

**ANÁLISIS COMPARATIVO ENTRE REDES SOM Y REDES BAYESIANAS
APLICADO A LA DETECCIÓN DE FALLAS EN ESTRUCTURAS**

**EFRAÍN GUILLERMO MARIOTTE PARRA
MAURICIO PEDROZA TORRES**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2011

**ANÁLISIS COMPARATIVO ENTRE REDES SOM Y REDES BAYESIANAS
APLICADO A LA DETECCIÓN DE FALLAS EN ESTRUCTURAS**

**EFRAÍN GUILLERMO MARIOTTE PARRA
MAURICIO PEDROZA TORRES**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO MECÁNICO**

**DIRECTORES:
JABID QUIROGA
MAGISTER, INGENIERO MECÁNICO**

**SILVIA OVIEDO
INGENIERA ELECTRÓNICA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA MECÁNICA
BUCARAMANGA**

2011

A mi creador porque me lo dio todo sin merecerlo

*A la arquitecta de mi vida,
Por guardarse los mejores diseños para planear mi vida.*

*A la damita de mi corazón, mi fuente de bondad pura,
No sería la mitad de persona que soy sin ti*

*A la ingeniera de mi vida,
Por enseñarme la fórmula de la felicidad
Y comprobándola uniéndola su vida conmigo*

*A mi segunda madre
Por enseñarme que la familia son las personas que amas*

EFRAÍN GUILLERMO MARIOTTE PARRA

Por ustedes

Mamá, gracias por el cariño.

Papá, gracias por la confianza.

Hermano, gracias por la alegría.

Espero sea éste el primero de muchos triunfos a su lado.

*“Si quieres no ser jamás vencido, no tienes sino escoger combates
en los que de ti dependa exclusivamente el salir victorioso”.*

Epícteto

MAURICIO PEDROZA TORRES

AGRADECIMIENTOS

A MSc. Jabid Quiroga, director del proyecto, por toda la confianza depositada en nosotros, además de su colaboración y excelente disposición en el planteamiento de nuevas alternativas en cada fase de este proyecto.

A la Ingeniera Silvia Oviedo, codirectora del proyecto, por todas sus oportunas correcciones.

A la Universidad Industrial de Santander, foco de todas las grandes experiencias que aportaron a nuestra formación como futuros profesionales.

A la Escuela de Ingeniería Mecánica, que por medio de los más capacitados formadores, inculcaron en nosotros el amor por nuestra carrera, además de una inagotable sed de conocimiento.

CONTENIDO

	pág
INTRODUCCIÓN	18
1. OBJETIVOS DEL TRABAJO DE GRADO	20
1.1 OBJETIVO GENERAL	20
1.2 OBJETIVOS ESPECIFICOS	20
2. MARCO TEORICO	21
2.1 DINAMICA DE ESTRUCTURAS	21
2.2 MONITOREO DE SALUD ESTRUCTURAL	23
2.3 REDES DE MAPAS AUTO-ORGANIZADOS (SOM)	24
2.4 REDES BAYESIANAS	27
3. ESTRUCTURAS ANALIZADAS	31
4. IMPLEMENTACIÓN Y EXPERIMENTACIÓN	33
4.1 CONFIABILIDAD DE LOS VALORES OBTENIDOS POR OPENSEES®	34
4.2 METODOLOGIA SOM	37
4.3 METODOLOGIA BAYESIANA	52
4.4 METODOLOGIA BAYSOM	65
5. CONCLUSIONES	76
6. RECOMENDACIONES	78
BIBLIOGRAFIA	79
ANEXOS	82

LISTA DE TABLAS

	pág
Tabla 1. Resultados de frecuencias y periodos OpenSees[®]	35
Tabla 2. Estadísticas de rendimiento (red SOM)	51
Tabla 3. Estadísticas de rendimiento (red Bayesiana)	64

