac{\partial \Phi}{\partial b} &= 0 \Rightarrow \sum_{i=1}^n \alpha_i Y_i = 0 \\ \frac{\partial \Phi}{\partial w} &= 0 \Rightarrow w \sum_{i=1}^n \alpha_i Y_i(a_i, v_i, p_i) \\ \frac{\partial \Phi}{\partial \xi} &= 0 \Rightarrow \alpha_i + \beta_i = C.\end{aligned}\tag{4.31}$$

Por consiguiente, de las Ecuaciones 4.29, 4.30 y 4.31, el problema del dual es:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Y_i Y_j \langle (a_i, v_i, p_i), (a_j, v_j, p_j) \rangle + \sum_{k=1}^n \alpha_k,\tag{4.32}$$

y puesto que la solución del problema es dada por:

$$\alpha^* = \arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j Y_i Y_j \langle (a_i, v_i, p_i), (a_i, v_i, p_i) \rangle - \sum_{k=1}^n \alpha_k,\tag{4.33}$$

con restricciones tales que:

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, N; \quad \sum_{j=1}^n \alpha_j Y_j = 0.\tag{4.34}$$

La condición de Kuhn-Tucker está dada por:

$$\alpha_i^* [Y_i(\langle w^*, (a_i, v_i, p_i) \rangle) + b^*] - 1 + \xi_i = 0,\tag{4.35}$$

en donde es la solución al problema de minimización que es idéntico para el caso separable, a excepción de una modificación de las restricciones de los multiplicadores de Lagrange.

4.2.5. Funciones Kernel para el problema de detección de daño

La idea de aplicar una función *kernel* al conjunto de datos, es permitir realizar operaciones en el espacio de entrada hacia un espacio de más alta dimensión. Por lo tanto, el producto punto no necesita ser evaluado en el espacio de características. Esto proporciona una forma de hacer frente a problemas en donde los datos no se separan linealmente. Sin embargo,

el cálculo es todavía dependiente del número de patrones de entrenamiento para proporcionar una buena clasificación de los datos. Para un problema de alta dimensión requerirá generalmente un conjunto de entrenamiento grande.

La idea básica en diseño de SVM's no lineales, es transformar los vectores de entrada, v.g. $a, v, p \in \mathfrak{R}^n$, dentro de vectores $\Phi(a, v, p)$ de un espacio de características altamente dimensional F (donde Φ representa el mapeo: $\mathfrak{R}^n \rightarrow \mathfrak{R}^f$), y resolver el problema de clasificación lineal en este espacio de características.

$$a, v, p \in \mathfrak{R}^n \rightarrow \Phi(a, v, p) = [\phi_1(a, v, p), \phi_2(a, v, p), \dots, \phi_n(a, v, p)]^T \in \mathfrak{R}^f. \quad (4.36)$$

El conjunto de hipótesis que consideraremos serán funciones de tipo:

$$f(x) = \sum_{i=1}^n w_i \phi_i(a_s, v_s, p_s) + b, \quad (4.37)$$

donde $\phi : X \rightarrow F$ es un mapeo no lineal desde un espacio de entrada a un espacio denominado característico (de mayor orden). Una propiedad de las máquinas de aprendizaje lineal es que éstas pueden ser expresadas en una representación dual, por lo tanto, la regla de decisión puede ser evaluada usando productos punto, tales que:

$$f(x) = \sum_{i=1}^n \alpha_i Y_i \langle \phi(a_i, v_i, p_i), \phi(a, v, p) \rangle + b. \quad (4.38)$$

A este método de cálculo directo se le llama función *kernel*.

Dentro de las distintas funciones de *kernel* existentes, el *kernel* lineal, las funciones polinomiales y la función Gaussiana (también conocida como: RBF *Radial Basis Function*) son las más utilizadas para el problema de la separación no lineal. A continuación, se definen algunos de los *kernels* existentes, los cuales serán utilizados para definir el que funcione mejor, para resolver nuestro problema de clasificación de daño.

- **Kernel lineal**

$$K((a_i, v_i, p_i), (a_s, v_s, p_s)) = (a_i, v_i, p_i)^T \cdot (a_s, v_s, p_s).$$

- **Kernel polinomial**

$$K((a_i, v_i, p_i), (a_s, v_s, p_s)) = ((a_i, v_i, p_i) \cdot (a_s, v_s, p_s) + 1)^d.$$

- **Kernel Gaussiano (RBF)**

$$K((a_i, v_i, p_i), (a_s, v_s, p_s)) = \exp\left(-\frac{(a_i, v_i, p_i) - (a_s, v_s, p_s)^2}{2\sigma^2}\right).$$

■ **Kernel perceptrón multi-capa**

$K((a_i, v_i, p_i), (a_s, v_s, p_s)) = \tanh(p \langle (a_i, v_i, p_i), (a_s, v_s, p_s) + \phi \rangle)$.
donde ϕ es el *offset*.

Máquinas de soporte no lineales

El principio de SVM no lineal consiste en mapear el espacio de entrada a un espacio de representación de dimensión alta a través de una función no lineal elegida a priori (véase 4.2.5). Este concepto puede ser visto en la Figura 4.7, donde se muestra una clasificación de dos clases del conjunto de datos (Figura 4.7a), mapeadas a una mayor dimensión (Figura 4.7b) en donde los objetos pueden ser divididos en sus correspondientes clases.

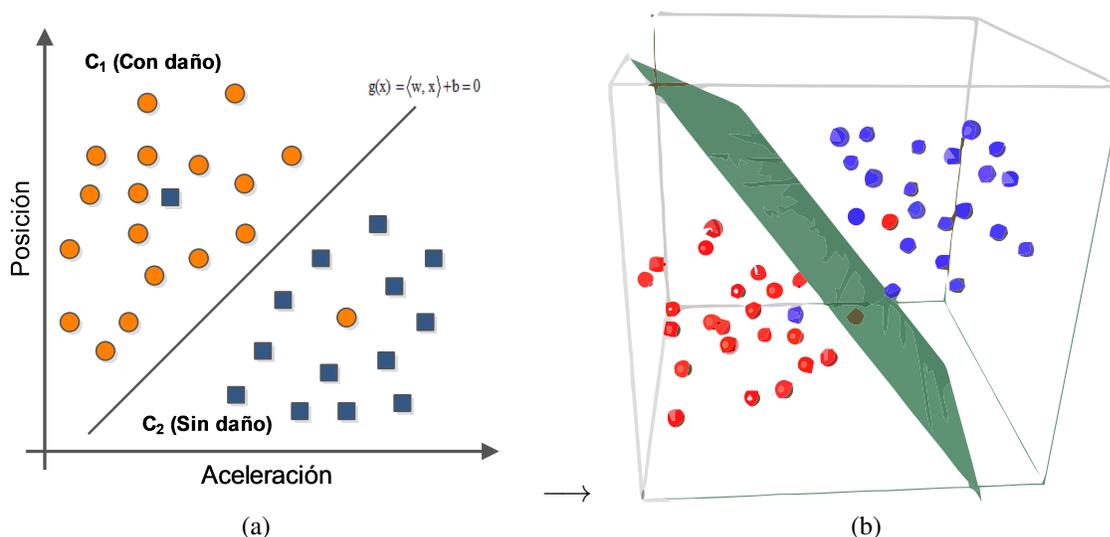


Figura 4.7: SVM mapeada en un plano dimensional mayor

Sin embargo, surge un problema computacional, la dimensión del espacio de representación puede ser muy alta y la dificultad radica en cómo construir un hiperplano de separación en este espacio. Como ya se vio en la sección anterior, la respuesta al problema parte de construir dicho hiperplano, el mapeo $z = \phi((a_i, v_i, p_i))$ no necesita ser explícito, de manera que reemplazando x por $\phi((a_i, v_i, p_i))$, se tiene:

$$\begin{aligned} \max \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j Y_i Y_j \phi(a_i, v_i, p_i)' \phi(a_j, v_j, p_j) \\ \text{suje}to \ a : \sum_i^n Y_i \alpha_i = 0, \ \alpha_i \geq 0 \ \forall i. \end{aligned} \tag{4.39}$$

De la Ecuación 4.39, el algoritmo de entrenamiento solo depende de los datos a través de los productos punto en el espacio de representación, de funciones de la forma $\phi(a_i, v_i, p_i) \phi(a_j, v_j, p_j)$. Sea dada una función *kernel* simétrica K tal que $K(x_i, x_j) = \phi(a_i, v_i, p_i) \phi(a_j, v_j, p_j)$, de modo que el algoritmo de entrenamiento dependa solo de K y el mapeo ϕ no sea usado explícitamente.

4.2.6. Condición de Mercer para los kernels

Existen diferentes tipos de *kernels* que pueden ser usados en una SVM, sin embargo, la condición de Mercer provee una caracterización a cerca de cuando una función es un *kernel* (Courant y Hilbert, 1959; Vapnik, 1998) [17]. Dado un *kernel* $K(x, y), x, y \in \mathbb{R}^n$, es un producto punto en algún espacio característico:

$$K(x, y) = \phi^T(x)\phi(y) \Leftrightarrow K(x, y) = K(y, x), \quad (4.40)$$

i.e. que pueda mapearse nuevamente a su función original. Y dada la integral:

$$\int K(x, z)f(x)f(z) dx dz \geq 0, \quad (4.41)$$

para todas las funciones f que satisfacen:

$$\int f(x)^2 dx < \infty. \quad (4.42)$$

Si K es una función simétrica definida positiva que satisface a 4.40, entonces, $K(x, y)$ puede ser expandida como:

$$K(x, y) = \sum_{i=1}^{\infty} \lambda_i \hat{\phi}_i(x) \hat{\phi}_i(y), \quad (4.43)$$

donde λ_i son los valores propios y $\phi_i(x)$, las funciones propias que satisfacen:

$$\int K(x, y)\phi_i(x)dx = \lambda_i\phi_i(x), \quad (4.44)$$

de tal modo que: $\hat{\phi}_i$ es normalizada, como:

$$\int \hat{\phi}_i^2(x)dx = 1. \quad (4.45)$$

4.3. Métricas de evaluación para la clasificación de las clases C_1 y C_2

La evaluación del desempeño de un modelo de clasificación, se basa en realizar un conteo en el número de aciertos y errores predichos por el modelo, o de cualquier otra técnica que permita saber la precisión del modelo construido.

Efectividad del modelo con la matriz de confusión

De los recuentos de aciertos o errores obtenidos del modelo construido; éstos se tabulan en una tabla conocida como *matriz de confusión*. Por ejemplo, consideremos nuestro conjunto de datos con a_i, v_i, p_i y nuestras clases C_1 y C_2 . Una vez construido el modelo con ayuda de una SVM y habiendo clasificado los nuevos objetos, ahora toca el turno de promediar el funcionamiento del clasificador, y con la matriz de confusión es posible. La Tabla 4.3 representa la matriz de confusión, donde cada entrada f_{ij} , indica el número de registros de la clase real C_i vs. la clase predicha (v.g. f_{01} es el número de registros de la clase 0 incorrectamente predicha como clase 1). Basado en las entradas de la matriz de confusión, el número total de predicciones correctas hechas por el modelo, es la suma de; ($f_{11} + f_{00}$). Así mismo, el número total de predicciones incorrectas es ($f_{10} + f_{01}$), tal como sigue:

		Clase predicha	
		Clase = 1	Clase = 2
Clase actual	Clase = 1	f_{11}	f_{10}
Clase actual	Clase = 2	f_{01}	f_{00}

Aunque una matriz de confusión proporciona la información necesaria para determinar la eficacia de un modelo de clasificación. Se necesita de un porcentaje de precisión y un porcentaje de la tasa de error, con el fin de evaluar si un modelo es lo suficientemente bueno como para ser utilizado en futuras clasificaciones. Las métricas para la obtener la precisión está dada por:

$$Precisión = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.46)$$

De manera equivalente, el rendimiento de un modelo se puede expresar en términos de su tasa de error, que está dada por la siguiente ecuación:

$$Tasa\ de\ error = \frac{\text{Número de predicciones erróneas}}{\text{Número total de predicciones}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}. \quad (4.47)$$

La mayoría de los algoritmos de clasificación buscan modelos que alcanzan la precisión más alta, o de manera equivalente; la tasa de error más baja, cuando se aplica al conjunto de pruebas.

Comparación de modelos mediante el análisis de curvas ROC

Otro de los métodos utilizados para comparar la eficiencia de los modelos construidos, es el análisis de curvas *ROC* (*Receiver Operating Characteristic* por sus siglas en inglés). El cual es una representación gráfica que ilustra el rendimiento de un sistema clasificador binario como su umbral de clasificación. El gráfico es creado mediante el trazado de la fracción de verdaderos positivos f_{11} del total de positivos reales ($TVP = Tasa\ de\ Verdaderos\ Positivos$) vs. la fracción de falsos positivos f_{01} , de los negativos totales reales ($TFP = Tasa\ de\ Falsos\ Positivos$). EL TVP es también conocido como sensibilidad o *recall* en el aprendizaje automático. El TFP se conoce también como la caída de salida. La curva ROC es entonces la sensibilidad como una función de la caída de salida. En general, si tanto la probabilidad de las distribuciones para de detección y los falsos positivos son conocidos, la curva ROC se puede generar mediante el trazado de la función de distribución acumulativa (área bajo la distribución de probabilidad de $-\infty$ a $+\infty$) de la probabilidad de detección en el eje Y , frente a la función de distribución acumulada de la probabilidad de los falsos positivos el eje X .

4.4. Resultados de los experimentos

Una vez vista la teoría que existe detrás de las SVM's, que van desde la formulación matemática hasta la evaluación de los modelos generados. Ahora toca el turno de mostrar los resultados obtenidos de construir y probar modelos para tratar de predecir el comportamiento de nuevos conjuntos de datos.

Conjunto de datos utilizados para el entrenamiento

En la Figura 4.8 se muestra un ejemplo de la agrupación de los datos en las tablas que se utilizaron para el entrenamiento de los modelos, como se ha venido insistiendo en secciones anteriores, se pueden observar los atributos de la aceleración, velocidad y posición, así como la clase en donde se tiene etiquetados los datos con daño y sin daño. El número total de datos por estado es de 60,000, lo que significa que en cada nivel se tomaron 20,000 para P1, 20,000 para P2, y 20,000 para P3, en donde cada dato tiene su etiqueta de clasificación según corresponde, si presenta daño o no.

Los datos mostrados en la Figura 4.8, provienen del escenario 1 del estado 1, donde se presenta el daño en P2 y de donde se han obtenido los valores del acelerómetro A2.

Datos de entrenamiento			
Aceleración	Velocidad	Posición	Clase
2.5983	2.8034	2.5885	Dañado
2.6081	2.8034	2.5934	Dañado
2.6081	2.7985	2.6032	Sin daño
2.6032	2.7985	2.6081	Dañado
2.6081	2.8034	2.5788	Dañado
2.6129	2.8278	2.5836	Sin daño
2.6081	2.8523	2.5690	Sin daño
2.6032	2.8425	2.5788	Dañado
2.6032	2.8278	2.5739	Dañado
2.6178	2.8230	2.5641	Sin daño
2.6178	2.8327	2.5788	Sin daño
2.6227	2.8327	2.5739	Dañado
2.6276	2.8278	2.5690	Sin daño
2.6374	2.8034	2.5592	Dañado
⋮	⋮	⋮	⋮

(a)

Datos para las pruebas			
Aceleración	Velocidad	Posición	Clase
2.6618	2.7546	2.7057	?
2.6618	2.7497	2.7057	?
2.6667	2.7497	2.7057	?
2.6667	2.7448	2.7106	?
2.6618	2.7497	2.7057	?
2.6618	2.7546	2.7057	?
2.6667	2.7448	2.7106	?
2.6667	2.7399	2.7106	?
2.6667	2.7448	2.7106	?
2.6667	2.7399	2.7057	?
⋮	⋮	⋮	⋮

(b)

Figura 4.8: Representación de los datos para entrenar y probar los modelos

Elección del *kernel*

Para la primera prueba de la elección del *kernel* se decidió particionar los datos en 50 % y 50 %, i.e. 50 % del escenario 1 del estado 1, para entrenar el modelo y el otro 50 % proveniente del escenario 2 del estado 1, para probar el modelo construido. La Figura 4.9, muestra cómo han sido divididos los datos de los experimentos, donde *SD* = Sin Daño y *CD* = Con Daño.

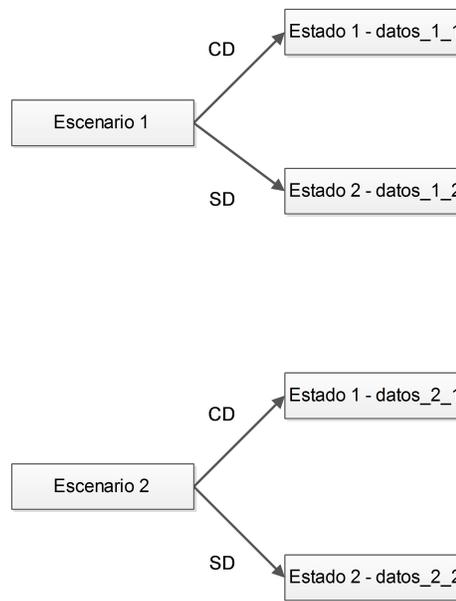


Figura 4.9: Agrupación de los datos por escenarios

Esta primera prueba se realizó con los atributos de la Aceleración y la posición (ya que fue la mejor combinación encontrada entre las tres variables, debido a que la velocidad está altamente cor-relacionada con la aceleración y la posición), con el fin de conocer cuál de los *kernels* resulta mejor para la predicción de modelos futuros y de esta manera utilizarlo como base para realizar las pruebas siguientes. Dicho lo anterior, se probaron varios *kernels* el primero de ellos fue el *kernel* lineal.

Kernel lineal

Los valores para este *kernel* se muestran en las tablas de la parte de abajo. La segunda tabla, muestra la matriz de confusión para este experimento, se observa que se obtiene un porcentaje de aciertos del 65.69 % y un porcentaje de errores del 34.30 %, sin embargo, es de notar que la diagonal principal es la de interés, ya que contiene los valores de acierto que ha logrado obtener el clasificador para clasificar el edificio como Con daño. Por otro lado, se observa que no predice muy bien cuando se presenta el daño, obteniendo valores muy bajos de aciertos con 16.46 %. Lo que indica que este *kernel* no es bueno para la clasificación del daño.

Valor	Aceleración	Posición
Min	0.0001	0.01913
1er Cu	1.4532	46.56836
Media	4.2991	51.93634
3er Cu	5.9951	51.93634
Max	93.7421	168.09554

Val. de aceleraciones y posiciones		
	Clase predicha	
Clase actual	Dañado	Sin daño
Dañado	3292	16708
Sin daño	3875	36125
Porcentaje de aciertos global: 65.69 %		
Porcentaje de error global: 34.30 %		
Aciertos Con daño: 16.46 %		
Aciertos sin daño: 90.3125 %		

Kernel Polinomial

Para este *kernel* se probaron distintos grados de polinomios. Uno de los resultados obtenidos se muestran en la tabla de la parte de abajo. N.b. que el *kernel*, no resultó para la construcción del modelo, puesto que clasificó a todos los objetos como Sin daño, con un porcentaje de aciertos global de 66.66 %, pero con un 0 % de aciertos para predecir el daño. Los mismos resultados se obtuvieron con otros *kernels* con diferente grado del polinomio.

Val. de aceleraciones y posiciones		
	Clase predicha	
Clase actual	Dañado	Sin daño
Dañado	0	20000
Sin daño	0	20000
Porcentaje de aciertos global: 66.66 %		
Porcentaje de error global: 33.33 %		
Aciertos de daño: 0 %		
Aciertos sin daño: 100 %		

Kernel Gaussiano Radial

En este *kernel* se obtuvieron los resultados que se muestran en la tabla de abajo. Este *kernel* resulta con mejores valores de predicción, tanto para predecir: Con daño o Sin daño, puesto que tiene un acierto global de 69.20 %, y un 66.47 % para los aciertos Con daño. Así mismo, se obtuvo un 70.57 % de aciertos sin daño. Lo cual es muy buen *kernel*, puesto que el problema de clasificación para este tipo de problemas es difícil.

Val. de aceleraciones y posiciones		
	Clase predicha	
Clase actual	Dañado	Sin daño
Dañado	13294	6706
Sin daño	11770	28230
Porcentaje de aciertos global: 69.20 %		
Porcentaje de error global: 30.79 %		
Aciertos de daño: 66.47 %		
Aciertos sin daño: 70.57 %		

Kernel Sigmoide

En este *kernel* se obtuvieron los resultados que se muestran a continuación en la tabla de abajo. Este *kernel* no tiene buenos valores de predicción para predecir Con daño, o para predecir Sin daño. Ya que solo se obtuvo un acierto global del 53.88 % y solo un 27.46 % de aciertos para clasificar el daño.

Val. de aceleraciones y posiciones		
	Clase predicha	
Clase actual	Dañado	Sin daño
Dañado	5492	14508
Sin daño	13163	26837
Porcentaje de aciertos global: 53.88 %		
Porcentaje de error global: 46.11 %		
Aciertos de daño: 27.46 %		
Aciertos sin daño: 67.09 %		

Área bajo la curva ROC

En la tabla de abajo se muestran los valores del área bajo la curva de los valores que se obtuvieron para cada uno de los *kernels* probados. Como vemos, el mejor *kernel* con los mejores valores de predicción, es el *kernel* Gaussiano radial. El área bajo la curva de un modelo ideal, sería del 100 %. Cabe destacar que se realizaron otras pruebas con las combinaciones de los datos pero todos mostraron mejor desempeño con un *kernel* Gaussiano Radial.

Kernel	Curva ROC
Lineal	66.01 %
Polinomial	53.41 %
Radial	67.01 %
Sigmoide	52.47 %

En la Figura 4.10, se muestra la gráfica de la curva ROC para los valores obtenidos de los diferentes *kernels*, la línea central indica que el área bajo la diagonal es del 50 %. Valores por encima de esta diagonal indican que el modelo comienza a ser buen clasificador (en la Figura 4.10 se grafican el TPR y el FPR, por sus siglas en inglés, n.b que la curva azul pertenece al *kernel* Gaussiano).

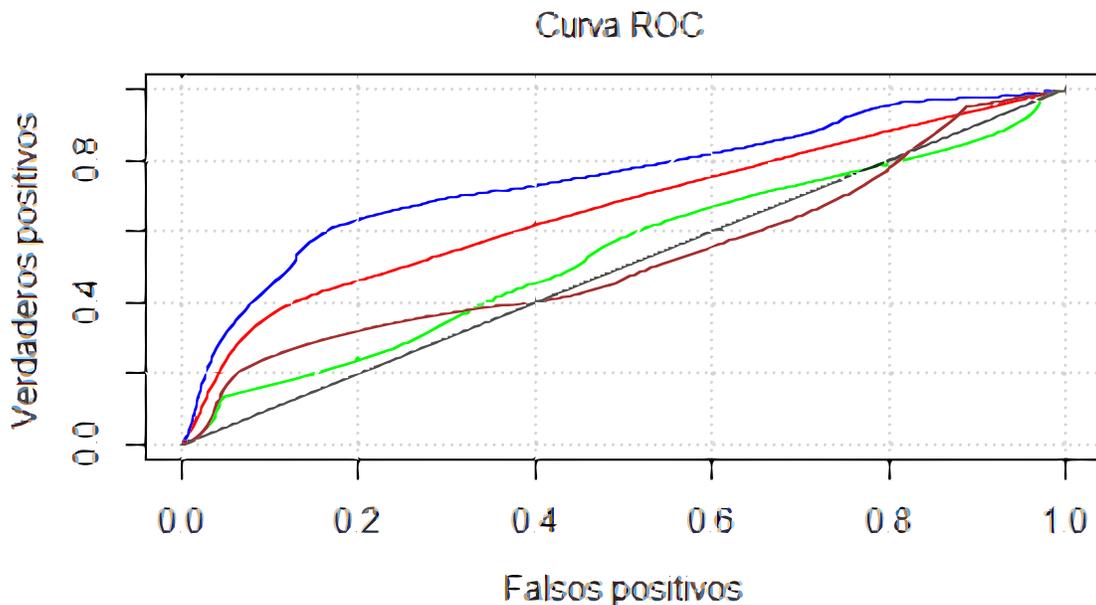


Figura 4.10: Valores de la curva ROC de los *kernels* utilizados

Una prueba más, consistió en tomar particiones de los datos con un 70 % y 30 % respectivamente, es decir 70 % de los datos del escenario 1 del estado 1 para entrenar el modelo

y el otro 30 % de los datos, para el escenario 2 del estado 1 para probar el modelo. Los resultados se muestran en la tabla de la parte de abajo, la cual el *kernel* Gaussiano Radial sigue teniendo buenos resultados con un 73.85 % por encima de los otros *kernels*. Ha de saberse que un área mayor al 50 % comienza a ser buen clasificador.

Kernel	Curva ROC
Lineal	70.61 %
Polinomial	50.95 %
Radial	73.85 %
Sigmoide	57.67 %

Otra prueba más, consistió en tomar un 100 % de los datos del escenario 1 y del escenario 2, para el estado 1, para entrenar el modelo. Por otro lado, se tomaron el 100 % de los datos del escenario 1 y 2, del estado 2, para probar el modelo. Visto de otra manera, el 50 % del estado 1 y el 50 % del estado 2. Los resultados se pueden observar en la tabla siguiente:

Kernel	Curva ROC
Lineal	59.22 %
Polinomial	55.77 %
Radial	67.22 %
Sigmoide	57.37 %

Dados los resultados de estos y otros experimentos, resulta ser que el *kernel* Gaussiano Radial, es un buen *kernel* para la construcción de modelos. Por tanto, fue elegido para realizar las pruebas siguientes.

Combinación de datos para la construcción de modelos

Una vez elegido el *kernel* adecuado para la construcción de modelos toca realizar las pruebas para las distintas combinaciones de los datos, i.e. hacer una combinación entre estados y escenarios con el fin de evitar el sobre-ajuste. El primer ejemplo ya se realizó en la sección anterior, en donde vemos que para el modelo construido con el *kernel* radial se obtuvo una precisión del 69.20 % de acierto global, con la separación de los datos 50 % y 50 %. Siguiendo el enfoque 50 % y 50 %, ahora se tomaron todos los datos del escenario 1 como entrenamiento y todos los datos del escenario 2 para prueba. De igual manera se utilizó un *kernel* Gaussiano Radial. Las Tablas 1 y 2 de la parte de abajo, muestran los valores obtenidos de ambos conjuntos (la media, los rangos inter-cuartílicos, los valores mínimos y máximos). La tercera tabla, muestra los resultados obtenidos para esta prueba. Como vemos, este modelo funciona bien para la clasificación de daño con un 65.50 % de aciertos global y para aciertos con daño 66.47 %.

Valor	Aceleración	Posición
Min	0.00005	0.00886
1er Cu	1.55614	44.31065
Media	4.2991	47.24596
3er Cu	51.94785	5.66094
Max	173.90858	93.74209

Valor	Aceleración	Posición
Min	0.000	0.00306
1er Cu	1.735	41.16471
Media	4.513	47.44488
3er Cu	6.241	55.40725
Max	91.940	168.76754

Val. de aceleraciones y posiciones		
Clase actual	Clase predicha	
	Dañado	Sin daño
Dañado	13294	6706
Sin daño	34686	65314
Porcentaje de aciertos global: 65.50 %		
Porcentaje de error global: 34.49 %		
Aciertos de daño: 66.47 %		
Aciertos sin daño: 65.31 %		

En la Figura 4.11 se muestran gráficamente los resultados obtenidos del clasificador. Se puede observar la construcción del hiperplano que delimita a las clases; así mismo, se observan los vectores de soporte (VS) encerrados en un círculo, también se observan los datos de entrenamiento y los datos que predijo el clasificador como: Con daño y Sin Daño.

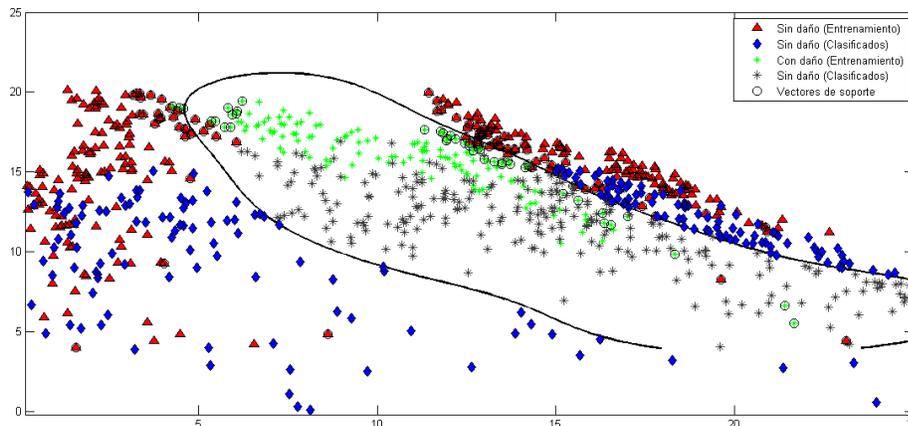


Figura 4.11: Clasificador construido para aceleración y posición

Para el siguiente experimento, se tomaron los datos del escenario 1 del estado 1 para el entrenamiento, así como los datos del escenario 2 del estado 1 para las pruebas, dividiendo los datos un 70 % para entrenamiento y 30 % para prueba (ver tabla de la parte de abajo). Sin embargo, a diferencia de las pruebas anteriores, los datos presentan solo los atributos de aceleración y velocidad.

Val. de aceleraciones y velocidades		
Clase actual	Clase predicha	
	Dañado	Sin daño
Dañado	1368	38632
Sin daño	678	79322
Porcentaje de aciertos global: 67.24 %		
Porcentaje de error global: 32.75 %		
Aciertos de daño: 3.42 %		
Aciertos sin daño: 99.15 %		

Como se ve en la tabla, este enfoque resultó no ser bueno para clasificar, mostrando un bajo desempeño. A continuación, se muestra la figura 4.12 del clasificador obtenido de las pruebas, en el cual no todos los datos fueron clasificados correctamente.

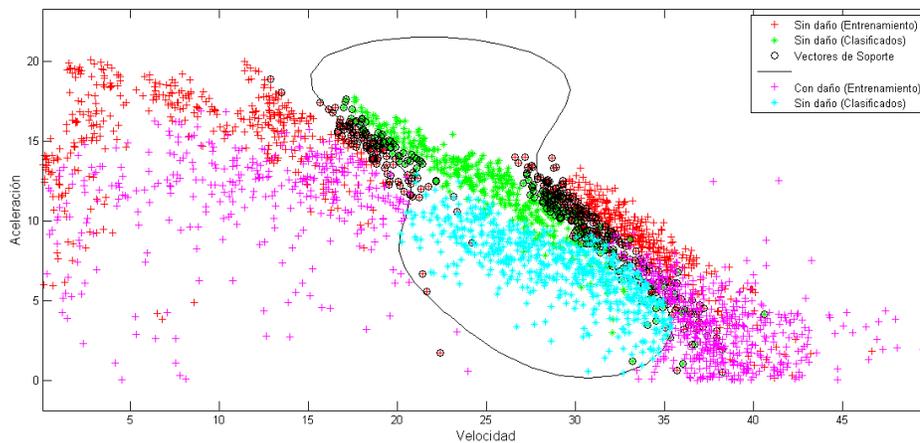


Figura 4.12: SVM construida para aceleración y posición

Otro experimento, consistió en combinar los tres valores, i.e. la aceleración, velocidad y posición para observar el comportamiento de la SVM a la hora de clasificar. Para este experimento se tomaron los valores del escenario 1 del estado 1 para entrenamiento y los valores del escenario 2 del primer y segundo estado para prueba. Para este modelo se obtuvieron los siguientes resultados como se ven en la tabla de la parte de abajo. Como se

observa, este modelo tiene un alto desempeño para la clasificación con un 75 % de acierto y solo un 25 % de error, siendo este el de mayor desempeño para los experimentos.