LISTA DE FIGURAS

	pág
Figura 1. Modos de vibración	22
Figura 2. Grafica de probabilidad condicional	27
Figura 3. Tablas de probabilidad condicional	28
Figura 4. Red Bayesiana	28
Figura 5. Tablas de probabilidad condicional (CPT)	29
Figura 6. Viga simplemente apoyada conectividad de sus elementos	31
Figura 7. Armadura simplemente apoyada conectividad de sus elementos	32
Figura 8. Modelo viga usando SAP2000[®]	34
Figura 9. Resultados de frecuencias y periodos SAP2000[®]	35
Figura 10. Modelo armadura usando SAP2000[®]	36
Figura 11. Editor de variables de MATLAB[®], eigenvectores y eigenvalores	37
Figura 12. Matriz de información modal y vector de entrada	38
Figura 13. Área de la sección transversal y momento de inercia de la viga	39
Figura 14. Datos de entrada al programa de OpenSees[®]	40
Figura 15. Calidad del entrenamiento (red SOM)	42
Figura 16. Herramientas de análisis gráficos (red SOM)	43
Figura 17. Modos de vibración (efecto espejo)	45
Figura 18. Diagrama de funcionamiento (Metodología SOM)	47
Figura 19. Simbología y código de colores (red SOM)	48
Figura 20. Resultados gráficos de las predicciones SOM	48

Figura 21. Grafica de barras (resultados SOM)	49
Figura 22. Estructuración y reordenamiento de información modal	53
Figura 23. Metodología de obtención de valores característicos	54
Figura 24. Ejemplos de valores característicos e intervalos	55
Figura 25. Selección y definición de intervalos	56
Figura 26. Ejemplo de tabla de probabilidad condicional	57
Figura 27. Esquema de red Bayesiana	58
Figura 28. Diagrama de funcionamiento (Metodología Bayesiana)	60
Figura 29. Simbología y código de colores (red Bayesiana)	61
Figura 30. Resultados gráficos de las predicciones Bayesianas	61
Figura 31. Datos estadísticamente representativos y no representativos	62
Figura 32. Datos estadísticos (representativos y no representativos)	63
Figura 33. Diagrama de funcionamiento (Metodología BAYSOM)	67
Figura 34. Simbología y código de colores (BAYSOM, rangos), viga	68
Figura 35. Resultados de las predicciones BAYSOM (rangos), viga	68
Figura 36. Grafica de barras (resultados BAYSOM, rangos), viga	69
Figura 37. Simbología y código de colores (BAYSOM, valores), viga	70
Figura 38. Resultados de las predicciones BAYSOM (valores), viga	70
Figura 39. Grafica de barras (resultados BAYSOM, valores), viga	71
Figura 40. Simbología y código de colores (BAYSOM, rangos), armadura	72
Figura 41. Resultados de las predicciones BAYSOM (rangos), armadura	72
Figura 42. Grafica de barras (resultados BAYSOM, rangos), armadura	73
Figura 43. Simbología y código de colores (BAYSOM, valores), armadura	74

Figura 44. Resultados de las predicciones BAYSOM (valores), armadura 74

Figura 45. Grafica de barras (resultados BAYSOM, valores), armadura 75

LISTA DE ANEXOS

	pág
Anexo A. ANÁLISIS FÍSICO DE LA FORMA DEL MODO	83
Anexo B. ARCHIVOS OPENSEES®	86
Anexo C. ARCHIVOS MATLAB®	96

RESUMEN

TÍTULO: ANÁLISIS COMPARATIVO ENTRE REDES SOM Y REDES BAYESIANAS APLICADO A LA DETECCIÓN DE FALLAS EN ESTRUCTURAS*

AUTORES: EFRAÍN GUILLERMO MARIOTTE PARRA
MAURICIO PEDROZA TORRES**

PALABRAS CLAVE:

Redes de mapas auto-organizados (SOM), Redes Bayesianas, Caracterización modal, Salud estructural, Detección de daños, OpenSees, MATLAB.