Val. de aceleraciones, velocidades y posiciones		
Clase actual	Clase predicha	
	Dañado	Sin daño
Dañado	15368	4632
Sin daño	12026	27974
Porcentaje de aciertos global: 75.23 %		
Porcentaje de error global: 25.76 %		
Aciertos de daño: 76.84 %		
Aciertos sin daño: 69.935 %		

En la Figura 4.13 se muestra el gráfico para este clasificador construido, en donde se muestra graficado solo los valores de la posición y la aceleración. Estos valores solo son una muestra de los valores reales, puesto que graficar todos los valores no serían vistos correctamente. En el gráfico se muestran los vectores soporte que fueron utilizados para construir el hiperplano que divide a las dos clases, también se muestran los puntos que se utilizaron para el entrenamiento y que fueron graficados, así como los nuevos datos sin etiqueta y se muestra cómo fueron clasificados en su clase a la que pertenecen.

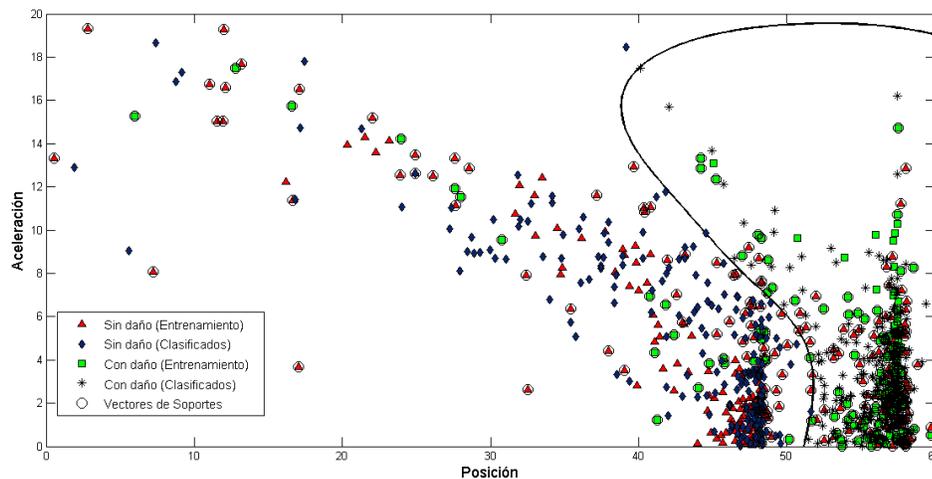


Figura 4.13: SVM para la clasificación del daño para aceleración y posición

Cabe destacar que siempre existen valores que no son clasificados correctamente y están traslapados en una clase a la que no corresponden, estos ejemplos se les denomina no clasificados correctamente. Aunque en la figura solo son pocos los puntos que están como no clasificados.

Finalmente en la Figura 4.14 se muestra el área bajo la curva que se obtuvo (porcentaje del área bajo la curva 74.05 %), en donde este modelo muestra buenos resultados frente a

otros modelos construidos.

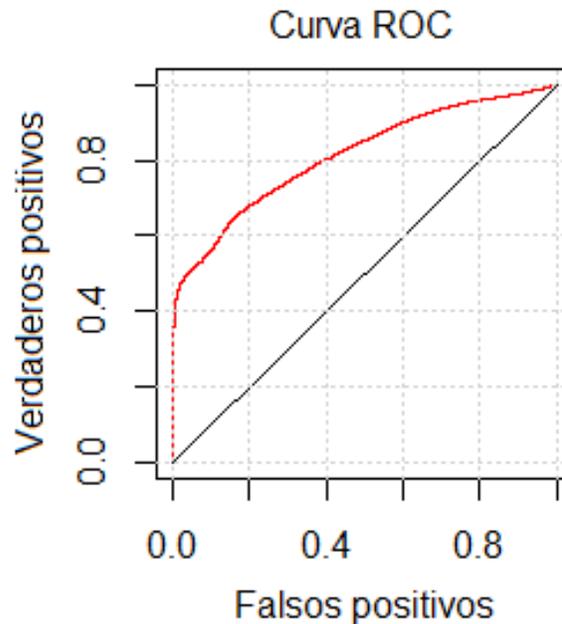


Figura 4.14: Área bajo la curva del modelo con aceleración, velocidad y posición

Comparación con otras técnicas

Una vez obtenidos los valores de los clasificadores y comprobado que las SVM's funcionan bien para clasificar entre la presencia de daño y cuando no existe el daño. Ahora toca comprar el desempeño de las SVM's, con las otras técnicas mencionadas en el Capítulo 2, K-NN y las Redes Neuronales.

Para el primer experimento se obtuvieron los resultados que se observan en la tabla de abajo. en el cual los datos fueron divididos 50% y 50%, i.e. 50% del escenario 1 del estado 1, para entrenar el modelo y el otro 50% proveniente del escenario 2 del estado 1, para probar el modelo construido.

	Precisión %	Error %	ROC
SVM	69.20	30.79	0.75
KNN	64.84	35.16	0.71
Redes Neuronales	64.81	35.19	0.67

De los resultados obtenidos, se observa que las SVM's resultan ser los modelos con mejor precisión a la hora de la clasificación, con un 69.20% de precisión y con un área bajo

la curva ROC, de 0.75 o el 75 %. Sin embargo, en este experimento las Redes Neuronales también resultaron ser buenos clasificadores con un 64.81 % de precisión.

Para la siguiente prueba, se tomaron el 100 % de los datos del escenario 1 del estado 1 y 2 para entrenar el modelo. Así mismo, se tomaron el otro 100 % de los datos del escenario 2 para probar el modelo. Los resultados obtenidos se observan la tabla siguiente:

	Precisión %	Error %	ROC
SVM	65.50	34.5	0.71
KNN	58.22	41.78	0.66
Redes Neuronales	62.12	37.88	0.70

En otra de las pruebas realizadas, se combinaron los tres atributos. Se tomaron los valores del escenario 1 del estado 1 para entrenamiento y los valores del escenario 2 del primer y segundo estado para prueba. Para este modelo se obtuvieron los siguientes resultados como se ven en la tabla de la parte de abajo.

	Precisión %	Error %	ROC
SVM	75.23	24.77	0.78
KNN	68.03	31.97	0.73
Redes Neuronales	71.83	28.17	0.77

4.5. Conclusión

El principal objetivo de las SVM's es producir un modelo que obtenga las etiquetas de la clase de un conjunto de datos de prueba, a partir de un conjunto de atributos. Sin embargo, el desempeño de una SVM depende de los parámetros utilizados sobre un conjunto de datos específico. Por tanto, es necesario construir y validar los modelos para escoger el que mejor se ajuste al problema que tenemos que resolver.

Con el objetivo de mostrar la validez, precisión y rapidez de clasificación, se construyeron diferentes clasificadores. En este capítulo se mostraron los resultados, obtenidos de los datos de los experimentos realizados, de cada uno de los escenarios. Algunos experimentos mostraron un buen desempeño al utilizar todas las variables predictivas, sin embargo, esto no quiere decir que entre mayor número de variables, mejor será el modelo; puesto que no todas las variables aportan el mismo peso para la clasificación. Hay que notar que el problema que se quiere resolver, es un problema difícil, ya que por la naturaleza de los datos (i.e que aparecen muy cercanos unos con otros), el problema se vuelve linealmente no separable. Que el problema se vuelva linealmente no separable, implica que se tengan que utilizar *kernels* que ayuden a mapear los datos a un nivel superior, en donde pueden ser separados de manera más fácil.

De los *kernels* utilizados, el RBF es uno de los *kernels* con mejores resultados para entrenar una SVM, mostrando mayor desempeño frente a otros *kernels* como: el lineal, sigmoide y polinomial. Sin embargo, la utilización de *kernels* que mapean a una dimensión

mayor, acarrea con sigo un problema de programación cuadrático, lo que implica que la máquina requiere de más iteraciones para alcanzar un nivel de convergencia óptimo.

Por otro lado, al igual que las SVM's, las Redes Neuronales y K-NN, resultan ser buenos a la hora de la clasificación, sin embargo, las SVM's llevan la ventaja sobre estas dos técnicas.

Para finalizar, n.b. que el enfoque utilizado para las SVM's, únicamente es capaz de clasificar dos clases Con Daño y Sin Daño. Sin embargo, de acuerdo con los niveles de clasificación de daño vistos en el Capítulo 2, dentro del paradigma del SHM; no solo basta con clasificar una estructura con daño o sin daño, sino que se requiere de mayor información para la toma de decisiones. En nuestro caso: la localización del daño por pisos, así como el nivel de daño que sufrió el edificio.

Capítulo 5

Árboles de Decisión para la detección de daños

En el Capítulo 4, se construyeron y evaluaron distintos modelos para la detección de daño en el edificio, con ayuda de las Máquinas de Soporte Vectorial. Así mismo, se comparó cada modelo entre sí, para determinar cuál es el óptimo para el problema de clasificación del daño (de acuerdo a las configuraciones de los datos). Sin embargo, a pesar de que es un método que ofrece buenos resultados, no es capaz de determinar la localización del daño, ya que solo clasifica objetos en dos entidades, i.e que no es multi-clase, y aunque existen algunas modificaciones que lo permiten no es viable, debido a que muchas veces el algoritmo no converge a la solución. Por tanto, con la intención de resolver este problema, en este capítulo se hace uso de otra de las técnicas de la Minería de Datos, denominada; *Árboles de Decisión*. Además, al final de este capítulo se hace la comparación entre los Árboles de Decisión, las Máquinas de Soporte Vectorial, con otras dos de las técnicas utilizadas, las Redes Neuronales y el K-NN.

5.1. Árboles de Decisión para la localización y severidad del daño

Una vez revisado el concepto de clasificación (ver Capítulo 4), podemos ahora definir a los Árboles de Decisión (*AD*¹). En Minería de datos, un AD es un modelo predictivo, el cual puede ser utilizado para representar tanto modelos de clasificadores como modelos de regresión, y cuya principal característica que los hace diferentes de otros enfoques de clasificación, es la representación de sus reglas para la toma de decisiones en forma de árbol.

¹A partir de ahora, los Árboles de Decisión podrán ser designados como AD.

5.1.1. Componentes del Árbol de Decisión

Un AD es una estructura muy sencilla, formada por nodos y ramas. A continuación, se describen cada una de sus partes de la que se compone.

- **Nodo raíz**

Un nodo raíz es el atributo principal a partir del cual se inicia el proceso de clasificación. Este nodo no tiene aristas (brazos) de entrada, sin embargo si tiene aristas de salida de donde se derivan más nodos.

- **Arcos o brazos**

Los arcos contienen valores posibles del nodo padre, los cuales sirven de conexión de unos nodos con otros.

- **Nodos internos**

Estos nodos se caracterizan por tener exactamente una arista entrante y dos o más aristas salientes.

- **Nodos Hoja**

Son unos nodos distintos a los nodos raíz e internos, los cuales son terminales del árbol, ya que al final clasifican el ejemplo como verdadero o falso. Estos nodos tienen exactamente una arista de entrada y no tienen aristas de salida.

En la Figura 5.1 se muestra la representación de los componentes de un AD con respecto a nuestro conjunto de datos, con los elementos antes citados.

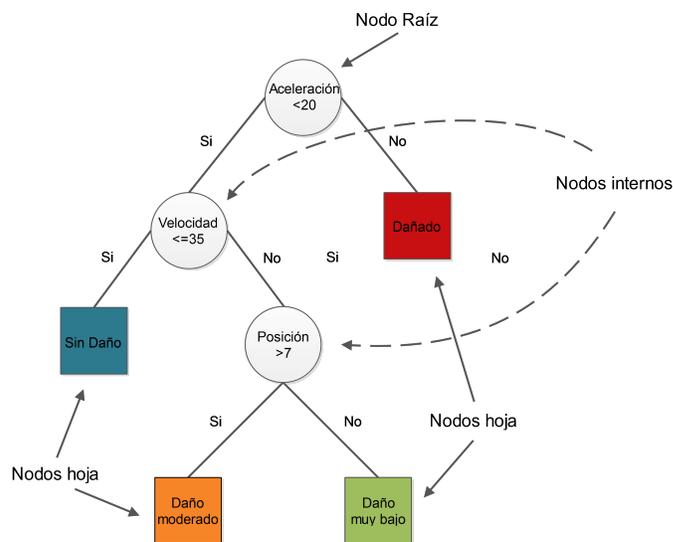


Figura 5.1: Componentes de un Árbol de Decisión

Como vemos, en un Árbol de Decisión en los nodos no terminales (los cuales incluyen el nodo raíz y el resto de los nodos internos), contienen las condiciones o preguntas que separan las decisiones hacia un extremo o el otro. También a cada nodo hoja se le es asignado una etiqueta de la clase C_i , que en nuestro caso le pertenece $C_1 = \text{Con Daño}$ y $C_2 = \text{Sin Daño}$. Otro ejemplo de un AD que se propone con respecto a nuestro problema, se muestra en la Figura 5.2, la cual representa la localización del daño por piso, i.e. definir en cuál(es) de los tres niveles del edificio se presenta el daño, dado nuestro conjunto de datos X .

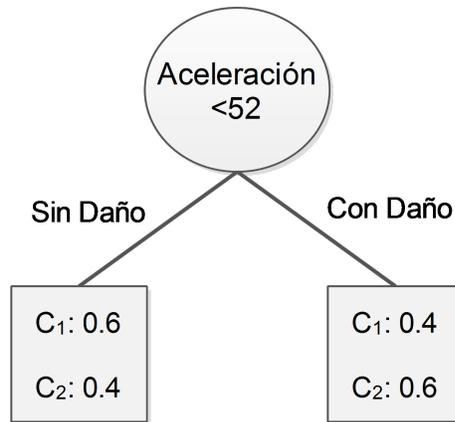


Figura 5.2: Árbol de Decisión para la detección de daño por niveles

Llegar a este nivel de localización, junto con la determinación de la severidad del daño, «según la escala de Rytter (ver Capítulo 2)» representan las condiciones deseadas dentro del SHM.

5.1.2. Construcción de Árboles de Decisión para el proceso de detección de daños

Los AD basan su funcionamiento en una serie de observaciones y construcciones lógicas que generan un conjunto de reglas, las cuales sirven para representar y categorizar una serie de condiciones que se presentan de manera secuencial. Los Árboles se crecen desde la raíz y se disgregan de manera sucesiva por cada uno de los nodos internos, hasta alcanzar un nivel de profundidad determinado. Por ejemplo; considerando nuevamente nuestro conjunto de datos, los cuales intervienen la **aceleración**, **velocidad** y **posición** como las variables predictivas, y las clases C_1 y C_2 , como los atributos objetivo. La Tabla 5.3, contiene las clases a las que pertenecen estos datos, contando con el hecho de saber si han sufrido daño. En este caso, el algoritmo debe de determinar un rango entre las tres variables de lo que podría considerarse “normal” y lo que no lo es, mediante una serie de preguntas hasta llegar a la toma de una decisión.

Datos de entrenamiento			
Aceleración	Velocidad	Posición	Clase
2.5983	2.8034	2.5885	Dañado
2.6081	2.8034	2.5934	Dañado
2.6081	2.7985	2.6032	Sin daño
2.6032	2.7985	2.6081	Dañado
2.6081	2.8034	2.5788	Dañado
2.6129	2.8278	2.5836	Sin daño
2.6081	2.8523	2.5690	Sin daño
2.6032	2.8425	2.5788	Dañado
2.6032	2.8278	2.5739	Dañado
2.6178	2.8230	2.5641	Sin daño
2.6178	2.8327	2.5788	Sin daño
2.6227	2.8327	2.5739	Dañado
2.6276	2.8278	2.5690	Sin daño
2.6374	2.8034	2.5592	Dañado
⋮	⋮	⋮	⋮

Figura 5.3: Representación del conjunto de datos para la clasificación del daño

De acuerdo a la tabla anterior, se podría formular una serie de preguntas sobre los atributos de cada registro, como por ejemplo; ¿es la aceleración ≥ 53 ?, si la respuesta es afirmativa podría seguirse preguntando; ¿la posición es > 33 ?, etc. Cada que se realiza una pregunta se recibiría una respuesta, y de acuerdo a la respuesta se puede seguir formulando más preguntas, hasta llegar a una conclusión sobre la etiqueta de clase a la que pertenecerá un nuevo objeto (ver Figura 5.4).

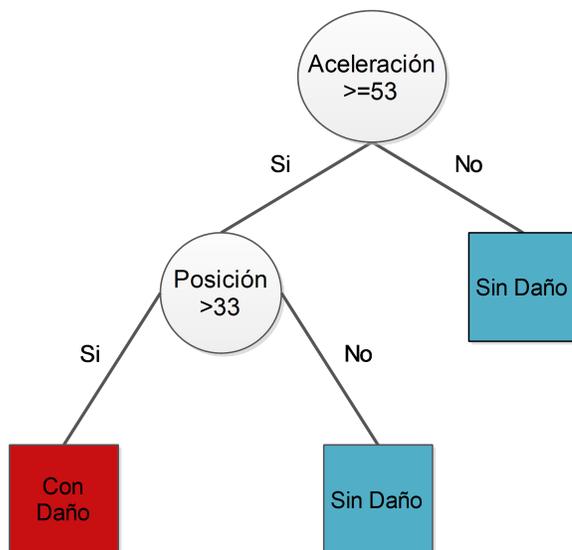


Figura 5.4: Construcción de un AD para la clasificación del daño

Algoritmos para la construcción de un Árbol de Decisión

En principio, existen un número exponencial de Árboles de Decisión que pueden ser construidos, dado un conjunto de atributos. En tanto que unos Árboles pueden ser más precisos que otros, encontrar el óptimo en términos computacionales es imposible, ya que el tamaño de búsqueda crece exponencialmente. Sin embargo, se han desarrollado algoritmos eficientes para tratar de mejorar este aspecto. Estos algoritmos usualmente emplean un grado de estrategia que hace que el algoritmo tenga convergencia a un óptimo local, con respecto de cuáles atributos utilizar para clasificar los datos. Uno de los principales algoritmos es el de *Hunt*, el cual es la base de otros algoritmos que son utilizados, en los que se incluyen; el *ID3*, *C4.5* y el *CART*.

Algoritmo de Hunt

En el algoritmo de Hunt, un AD crece de forma recursiva mediante la partición de los datos de entrenamiento en subconjuntos sucesivamente más puros. Por ejemplo; sea nuestro conjunto X el conjunto de datos de entrenamiento (ver Capítulo 4), que son los individuos; y dadas nuestras clases C_i , con $i = \{Con\ Daño, o\ bien, Sin\ Daño\}$, entonces el problema de la clasificación, es encontrar una función que construya el AD, tal que:

$$f : X \rightarrow C \mid \forall (a_i, v_i, p_i) \in C_i, \quad (5.1)$$

en donde la función, en este caso es un AD.

A continuación, se describe una definición recursiva del algoritmo de Hunt.

- **Paso 1:** Si todos los datos de X_i pertenecen a la misma clase C_i , entonces i es un nodo hoja etiquetado como C_i .
- **Paso 2:** Si X_i contiene datos que pertenecen a más de una clase, un atributo de condición es seleccionado para dividir los registros en sub-conjuntos más pequeños. Un nodo hijo se crea para cada resultado de la condición de prueba y los datos en X_i se distribuyen a los hijos basado en los resultados. El algoritmo es después aplicado de forma recursiva a cada nodo hijo.

En resumidas palabras: el algoritmo de Hunt funciona si para cada combinación de valores de atributos para los datos de entrenamiento se tiene una etiqueta de la clase única.

Durante la construcción del algoritmo se deben considerar algunos casos que pueden ocurrir, por ejemplo; que alguno de los nodos internos relacionados al Paso 2 del algoritmo de Hunt esté vacío. Esto puede pasar si ninguno de los datos de entrenamiento tiene los valores que se asocian con dicho nodo, en este caso el nodo sería un nodo terminal, asignándole la clase con mayor población asociado con su nodo padre. La otra cuestión surge cuando: todos los datos asociados con X_i tienen valores iguales (menos la clase), entonces

no sería posible dividir los registros y se asignarían a un nodo terminal, asociándolos con una clase de la que se tenga la mayoría de los registros que ya hayan sido asignados.

Cuestiones que surgen en el algoritmo de Hunt

Existen algunas preguntas que surgen a la hora de construir un AD y que deben ser resueltas o tratar de resolver lo mejor posible, ya que son de importancia para construir buenos modelos. A continuación, se describen cada una de éstas preguntas.

1. **¿Cómo se deben dividir las variables de entrenamiento?** Esta primera pregunta engloba cada paso recursivo del proceso para el crecimiento del Árbol en el que se debe seleccionar un atributo de la condición de la tabla de prueba para dividir los registros en subconjuntos más pequeños.
2. **¿Qué variables (atributos) utilizar y en qué orden?, ¿qué criterio utilizar para seleccionar la mejor división?** En respuesta a estas preguntas, se considera el error de clasificación, el índice de *Gini* y la *Entropía* (estas medidas son utilizadas para medir la pureza de un nodo).
3. **¿Cómo se debe detener el procedimiento de división? o cuándo termina el algoritmo?** No es fácil dar una respuesta concreta, ya que implica seleccionar un modelo. Se debe tomar en cuenta; qué tanto se quiere refinar las reglas generadas y eso es un proceso subjetivo. Una condición de parada es necesaria para terminar el proceso de crecimiento del Árbol. Lo ideal sería cuando todos los nodos sean puros o lo más puros posibles, sin embargo, esto ocasionaría reglas demasiado largas o construcciones muy lentas. Una idea sería controlar el error de clasificación, el índice de Gini o bien; la Entropía en el conjunto de datos de prueba, de manera que se detendrá cuando el índice sea lo mínimo posible. La poda (*pruning* en inglés) es la técnica más popular, propuesto por Breiman, et al. (1984) [26].

A continuación, se citan de manera general algunos otros algoritmos que son utilizados en la construcción de un AD.

- **ID3:** desarrollado por J. Ross Quinlan (1983), el cual es un algoritmo que aprende a partir de la diferencia que existe entre los datos para analizar, i.e. un procedimiento de divide y vencerás, que maximiza la información obtenida, la cual se utiliza como una métrica para seleccionar el mejor atributo que divida los datos en clases homogéneas.
- **J48:** este algoritmo se construye iterativamente al ir agregando nodos o ramas que minimizan la diferencia entre los datos. Es un descendiente del ID3 y se extiende en el sentido de su capacidad de utilizar atributos numéricos y vacíos para generar reglas del Árbol.
- **Naive Bayes:** es un algoritmo que genera un AD a partir del clasificador Bayesiano, que es el modelo más simple de clasificación, ya que asume independencia entre todos los atributos dada una clase. El fundamento principal de este clasificador, es

la suposición de que todos los atributos son independientes del valor de la variable clase.

5.1.3. Atributos para la construcción de Árboles de Decisión

En el contexto de la creación de un AD, se pueden utilizar distintos tipos de atributo para su construcción, y esto influye en cómo serán representados. En la Figura 5.5, se observa la organización de la manera en que se clasifican los tipos de variables, las cuales pueden ser utilizadas para la construcción de los AD's.

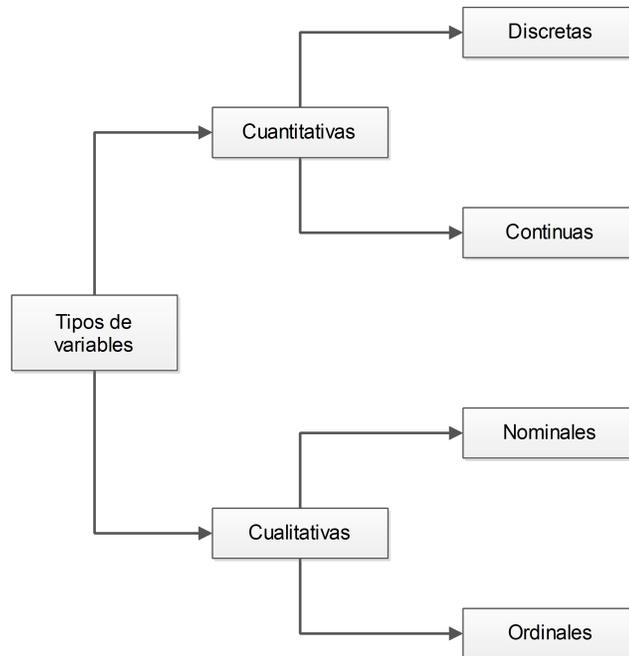


Figura 5.5: Tipos de variables para la construcción de Árboles de Decisión

Atributos binarios

Este tipo de atributos, como su nombre lo indica solo tiene dos tipos posibles de salidas como se muestra en la Figura 5.6.



Figura 5.6: Construcción de un Árbol con un atributo y dos salidas.

Atributos nominales

Una variable nominal es del tipo no numérico, y que además no está sometida a un criterio de orden. Y dado que un atributo nominal puede tener muchos valores, su condición de prueba se puede expresar en dos formas; tal como se muestra en la Figura 5.7 y Figura 5.8. En donde, considerando nuestro problema, se tiene un Árbol, en el cual se toma un atributo; *Piso en que se encuentra el daño*, en donde dadas ciertas condiciones se producirá una división de tres brazos. Por otro lado, algunos algoritmos para la construcción de AD's (v.g. CART), solo producen divisiones binarias del tipo:

$$\text{Número de particiones binarias} = 2^{k-1} - 1, \quad (5.2)$$

donde en la Ecuación 5.2, representa las maneras de crear una partición binaria de k atributos. En la Figura 5.8a, 5.8b y 5.8c se muestran las posibles formas dividir los Árboles en solo dos subconjuntos.

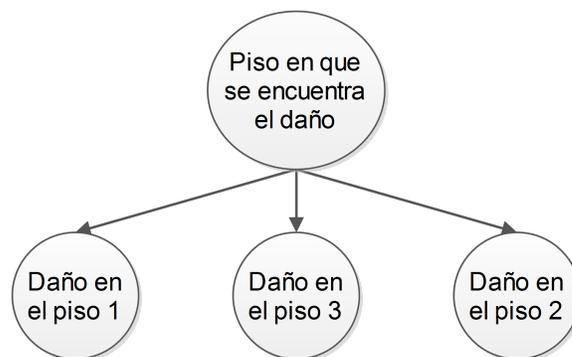


Figura 5.7: Árbol de múltiples salidas

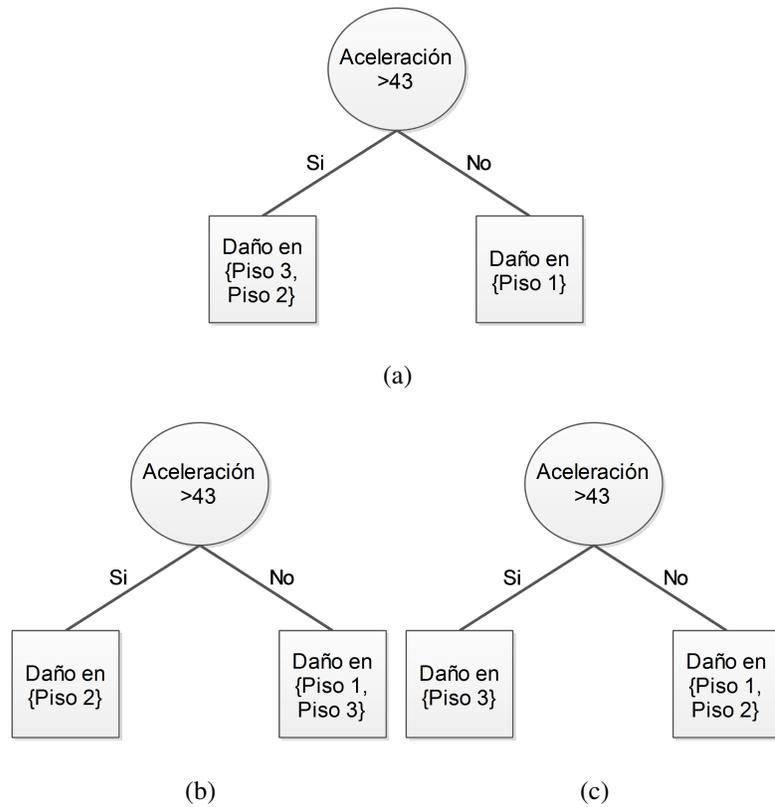


Figura 5.8: División binaria de un Árbol, agrupando los valores de los atributos

Atributos ordinales

Un atributo ordinal es similar a una variable nominal, la diferencia entre los atributos nominales, es que hay un orden claro de estos atributos. Un atributo ordinal también puede producir divisiones binarias o de más divisiones. La Figura 5.9 ilustra dos maneras de dividir los datos de entrenamiento. En las Figuras 5.9a y 5.9b mantienen un orden de acuerdo al tamaño, mientras que la Figura 5.9c no se cumple esta propiedad, ya que combina los valores de los atributos Moderado y Severo en el mismo conjunto.

Atributos continuos

Una variable continua puede tomar cualquier valor de los números reales, y siempre entre dos valores observables va a existir un tercer valor. El rango de los valores reales va desde $-\infty$ a $+\infty$. Para los atributos continuos; los datos de prueba pueden ser expresados como una prueba de comparación ($A < r$) o ($A \geq r$) con resultados binarios. O un rango con los resultados de la forma $r_i \leq A < r_i + 1$, para $i = 1, 2, \dots, N$. La diferencia entre estos enfoques se muestra en la Figura 5.10, para el caso binario, el algoritmo del Árbol debe tener en cuenta todas las posibles divisiones r , y seleccionar la que produce la mejor

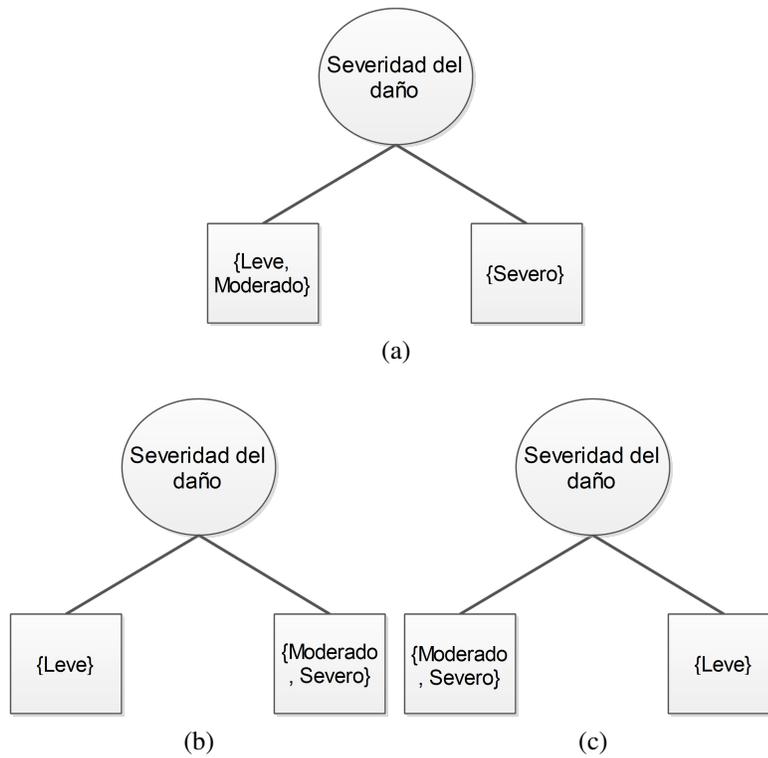


Figura 5.9: Diferentes maneras de agrupar valores de atributos ordinales

partición (ver Figura 5.10a). Para la división de múltiples brazos, el algoritmo debe tener en cuenta todos los rangos posibles de valores continuos. Una solución para esto es aplicar las estrategias de discretización. Después de la discretización, un nuevo valor ordinal será asignado a cada intervalo discretizado. Intervalos adyacentes también se pueden agregar en rangos más amplios, siempre y cuando la propiedad de orden se mantenga (ver Figura 5.10b).

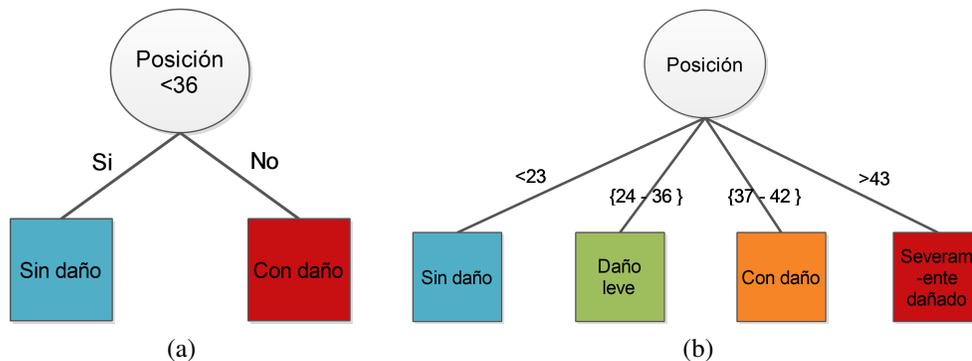


Figura 5.10: Construcción de Árboles con atributos continuos

5.1.4. Medidas para seleccionar la mejor división

Existen medidas que pueden ser utilizadas para determinar la mejor manera de dividir a los datos. Estas medidas se definen en términos de la distribución de clases de los datos antes y después de dividirse. Sea $p(C_i|t)$, donde se denota la porción de los datos pertenecientes a la clase C_i en un nodo dado t , o simplemente p_i . En un problema de dos clases, la distribución de probabilidad de que un nuevo objeto sea clasificado en cualquiera de los nodos, puede ser escrita como:

$$(p_0, p_1), \text{ donde } p_1 = 1 - p_0, \tag{5.3}$$

dado nuestro problema:

$$\begin{aligned} (p_a, p_v, p_p), \text{ donde } p_a &= 1 - p_v p_p \\ p_v &= 1 - p_a p_p \\ p_p &= 1 - p_a p_v, \end{aligned} \tag{5.4}$$

i.e. que la probabilidad caiga en cualquiera de los dos nodos, está dada por la diferencia de $1 - p(a)$. Podemos ver lo anterior en la Figura 5.11, en donde se muestran la distribución de la clase, antes de la división con peso (0.5, 0.5) debido a que hay un número igual de datos de cada clase. Si dividimos los datos mediante el atributo de aceleración, a continuación, las distribuciones de clase de los nodos secundarios podrían ser; (0.6, 0.4) y (0.4, 0.6), respectivamente.

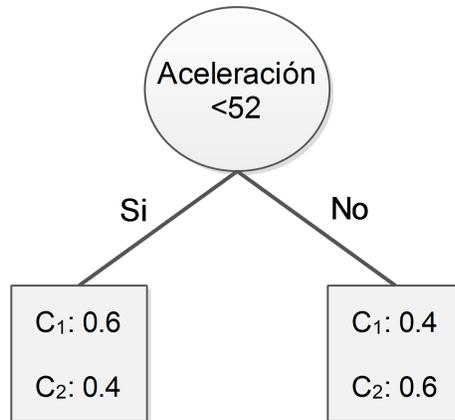


Figura 5.11: Probabilidades de clasificación para un nuevo objeto

Aunque las clases ya no se distribuyen de manera uniforme, los nodos secundarios aún contienen los datos de ambas clases. Las medidas desarrolladas para la selección de la mejor división, a menudo se basan en el grado de impureza de los nodos secundarios. Cuanto

menor sea el grado de impureza, los nodos serán más puros y por tanto; el algoritmo converge. Por ejemplo; un nodo con distribución de clase $(0, 1)$ tiene cero impureza, mientras que un nodo con una distribución uniforme de clase $(0.5, 0.5)$, tiene la mayor impureza posible. A continuación, se citan las medidas de impureza que son utilizadas para la poda del Árbol.

Índice de Gini

El coeficiente de Gini es una medida de la desigualdad. El cual, es un número entre 0 y 1, en donde 0 se corresponde con la perfecta igualdad y donde el valor 1, se corresponde con la perfecta desigualdad. El coeficiente de Gini se calcula como una proporción de las áreas en el diagrama de la curva de Lorenz. Si el área entre la línea de perfecta igualdad y la curva de Lorenz es a , y el área por debajo de la curva de Lorenz es b , entonces el coeficiente de Gini es: $\frac{a}{(a+b)}$ (ver Figura 5.12).

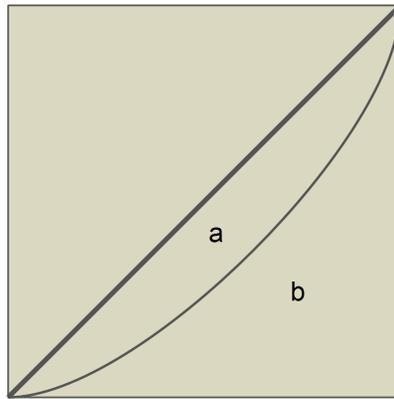


Figura 5.12: Coeficiente de Gini

En términos formales tenemos que:

$$G(t) = \left| 1 - \sum_{i=1}^{C-1} (X_{k+1} - X_k)(Y_{c+1} + Y_k) \right|. \quad (5.5)$$

Donde G , representa el coeficiente de Gini, X : es la proporción acumulada de la población, e Y es la proporción acumulada de los individuos por clase y C es el número de clases. *De manera resumida:* la Ecuación 5.5, puede escribirse como sigue:

$$G(t) = 1 - \sum_{i=1}^{c-1} [p(i|t)]^2. \quad (5.6)$$

Cabe destacar que una vez calculados los índices de cada nodo, el valor total del índice de Gini se calcula como el promedio ponderado del índice de Gini en cada nodo. La idea es que la suma ponderada de los nodos hijos sea menor que el índice de su padre, i.e. que los nodos se vuelvan más puros.

Entropía

En el ámbito de la teoría de la información, la entropía o *entropía de Shannon* (Claude E. Shannon) [27], mide la incertidumbre de una fuente de información. El concepto básico tiene mucho que ver con la incertidumbre que existe en cualquier experimento o señal aleatoria.

La entropía asociada a la variable aleatoria x , es un número que depende directamente de la distribución de probabilidad de la variable, e indica cómo será el resultado del proceso sujeto a incertidumbre. Desde un punto de vista matemático, cuanto más plana sea la distribución de probabilidad más difícil será acertar a cuál de las posibilidades pertenecerá cada instancia.

Tomando como base nuestros datos, la información que aporta un determinado valor, a_i, v_i, p_i , dadas las variables aleatorias a, v y p , la entropía se define como:

$$I(a_i, v_i, p_i) = \log_2 \frac{1}{p(a_i, v_i, p_i)} = -\log_2 p(a_i, v_i, p_i). \quad (5.7)$$

Por tanto, la entropía dadas nuestras variables a_i, v_i, p_i , dado $I(a_i, v_i, p_i)$, es la media ponderada de la cantidad de información de los distintos estados del evento, tal como sigue:

$$I(a_i, v_i, p_i) = - \sum_i^{c-1} p(a_i, v_i, p_i) \log_2 p(a_i, v_i, p_i), \quad (5.8)$$

dicho de otra manera:

$$I(a_i, v_i, p_i) = - \sum_i^{c-1} p(i|t) \log_2 p(i|t). \quad (5.9)$$

y dados los resultados, cuando la entropía es $= 0$ el nodo es puro, por otro lado cuando es 1 el nodo es impuro. Estos valores obtenidos son de ayuda para medir la pureza de los nodos en los que serán asignados los nuevos objetos.

Error de Clasificación

El error de clasificación al igual que el índice de Gini y la Entropía, es un indicador de qué tan puro es un nodo y está dado por:

$$Error(t) = 1 - \max_i[p(i|t)], \quad (5.10)$$

o también,

$$Error(t) = 1 - \max_j[p_x]. \quad (5.11)$$

En la Figura 5.13, se muestran las curvas medidas de las impurezas para los problemas de clasificación binaria para el **Índice de Gini, la Entropía y el Error de Clasificación**. Es necesario observar que todas las tres medidas alcanzan su valor máximo cuando la distribución de clase es uniforme (i.e. cuando $p = 0.5$). Los valores mínimos de las medidas se alcanza cuando todos los registros pertenecen a la misma clase (es decir, cuando p es igual a 0 o 1).

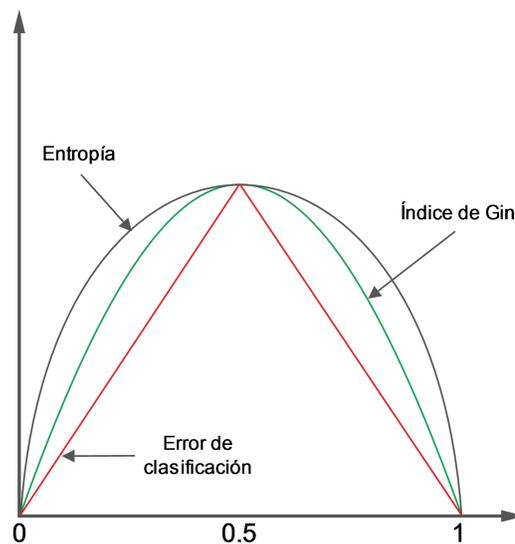


Figura 5.13: Gráficas para los distintos índices de impureza

5.1.5. Ganancia de información

Una vez calculados los valores de impureza, es necesario determinar cuanta ganancia se obtuvo al dividir nuevos nodos provenientes del nodo padre. Para determinar esta ganancia,

es necesario comparar el grado de impureza del nodo padre (antes de la división) con el grado de la impureza de los nodos secundarios (después de la división). La ganancia, Δ , es un criterio que se puede utilizar para determinar la eficacia de una división, y está dada como sigue:

$$\Delta = I(\text{Nodo padre}) - \sum_{i=1}^n \frac{N(a_i, v_i, p_i)}{N} I(a_i, v_i, p_i), \quad (5.12)$$

donde $I(\text{Nodo padre})$, es la medida de impureza de un nodo dado. N es el número total de datos en el nodo padre, n es el número de valores de atributos, y $N(a_i, v_i, p_i)$ es el número de registros del nodo hijo. La importancia de construir un AD, radica en obtener una condición de prueba que maximice la ganancia Δ . Y ya que $I(\text{Nodo padre})$ es el mismo para todas las condiciones, maximizar la ganancia es equivalente a minimizar las medidas de impureza del promedio ponderado de los nodos secundarios.

5.2. Resultados de los experimentos

Para el proceso de identificación de daños, los datos fueron separados en dos escenarios y dos estados al igual que para la clasificación para las Máquinas de Soporte Vectorial (ver Capítulo 4). El primer conjunto tiene dos atributos, los cuales son la posición y las aceleraciones pre-procesadas (ver Capítulo 3) y que son utilizados para entrenar al modelo. El conjunto de entrenamiento para construir el modelo también tiene un atributo que contiene las etiquetas de las clases, como Sin daño o Con daño para el estado 1. Por otro lado, el conjunto de datos restantes de la otra tabla es usada para probar y clasificar a los nuevos datos en nuevas clases, las cuales indiquen si la estructura tiene daño o no (ver Figura 5.14). Estos datos a los que nos referimos son los datos del escenario 2 del estado 2.

Algoritmo para la construcción de los Árboles de Decisión

Para entrenar el Árbol de Decisión, se tomó como referencia el algoritmo *CART* el cual trata de clasificar lo mejor posible los datos de entrada de acuerdo en el índice de impureza, el cual es el índice de Gini ya mencionado anteriormente. La entrada del algoritmo consiste en nuestro conjunto de datos de entrada X y un conjunto de atributos $A(a_i, v_i, p_i)$. El algoritmo trabaja recursivamente, seleccionando el mejor atributo para dividir los datos (ver paso 7) y expandiendo los nodos hoja del Árbol (ver pasos 11 y 12) hasta que se encuentre un criterio de parada (ver paso 1). El algoritmo está representado como sigue:

Datos de entrenamiento			
Aceleración	Velocidad	Posición	Clase
2.5983	2.8034	2.5885	Dañado
2.6081	2.8034	2.5934	Dañado
2.6081	2.7985	2.6032	Sin daño
2.6032	2.7985	2.6081	Dañado
2.6081	2.8034	2.5788	Dañado
2.6129	2.8278	2.5836	Sin daño
2.6081	2.8523	2.5690	Sin daño
2.6032	2.8425	2.5788	Dañado
2.6032	2.8278	2.5739	Dañado
2.6178	2.8230	2.5641	Sin daño
2.6178	2.8327	2.5788	Sin daño
2.6227	2.8327	2.5739	Dañado
2.6276	2.8278	2.5690	Sin daño
2.6374	2.8034	2.5592	Dañado
⋮	⋮	⋮	⋮

(a)

Datos para las pruebas			
Aceleración	Velocidad	Posición	Clase
2.6618	2.7546	2.7057	?
2.6618	2.7497	2.7057	?
2.6667	2.7497	2.7057	?
2.6667	2.7448	2.7106	?
2.6618	2.7497	2.7057	?
2.6618	2.7546	2.7057	?
2.6667	2.7448	2.7106	?
2.6667	2.7399	2.7106	?
2.6667	2.7448	2.7106	?
2.6667	2.7399	2.7057	?
⋮	⋮	⋮	⋮

(b)

Figura 5.14: Representación de los datos para el entrenamiento de los datos

ConstrucciónÁrbol (X, A)

1. **Si** $cond_parada(X, A) = verdadero$ **entonces**
2. $hoja = crearNodo()$.
3. $etiqueta.hoja = Clasificar(X)$.
4. retornar hoja.
5. **si-no**
6. $raíz = crearNodo()$.
7. $raíz.cond_prueba = encontrar_mejor_split(X, A)$.
8. Dado $V = \{v | v \text{ es un posible resultado de } raíz.cond_prueba\}$
9. **para** cada $v \in V$ **hacer**
10. $X_v = \{x | raíz.cond_prueba(x) = v \text{ y } e \in E\}$.
11. $hijo = ConstrucciónÁrbol(X_v, A)$.
12. agregar hijo como descendiente de raíz y etiquetar el nodo ($raíz \rightarrow hijo$) como v .
13. **fin para**
14. **fin si**
15. retornar raíz

En donde la función *crearNodo()* extiende la decisión del AD mediante la creación de un nuevo nodo. Un nodo en el AD tiene o bien; una condición de prueba denotado como *nodo.cond_prueba* o una etiqueta de clase, denotada como *etiqueta.nodo*.

También la función de *encontrar_mejor_split()* determina qué atributos deberían ser seleccionados como la condición de prueba para dividir los registros de entrenamiento. Como se señaló anteriormente, la elección de las condiciones de ensayo depende de qué medida de impureza se usa para determinar la exactitud de una división.

La función *Clasificar()* determina la etiqueta de clase que se debe asignar a un nodo hoja.

La función de *cond_parada()* se utiliza para terminar el proceso cuán grandes crecen los Árboles, comprobando si todos los registros tienen la misma etiqueta de clase o los mismos valores de los atributos. Otra forma de poner fin a la función recursiva es probar si el número de registros han caído por debajo de un umbral mínimo.

Clasificación del daño

Una vez visto el funcionamiento de algoritmo, toca clasificar el daño i.e. Con daño y Sin daño. El principal objetivo de aplicar esta técnica es el de obtener decisiones al final de los nodos hoja de los Árboles construidos.

En nuestro caso el conjunto de entrenamiento se utiliza para construir un modelo de clasificación, que posteriormente se aplica a la prueba de conjunto de nuevos datos, los cuales consta de registros desconocidos. Para la resolución de nuestro problema utilizamos este enfoque para construir el modelo de entrenamiento para predecir si nuevos datos presentan o no daño. Estos datos elegidos provienen del escenario 1 del estado 1 que es donde se presenta el daño en P2 para las grabaciones del acelerómetro A2. Para la primera prueba se decidió particionar los datos en 50 % y 50 %, es decir 50 % del escenario 1 del estado 1 para entrenar el modelo y el otro 50 % de los datos provienen del escenario 2 estado 1 para probar el modelo. En la Figura 5.16, se muestra el AD generado con sus respectivas reglas, en el cual cada nodo del AD tiene una condición así como la probabilidad de pertenecer a la clase, y cada arco (rama) representa un posible valor dependiendo de la respuesta (sí/no) de la condición. Este primer AD se encuentra sin podar ya que tiene una profundidad máxima de hasta 30 nodos.

De igual manera se realizó otro modelo con los datos provenientes del escenario 1 del estado 1 que es donde se presenta el daño en P2 para las grabaciones del acelerómetro A2, así como los datos del escenario 2 del estado 1. Para esta prueba se decidió particionar los datos en un enfoque con 70 % y 30 %, es decir 70 % del total de los datos del escenario 1 y 2 para entrenar el modelo y el otro 30 % restante para probar el modelo. En la Figura 5.15, se muestra el AD generado con sus respectivas reglas, en donde así mismo se obtiene el porcentaje de de probabilidad de que el el evento ocurra (i.e. que presente daño [Damaged] o que no presente daño [Undamaged]).

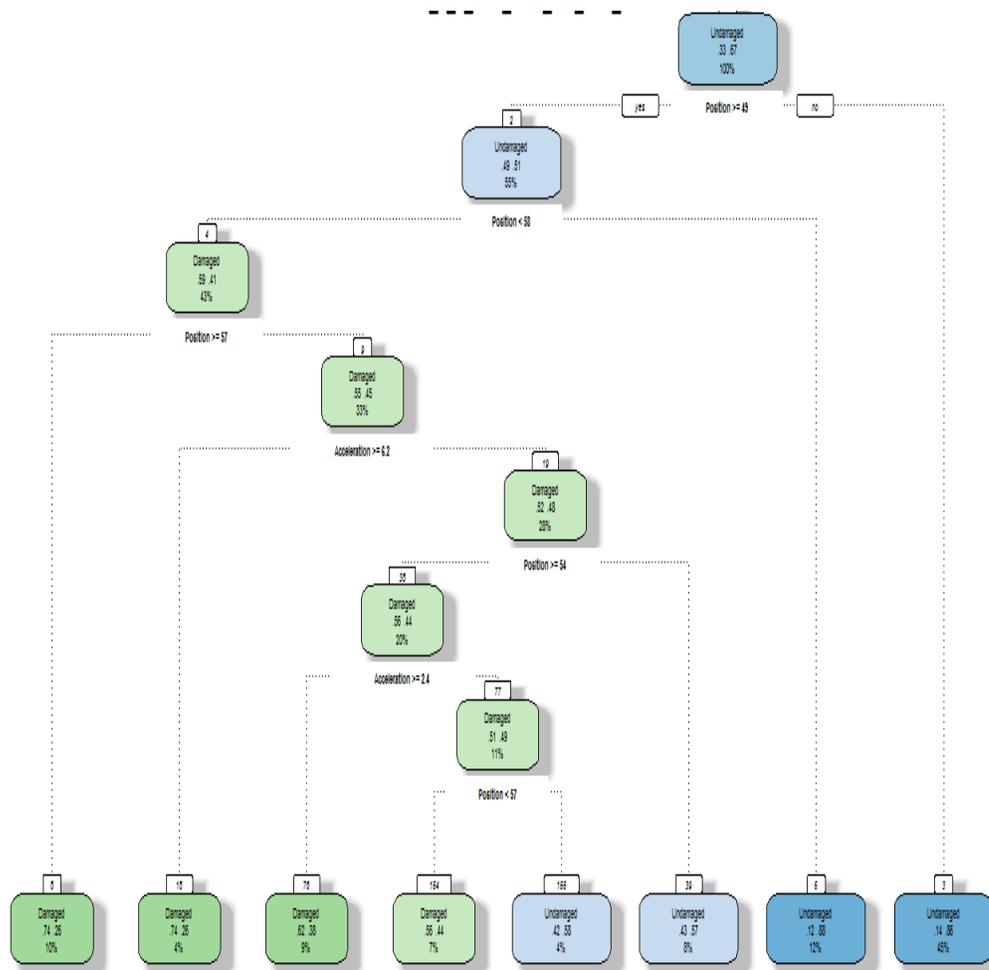


Figura 5.15: Árbol construido para la clasificación del daño

En el siguiente Árbol (ver Figura 5.16), es un AD construido, el cual ya está optimizado puesto, que al utilizar el valor del índice de Gini se asegura que tiene el mayor grado de división posible (de acuerdo con la biblioteca utilizada e1071 [ver Capítulo 3]). Para nuestro escenario, resultó ser que el modelo construido logra predecir con bastante certeza si el edificio presenta daño o no.

En la Tabla 5.1 se muestra la matriz de confusión de los datos reales, en comparación con los datos predichos, se puede observar que tiene un buen nivel de predicción con un 69% de aciertos.

Así mismo, se puede obtener una representación a través de las reglas generadas, las cuales se presentan de manera escrita y aunque son un poco más difíciles de leer aportan un gran valor para determinar las decisiones.

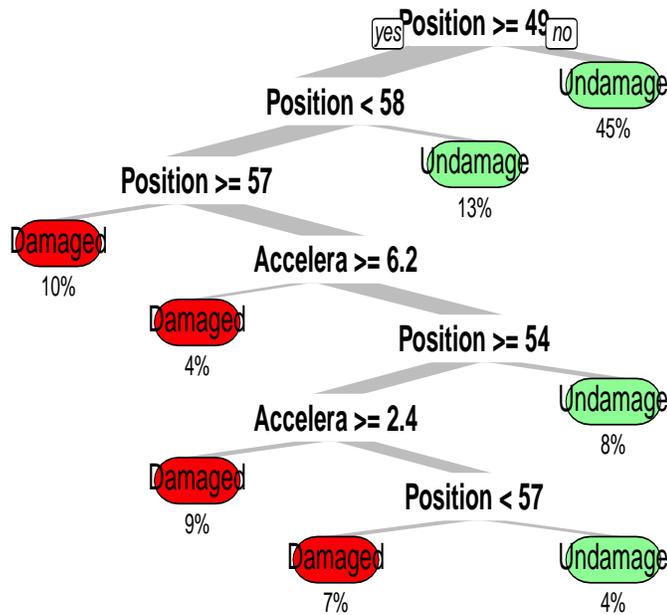


Figura 5.16: Reglas generadas para la clasificación del daño

Clase	Dañado	No dañado
Dañado	12206	7794
Sin dañado	12403	27597
P		recisión:69.33 %

Tabla 5.1: Matriz de confusión para la clasificación de daño

Regla: 5 [Clase=Sin Daño cover=7491 (12 %) prob=0.88]
 Posición>=48.72
 Posición>=57.59

Regla: 3 [Clase=Sin Daño cover=26861 (45 %) prob=0.86]
 Posición< 48.72

Regla: 76 [Clase=Con Daño cover=5393 (9 %) prob=0.38]
 Posición>=48.72
 Posición< 57.59
 Posición< 57.25
 Aceleración< 6.194
 Posición>=54.38

Aceleración ≥ 2.383

Severidad del daño

En este análisis también se clasificó el daño por severidad, i.e. que se le asignó una etiqueta a cada porción de los datos con: Sin daño (*Undamaged*), Levemente dañado (*Slight*), Dañado (*Damaged*) y Severamente dañado (*Severe*). Para asignar correctamente cada clasificación se dividieron los datos y se sacó un promedio, por ejemplo; aquellos datos que no presentaban daño se tomó el valor más alto de las aceleraciones menos el valor más bajo y se promediaron entre los 4 estados, para así asignar las etiquetas que representan sin daño y levemente dañado, de igual manera para las etiquetas con daño y severamente dañado. En la Figura 5.17 se muestra el AD resultante para esta prueba. Del mismo modo, se presenta su matriz de confusión en la Tabla 5.2.

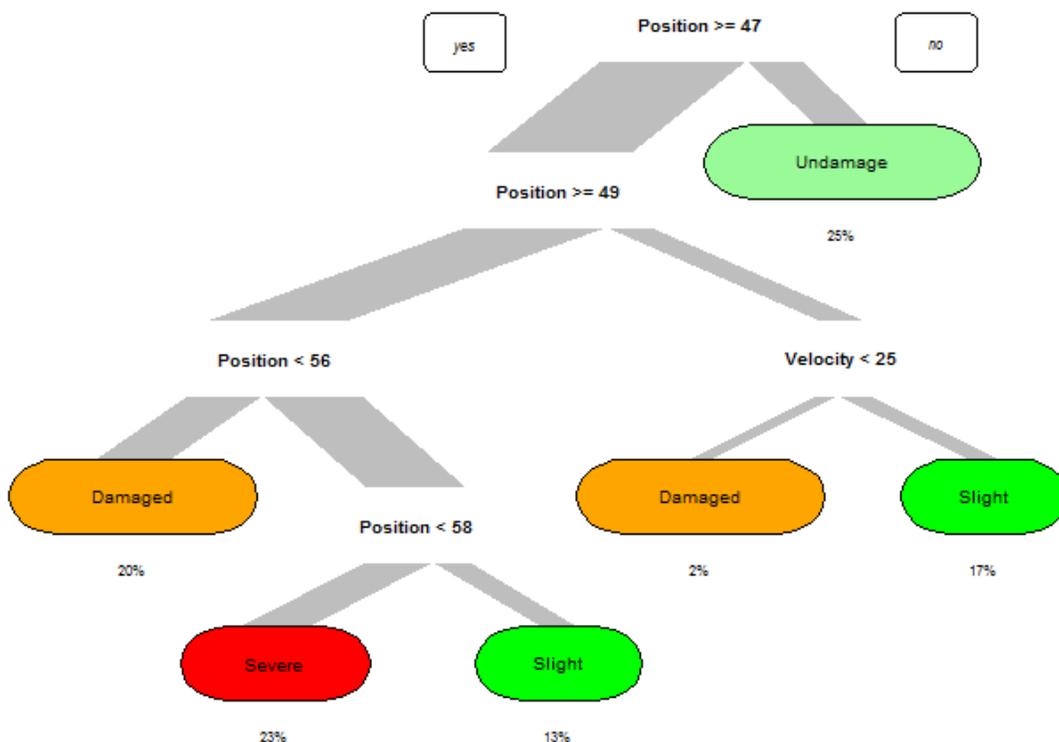


Figura 5.17: Reglas generadas para la identificación del daño por niveles

Clase	Dañado	S.dañado	Daño.leve	No dañado
Dañado	7579	46	229	2147
S.dañado	4581	5417	1	0
Daño.leve	477	8768	9999	756
No dañado	7455	458	221	11866
P		recisión: 67.20%		

Tabla 5.2: Matriz de confusión para la clasificación de daño por severidad

A continuación, en la Figura 5.18, se presenta un Árbol más, cambiando los parámetros, tales como la profundidad máxima y la división mínima (estos parámetros se encuentran en la función del paquete e1071):

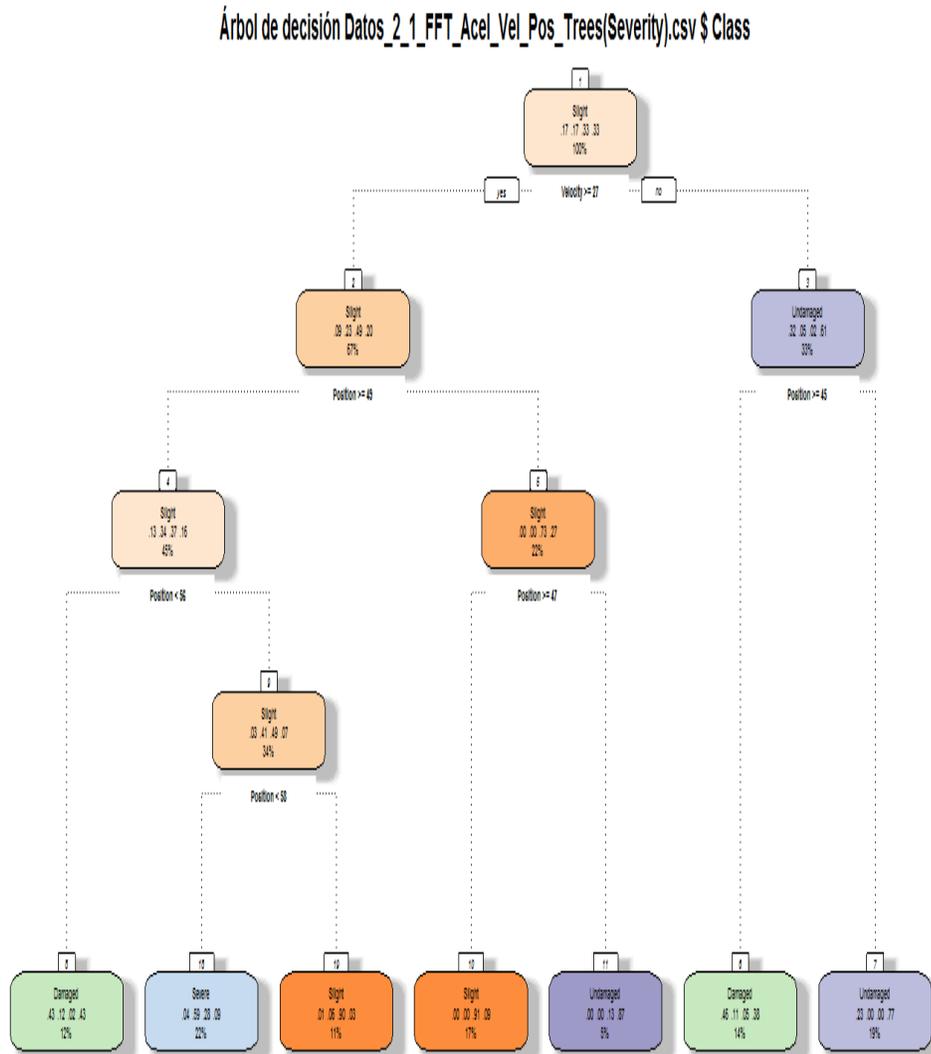


Figura 5.18: Árbol construido con la modificación de los parámetros

Algunas de las reglas generadas para este AD, son las siguientes:

Regla: 7 [Class=Undamaged cover=11477 prob (19 %)]
Velocidad < 26.62
Posición < 45.22

Regla: 6 [Class=Damaged cover=8238 prob (14 %)]
Velocidad < 26.62
Posición >= 45.22

Regla: 11 [Class=Undamaged cover=3059 prob (5 %)]
Velocidad >= 26.62
Posición < 49.15
Posición < 46.71

Regla: 18 [Class=Severe cover=13405 prob (22 %)]
Velocidad >= 26.62
Posición >= 49.15
Posición >= 56.14
Posición < 57.59

En este análisis también se obtuvo el índice de Gini para conocer que variables aportan mayor peso para construir el modelo. En la Figura 5.19 se muestra la representación de los valores.

Clasificación del daño por niveles

El AD construido 5.16, solo muestra si el edificio tiene daño o no. Por tanto, se necesita conocer en qué nivel o niveles está localizado el daño, para éste propósito los datos se etiquetaron con su respectivo nivel, i.e. los datos grabados de P1 como: Nivel 1 (Level1), P2 se etiquetó como Nivel 2 (Level2) y los del tercer piso (P3), como Nivel 3 (Level3). De igual manera, el modelo fue entrenado, el primer AD construido, solo tiene los atributos de la velocidad y la posición (ver Figura 5.20). El segundo AD, no solo se utilizaron los atributos de la posición y la aceleración, sino que en esta ocasión también se utilizó la velocidad como atributo, además de la clase que ya tiene las etiquetas. El AD generado se muestra en la Fig 5.21.

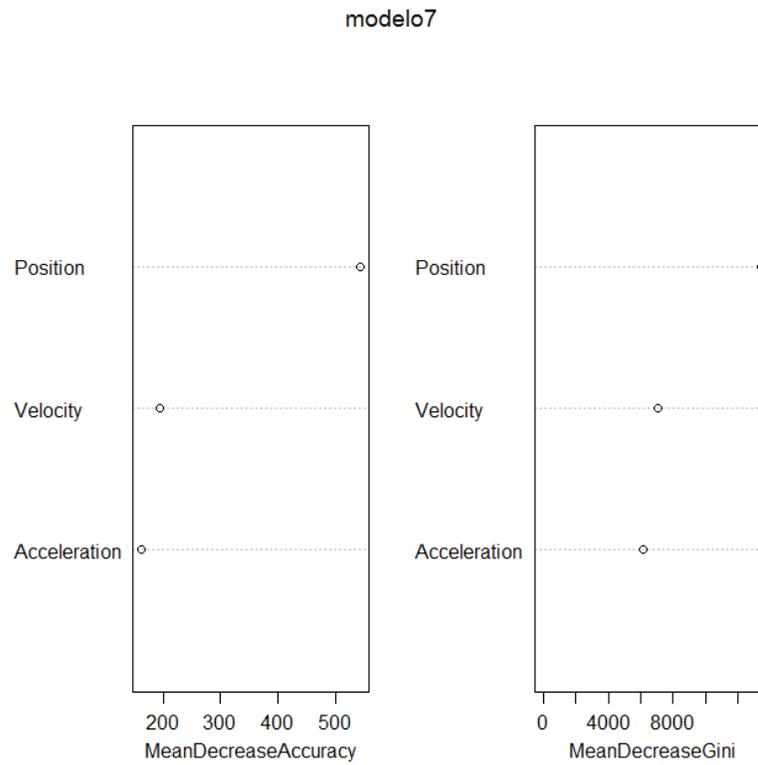


Figura 5.19: Importancia de las variables para el modelo clasificador

En la Tabla 5.3, se muestra la matriz de confusión para la clasificación por niveles. El Árbol mostrado, es generado con el cambio de los parámetros, i.e con el cambio la velocidad y posición respectivamente:

Clase	Nivel 1	Nivel 2	Nivel 3
Nivel 1	19981	18	1
Nivel 2	2802	17162	36
Nivel 3	2530	761	16709

P recisión: 67.17 %

Tabla 5.3: Matriz de confusión para la clasificación del daño por niveles

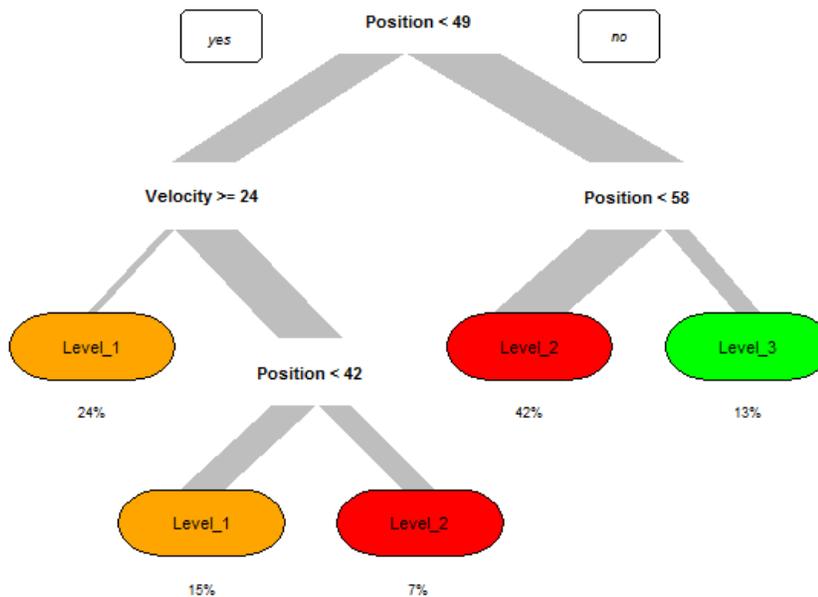


Figura 5.20: Reglas generadas para la identificación del daño por niveles (variables posición y velocidad)

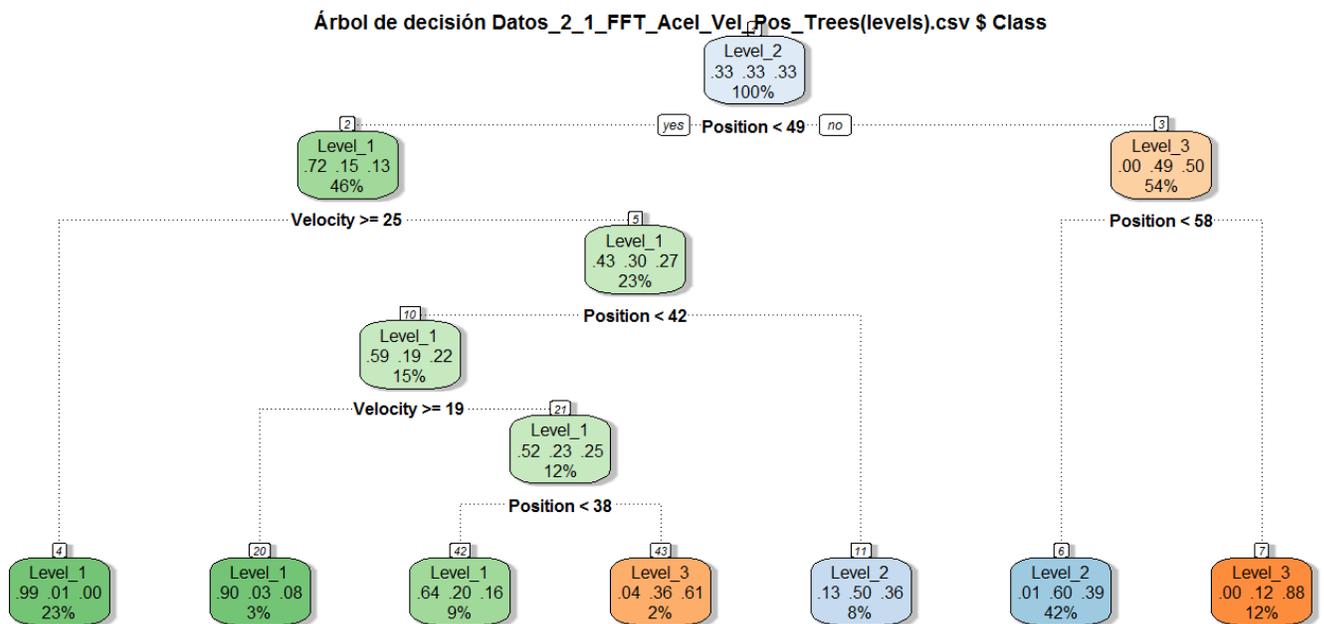


Figura 5.21: Distinto Árbol modificando parámetros

Comparación con otras técnicas

De las comparaciones con otros modelos (K-NN y ANN) solo se tiene el modelo que identifica la presencia o ausencia de daño, puesto que las SVM's y las ANN's solo son capaces de realizar la clasificación de dos clases.

Para el primer experimento se obtuvieron los resultados que se observan en la tabla de abajo. En el cual los datos fueron divididos 50 % y 50 %, i.e. 50 % del escenario 1 del estado 1, para entrenar el modelo y el otro 50 % proveniente del escenario 2 del estado 1, para probar el modelo construido.

	Precisión %	Error %	ROC
Árbol de Decisión	66.33	33.6	0.68
KNN	64.84	35.16	0.71
Redes Neuronales	64.81	35.19	0.67

De los resultados obtenidos, se observa que los AD's resultan ser después de las SVM's los modelos con mejor precisión a la hora de la clasificación, con un 66.33 % de precisión y con un área bajo la curva ROC, de 0.68 o el 68 %.

De igual manera se realizó otro modelo con los datos provenientes del escenario 1 del estado 1 de A2, así como los datos del escenario 2 del estado 1. Para esta prueba se decidió particionar los datos en un enfoque con 70 % y 30 %, es decir 70 % del total de los datos del escenario 1 y 2 para entrenar el modelo y el otro 30 % restante para probar el modelo. La tabla de abajo muestra los resultados obtenidos:

	Precisión %	Error %	ROC
Árbol de Decisión	61.12	38.88	0.64
KNN	55.18	44.82	0.61
Redes Neuronales	61.0	39.0	0.64

En otra de las pruebas, se tomaron el 100 % de los datos del escenario 1 del estado 1 y 2 para entrenar el modelo. Así mismo, se tomaron el otro 100 % de los datos del escenario 2 para probar el modelo. Los resultados obtenidos se observan la tabla siguiente:

Árbol de Decisión	65.04	34.91	0.69
KNN	58.22	41.78	0.66
Redes Neuronales	62.12	37.88	0.70

En otra de las pruebas realizadas, se combinaron los tres atributos. Se tomaron los valores del escenario 1 del estado 1 para entrenamiento y los valores del escenario 2 del primer y segundo estado para prueba. Para este modelo se obtuvieron los siguientes resultados como se ven en la tabla de la parte de abajo.

	Precisión %	Error %	ROC
Árbol de Decisión	74.63	25.37	0.77
KNN	68.03	31.97	0.73
Redes Neuronales	71.83	28.17	0.77

Como se puede observar los resultados obtenidos de los AD, muestran buenos resultados para la clasificación del daño.

5.3. Conclusión

Las técnicas desarrolladas para la construcción de Árboles de Decisión son computacionalmente rápidas, por lo que es posible construir distintos modelos, incluso cuando el tamaño del conjunto de entrenamiento es muy grande. Además, la clasificación de un nuevo individuo de prueba es extremadamente rápido, con un peor caso de complejidad de $O(w)$, donde w es la profundidad máxima del Árbol. Otra de las principales características de los Árboles es que se construyen mediante reglas del tipo *si – entonces* relativamente fáciles de interpretar.

Un aspecto importante es que, la presencia de atributos redundantes no afecta negativamente a la exactitud de AD, ya que solo se utiliza uno de los dos atributos con mayor ganancia de información para dividir una vez. Sin embargo, si el conjunto de datos contiene muchos atributos irrelevantes para la tarea de clasificación, algunos de los atributos irrelevantes pueden ser elegidos accidentalmente durante el proceso de construcción del Árbol, lo que resulta en un Árbol de Decisión redundante. Por tal motivo, es necesario utilizar índices de impureza para determinar lo mejor posible la exactitud del Árbol.

Finalmente, podemos decir que los Árboles de Decisión resultaron ser un buen clasificador, ya que no solo son capaces de clasificar entre la presencia o no de daño en el edificio, sino que también son capaces de ser algoritmos multi-clase, con los cuales se puede determinar la localización y severidad del daño.

Capítulo 6

Conclusiones y Trabajo futuro

En este capítulo, se resumen los resultados principales a los que se llegaron después de realizar los experimentos. Así mismo, se hacen algunas recomendaciones que se considera deben ser tomadas en cuenta, de acuerdo a las experiencias obtenidas en la realización de esta tesis. Por último, se incita al lector a seguir trabajando en el tema de la detección de daños en edificios y a proponer nuevos paradigmas en este ámbito.

6.1. Conclusiones

En este trabajo de tesis, se presentó una metodología basada en Minería de Datos para poder llevar a cabo el proceso de detección del daño, dentro del paradigma del SHM. Primero, en el Capítulo 3 se describió la estructura de prueba en la que se realizaron los experimentos, así como las herramientas, tanto de *hardware* como de *software* que fueron utilizadas. Una vez obtenidos los datos de las aceleraciones, se realizó un análisis estadístico, con el fin de obtener información del edificio. En este análisis, se obtuvieron los cuatro momentos estadísticos (media, desviación estándar, sesgo y curtosis), en donde se observaron algunos indicios de la presencia de daño en la estructura, sin embargo, la variación de los valores de la media y el sesgo con respecto a los dos estados comparados (Con Daño y Sin Daño), pueden ser muy pequeños como para aseverar que algo ocurre con ellos. Por otro lado, la curtosis ofrece un mejor indicio de que algo pudiera ocurrir con la medición de los datos con daño.

De los resultados obtenidos de las distribuciones de probabilidad, las gráficas mostraron que cuando los datos se vuelven atípicos, tienden a desajustarse de la Normal. Con lo cual, podemos inferir que los datos obtenidos se comportan de manera irregular, en nuestro caso representa la presencia del daño. Cabe destacar, que este análisis debe ser complemento de técnicas más robustas como las Máquinas de Soporte Vectorial, los Árboles de Decisión o las Redes Neuronales.

Posteriormente, se realizó el pre-procesamiento de los datos de acuerdo al paradigma de la Minería de Datos. El pre-procesamiento consistió en transformar y filtrar dichos datos con la Transformada Rápida de Fourier (FFT), con el propósito de trabajar en el dominio de la frecuencia, lo que implicó que fuese fácil de manejar vs. si estuvieran en el dominio del tiempo. De los resultados obtenidos, se observó que cuando ocurre el daño en el edificio, las frecuencias son un tanto mayores

Después, en el Capítulo 4 se implementaron las Máquinas de Soporte Vectorial para clasificar el daño. Ésta técnica mostró buen desempeño, ya que pudo predecir nuevos objetos con un alto porcentaje de aciertos. Así mismo, distintos parámetros fueron ajustados para llegar a construir un clasificador óptimo para obtener los mejores resultados. Sin embargo, se tiene que tomar en cuenta que debido a que es un problema cuadrático (QP), el número de iteraciones que tiene que llegar a realizar el algoritmo para llegar a un nivel de convergencia es alto.

Finalmente, en el Capítulo 5 se construyeron diferentes Árboles de Decisión, no solo para clasificar el daño, sino también para localizar y verificar el nivel de daño que sufrió la estructura tras las vibraciones. La aplicación de ésta técnica tiene grandes beneficios, uno de ellos es que permite observar de manera gráfica las decisiones tomadas tras entrenar al modelo, y así predecir nuevos escenarios. Además, el aprendizaje de los Árboles de Decisión es sencillo y fácil de implementar, y que mejores resultados puede ofrecer. Sin embargo, es de hacer notar que no todas las variables que están involucradas en el análisis pueden ofrecer el mismo peso a la hora de realizar el entrenamiento, algunas incluso aportan muy poca información, con lo cual pueden ser prescindibles para realizar el análisis.

De las comparaciones de los modelos, se puede concluir que; en general las dos técnicas propuestas tienen un alto índice de aciertos a la hora de la clasificación. Demostrando que el uso de métodos supervisados de la Minería de Datos, resultan ser una poderosa herramienta para el uso en el área del SHM. Así mismo, otras de las técnicas (Redes Neuronales y el K-NN) con las que fueron comparadas las SVM's y los AD's, en la mayoría de las veces; se obtuvieron un alto porcentaje de aciertos para la clasificación del daño. Quedando de manifiesto que pueden ser una buena opción para futuras comparaciones.

Consejos para un buen monitoreo del estado de las estructuras

A continuación, se proponen algunos consejos que podrían ser de utilidad (de acuerdo con las experiencias obtenidas) para llevar a cabo un buen monitoreo en las estructuras:

- En la medida de lo que sea posible, se necesita de la instalación de otro tipo de sensores (alámbricos y/o inalámbricos), con el fin de recabar mayor número de datos y tener la posibilidad de comparar nuevos parámetros en la búsqueda de nuevo conocimiento.
- Para la generación de daño, se deben de probar diferentes escenarios (v.g. poner y/o

quitar masa en determinados niveles, utilizar otras señales de excitación para la generación de ondas), con el fin de estudiar distintos comportamientos de la estructura.

- Para el entrenamiento de los modelos, se sugiere probar con distintos parámetros de los algoritmos (v.g. probar diferentes *kernels*, profundidad de poda), con el fin de mejorar el nivel de precisión.

Al final de esta tesis, se cumplieron los objetivos planteados en el Capítulo 1, ya que las técnicas planteadas se desempeñaron de buena manera para la detección, localización y severidad del daño en el edificio. Así mismo, se puede concluir que la metodología desarrollada en este trabajo puede ser aplicada en cualquier otro edificio de dimensiones considerables y/o de morfología parecida a la utilizada en este trabajo.

Actualmente no existe un método para determinar si los edificios son seguros para su ocupación después de algún incidente. Tal vez, algún día el SHM llegue a evolucionar a tal punto que reduzca significativamente la incertidumbre asociada con la seguridad de la estructura, para garantizar la seguridad de sus ocupantes.

6.2. Trabajo a futuro

Aún queda mucho trabajo por recorrer en ésta área del conocimiento. Ya hemos visto algunas de las técnicas más utilizadas por otros investigadores además de las nuestras para poder definir el daño en estructuras. Sin embargo, quedan algunas cuestiones que no podemos ignorar tales como; ¿en qué lugar se deben colocar los sensores para una mejor recolección de datos?, o; ¿cuál probabilidad de éxito tenemos para detectar el daño dado ciertas condiciones?. Éstas y otras preguntas, dan paso a seguir avanzando en el área del SHM, con el fin de solventar y mejorar otros aspectos además de detectar el daño.

Además de las interrogantes surgidas anteriormente, se plantean algunos puntos que en un futuro es factible que se lleven a cabo, con el fin de complementar este trabajo de tesis.

- En un futuro, se planea introducir esta metodología para detectar posibles fallas en otros edificios, mediante una red de sensores que constantemente estén monitoreando el sistema de una manera inteligente y así ser capaz de alertar a un operador en caso de que el sistema presente algún tipo de anomalía.
- Se tiene en mente la utilización de teléfonos móviles, con el fin de obtener las aceleraciones de las vibraciones que presenta el edificio, ya que estos dispositivos actualmente son muy comunes y cuentan con un acelerómetro, el cual puede ser aprovechado para recabar información de la estructura que se requiera estudiar.
- Debido a que se pretende combinar otro tipo de sensores para el edificio. Se planea utilizar otras técnicas como el PCA combinado con nuestras técnicas para obtener buenos resultados.

Finalmente podemos apuntar a nuevas tendencias dentro del SHM, ya que con el avance en las nuevas tecnologías de hardware y software cada vez es indispensable que existan nuevos métodos para la detección de daños dentro del SHM. Y aunque existen al día de hoy una gran variedad de técnicas, todavía queda investigar para mejorar la efectividad, el rendimiento y la exactitud de las mismas.

Bibliografía

- [1] C. R. Farrar, K. Worden, *Structural health monitoring a machine learning perspective*, 1st Edition, John Wiley & Sons, Ltd, United Kingdom, 2013.
- [2] C. R. Farrar, H. Sohn, S. W. Doebling, *Structural health monitoring at los Alamos national laboratory*, Research Article, 2000.
- [3] H. Sohn, C. R. Farrar, F. M. Hemez, D. D. Shunk, D. W. Stinernes, B. R. Nadler, J. J. Czarnecki, *A review of structural health monitoring literature: 1996–2001*, Report, 2004.
- [4] E. Figueiredo, G. Park, J. Figueiras, C. Farrar, K. Worden, *Structural health monitoring algorithm comparisons using standard data sets*, Los Alamos National Laboratory, Report, 2009.
- [5] O. S. Salawu, *Detection of structural damage through changes in frequency: A review*, *Engineering Structures*, Vol. 19, pp. 718-723, 1997.
- [6] S. W. Doebling, C. R. Farrar, *Estimation of statistical distributions for modal parameters identified from averaged frequency response function data*, Research Article, 2000.
- [7] S. W. Doebling, C. R. Farrar, M. B. Prime, *Summary review of vibration-based damage identification methods*, *The Shock and Vibration Digest*, Vol. 30, pp. 91-105, 1998.
- [8] W. Fan, P. Quiao, *Vibration-based damage identification methods: A review and comparative study*, Department of Civil and Environmental Engineering and Composite Materials and Engineering Center, Washington State University, Vol. 10, pp. 29-83, 2011.
- [9] M. Algaba, M. Solís, P. Galvín, *Detección de daño en estructuras mediante la aplicación de la transformada Wavelet a su deflexión estática*, Research Article, 2010.
- [10] B. H. Koh, S. J. Dyke, *Structural health monitoring for flexible bridge structures using correlation and sensitivity of modal data*, *Computers and Structures*, Vol. 85, pp. 117–130, 2007.

- [11] S. Saitta, B. Raphael, I. F. C. Smith, *Data mining techniques for improving the reliability of system identification*, Advanced Engineering Informatics Vol. 19, pp. 289–298, 2005.
- [12] S. Saitta, P. Kripakaran, B. Raphael, I. F. C. Smith, *Improving system identification using clustering*, Journal of Computing in Civil Engineering, Vol. 22, pp. 292-302, 2008.
- [13] S. Saitta, B. Raphael, I. F. C. Smith, *Combining two data mining methods for system identification*, Technical Report, pp. 606-614, 2006.
- [14] S. S. Sandhu, R. Kanapady, K. K. Tamma, C. Kamath, V. Kumar, *Damage prediction and estimation in structural mechanics based on data mining*, Research Article, 2001.
- [15] R. Kanapady, A. Lazarevic, *Data mining for damage detection in engineering structures*, Book Chapter, ENCYCLOPEDIA OF DATA WAREHOUSING AND MINING, pp. 245-248, 2005.
- [16] S. H. Liao, P. H. Chu, P. Y. Hsiao, *Data mining techniques and applications – A decade review from 2000 to 2011*, Expert Systems with Applications, Vol. 39, pp. 11303-11311, 2012.
- [17] S. R. Gunn, *Support vector machines for classification and regression*, Technical Report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, May, 1998.
- [18] A. Lazarevic, R. Kanapady, C. Kamath, *Effective localized regression for damage detection in large complex mechanical structures*, Industry/Government Track, Research Article, 2010.
- [19] P. Morreale, S. Holtz, A. Goncalves, *Data mining and analysis of large scale time series network data*, Research Article, 2013.
- [20] P. Lu, B. M. Phares, T. J. Wipf, J. D. Doornink, *A bridge structural health monitoring and data mining system*, Research Article, 2007.
- [21] J. C. Bezdek, R. Ehrlich, W. Full, *FCM: The fuzzy c-means clustering algorithm, computer and geosciences*, Vol. 10, pp. 191-203, 1984.
- [22] P. Beena, R. Ganguli, *Structural damage detection using fuzzy cognitive maps and Hebbian learning*, Applied Soft Computing, Vol. 11, pp. 1014-1020, 2011.
- [23] H. Kosorus, J. Hönlgl, J. Küng, *Using R, WEKA and RapidMiner in time series analysis of sensor data for structural health monitoring*, Research Article, 2011.
- [24] K. Worden, G. Manson, *The application of machine learning to structural health monitoring*, Phil. Trans. R. Soc. A, Vol. 365, pp. 515–537, 2007.
- [25] A. Rytter, *Vibrational based inspection of civil engineering structures*, PhD. Thesis, University of Aalborg, April, 1993.

-
- [26] P. N. Tan, M. Steinbach, V. Kumar, *Introduction to data mining*, 1st Edition, Addison-Wesley, USA, 2005.
- [27] A. R. Webb, K. D. Copsey, *Statistical pattern recognition*, 3rd Edition, John Wiley & Sons, Ltd, United Kingdom, 2011.
- [28] S. Theodoridis, K. Koutroumbas, *Pattern recognition*, 2nd Edition, Elsevier, USA, 2003.
- [29] D. Graupe, *Principles of artificial neural networks*, 2nd Edition, World Scientific Publishing Co. Pte. Ltd, USA, 2007.
- [30] C. M. Bishop, *Pattern recognition and machine learning*, 1st Edition, Springer Science+Business Media, LLC, USA, 2006.
- [31] Q. Zhou, Y. Ning, Q. Zhou, L. Luo, *Structural damage detection method based on random forest and data fusion*, Article, doi:10.1177/1475921712464572, 2012.
- [32] N. A. J. Lieven, D. J. Ewins, *Spatial correlation of mode shapes: The coordinate modal assurance criterion (COMAC)*, In: Proceedings of the 6th international modal analysis conference, pp. 690-695, 1988.
- [33] L. E. Mujica, M. Ruiz, F. Pozo, J. Rodellar, A. Güemes, *A structural damage detection indicator based on principal component analysis and statistical hypothesis testing*, Smart Materials and Structures, Vol. 23, 2013 doi:10.1088/0964-1726/23/2/025014.
- [34] D. A. Tibaduiza, L. E. Mujica, J. Rodellar, *Damage classification in structural health monitoring using principal component analysis and self-organizing maps*, Structural Control and Health Monitoring, 2012 doi: 10.1002/stc.1540.
- [35] L. E. Mujica, M. Ruiz, F. Pozo, J. Rodellar, A. Güemes, *A structural damage detection indicator based on principal component analysis and statistical hypothesis testing*, Smart Materials and Structures, Vol. 23, doi:10.1088/0964-1726/23/2/025014, 2013.
- [36] R. J. Allemang, D. L. Brown, *A correlation coefficient for modal vector analysis*, In: Proceedings of the 1st international modal analysis conference, pp. 110-116, 1982.
- [37] M. Lemistre, D. Baleagas, *Structural health monitoring system based on diffracted lamb wave analysis by multiresolution processing*, Smart Materials and Structures, Vol. 10, pp. 504-511, 2001.
- [38] H. Kim, H. Melhem, *Fourier and Wavelet Analyses for Fatigue Assessment of Concrete Beams*, Experimental Mechanics, Vol. 43, pp. 131-140, 2003.
- [39] M. A. Prada, J. Toivola, J. Kulla, J. Hollmén, *Three-way analysis of structural health monitoring data*, Neurocomputing, Vol. 80, pp. 119-128, 2012
- [40] S. Gupta, A. Khatkhate, A. Ray, E. Keller, *Identification of statistical patterns in complex systems via symbolic time series analysis*, Vol. 45, pp. 477-490, 2006.

- [41] D. W. Allen, S. Castillo, A. L. Cundy, C. R. Farrar, R. E. McMurry, *Damage detection in building joints by statistical analysis*, Los Alamos National Laboratory, Technical Report, 2010.
- [42] R. Yao, S. N. Pakzad, *Autoregressive statistical pattern recognition algorithms for damage detection in civil structures*, Mechanical Systems and Signal Processing, Vol. 31, pp. 355–368, 2012.
- [43] O. R. de Lautour, P. Omenzetter, *Damage classification and estimation in experimental structures using time series analysis and pattern recognition*, Mechanical Systems and Signal Processing, Vol. 24, pp. 1556-1569, 2010.
- [44] C. J. Li, A. Ray, *Neural network representation of fatigue damage dynamics*, American Control Conference, Vol. 5, pp. 3284-3288, 1995.
- [45] S. Thenozhi, *Control difuso de estructuras de edificios sujetas a vibraciones inducidas por los sismos o el viento*, PhD. Thesis, Center for Research and Advanced Studies of the National Polytechnic Institute, Department of Automatic Control, March, 2014.
- [46] S. Thenozhi, W. Y. Liu, A. L. Chau, X. L. Zhang, *Structural health monitoring of tall buildings with numerical integrator and convex-concave hull classification*, Research Article, 2012.
- [47] A. E. Zonst, *Understanding the FFT*, 1st Edition, Citrus Press, USA.
- [48] W. Li, H. Zhu, L. Ding, H. Luo, *Structural damage recognition by grouped data based on principal component analysis theory*, Automation in Construction, Vol. 22, pp. 258-270, 2012.
- [49] J. A. E. Sánchez, *Detección de daño en edificios mediante sus características dinámicas*, Tesis de doctorado, Ingeniería Civil (UNAM), 2007.
- [50] A. E. Aktan, K. L. Lee, C. Chuntavan, T. Aksel, *Modal testing for structural identification and condition assessment of constructed facilities*, Research Article, 1987.
- [51] N. Lin, C. Qun, *Structural health monitoring and damage detection using neural networks technique*, Research Article, 2013.
- [52] R. Rodríguez, J. A. Escobar, R. Gómez, *The baseline stiffnesses method for damage identification without baseline modal parameters and damage assessment of a reinforced concrete building*, Research Article, 2008.
- [53] Y. H. Teo, W. K. Chiu, F. K. Chang, N. Rajic, *Optimal placement of sensors for subsurface fatigue crack monitoring*, Theoretical and Applied Fracture Mechanics Vol. 52, pp. 40–49, 2009.
- [54] A. R. Ancona Lazcano, R. S. Estrada, S. A. Zamora Castro, F. M. Martínez, *Evaluación de métodos de detección de daño en estructuras mediante el uso de vibraciones*, Article from Seismic Society of Mexican Engineering, 2010.

-
- [55] S. R. Shiradhonkar, M. Shrikhande, *Seismic damage detection in a building frame via finite element model updating*, Computers and Structures, Vol. 89, pp. 2425-438, 2011.
- [56] E. Forstner, H. Wenzel, *The application of data mining in bridge monitoring projects*, Report, 2011.
- [57] F. N. Catbas, M. Susoy, D. M. Frangopol, *Structural health monitoring and reliability estimation: long span truss bridge application with environmental monitoring data*, Engineering Structures, Vol. 30, pp. 2347-2359, 2008.
- [58] J. D. Achenbach, *Structural health monitoring – What is the prescription?*, Mechanics Research Communications, Vol. 36, pp. 137-142, 2008.
- [59] L. Ljung, *Perspectives on system identification*, Annual Reviews in Control Vol. 34, pp. 1–12, 2010.
- [60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, *The WEKA data mining software: An update, sigkdd explorations*, Vol. 11, 2009.
- [61] P. Miroslav , B. Michal , H. Tomás, *Modal assurance criterion*, Procedia Engineering, Vol. 48, pp. 543-548, 2012.
- [62] N. Baghiee, M. R. Esfahani , K. Moslem, *Studies on damage and FRP strengthening of reinforced concrete beams by vibration monitoring*, Engineering Structures, Vol. 31, pp. 875-893, 2009.
- [63] P. M. M. Reda-Taha, J. Lucero, *Damage identification for structural health monitoring using fuzzy pattern recognition*, Engineering Structures, Vol. 27, pp. 1774-1783, 2005.
- [64] S. Kaewunruen, A. M. Remennikov, *Application of experimental modal testing for estimating dynamic properties of structural components*, Vol. 21, pp. 134-139, 2005.
- [65] S. R. Shiradhonkar, M. Shrikhande, *Seismic damage detection in a building frame via element model updating*, Computers and Structures, Vol. 89, pp. 2425-2438, 2011.
- [66] S. Tesfamariam, Z. Liu, *Earthquake induced damage classification for reinforced concrete buildings*, Structural Safety, Vol. 32, pp. 54-164, 2010.
- [67] Y. J. Yan, L. Cheng, Z. Y. Wu, L. H. Yam, *Development in vibration-based structural damage detection technique*, Mechanical Systems and Signal Processing, Vol. 21, pp. 2198–2211, 2007.