DESCRIPCIÓN:

Los desafíos que deben ser afrontados por el campo de la detección de fallas pueden llegar a ser de capital consideración, más aun si hablamos de “predicción de fallas”. En busca de mantenerse a la vanguardia, y haciendo uso de las herramientas conceptuales y tecnológicas de que la ingeniería actual dispone, se llevo a cabo un estudio basado en dos enfoques teóricos distintos, redes de mapas auto-organizados (*Self-Organizing Map, SOM*) y redes bayesianas. Evaluando su desempeño en el campo de la salud estructural.

El volumen de información asociado a las metodologías de detección de fallas, solo puede ser eficientemente manipulado a través de herramientas computacionales, por medio de la implementación de OpenSees® en el campo de la obtención de información modal, y MATLAB® en el área del cálculo y la programación, se busco elevar el grado de comprensión de cualquier estructura analizada, dando lugar a diversos escenarios de prueba, modificación de parámetros y abstracción de modelos matemáticos, sin necesidad de la construcción de un sistema físico.

Las conclusiones fruto de esta investigación tienen como objetivo clarificar los pros y contras que conlleva la implementación de uno u otro enfoque (redes SOM vs redes bayesianas), y finalmente como éstas metodologías pueden trabajar de manera conjunta para el eficiente cumplimiento de los objetivos planteados en el área de la detección y cuantificación del daño en estructuras, utilizando como escenario de pruebas el análisis llevado a cabo en una viga simplemente apoyada y una armadura de 13 elementos. Aparte del objetivo académico-investigativo, otro de los principales objetivos de este proyecto, es dar los primeros pasos hacia el desarrollo de nuevas herramientas digitales capaces de reducir los gastos en equipos, sensores y mediciones experimentales, contribuyendo a las labores de mantenimiento, velando siempre por la salud estructural y fomentando el uso de las nuevas tecnologías.

* Proyecto de Grado

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingeniería Mecánica. Directores: Jabid Quiroga, Silvia Oviedo.

ABSTRACT

TITLE: COMPARATIVE ANALYSIS BETWEEN SOM NETWORKS AND BAYESIAN NETWORKS APPLIED TO STRUCTURAL FAILURE DETECTION*

AUTHORS: EFRAÍN GUILLERMO MARIOTTE PARRA
MAURICIO PEDROZA TORRES**

KEYWORDS:

Self-Organizing Map (SOM), Bayesian network, Modal characterization, Structural health, Damage detection, OpenSees, MATLAB.

DESCRIPTION:

The challenges that must be faced by the field of fault detection can be up to capital consideration, even more if we spoke about "failure prediction". Seeking to stay ahead and using the conceptual and technological tools that current engineering provides, we carried out a study based on two different theoretical approaches, *Self-Organizing Map (SOM)* and Bayesian networks. Evaluating their performance in the field of structural health.

The volume of information associated with fault detection methods only can be efficiently manipulated through computational tools, using OpenSees® in the field of modal information retrieval, and MATLAB® in the calculation and programming area, we seek to raise the level of understanding of any analyzed structure, giving place to different test scenarios, parameters modification and abstraction of mathematical models, without building a physical system.

the results of this research seek to show the pros and cons that the implementation of either approach entails (SOM networks vs Bayesian networks), and finally how these methodologies can work together for the efficient fulfillment of the objective raised in the area of detection and quantification of structural damage, using as test scenario the carried out analysis in a simply supported beam and an armor of 13 elements. Apart from the academic-research objective, another one of the main objectives of this project is to give the first steps towards the development of new digital tools able to reduce the expenses in equipment, sensors and experimental measurements, contributing to the maintenance works, always guarding by the structural health and promoting the use of the new technologies.

* Undergraduate Degree Work

** Physical-Mechanical Engineering Department, School of Mechanical Engineering. Directors: Jabid Quiroga, Silvia Oviedo.

INTRODUCCIÓN

La actual era tecnológica, capaz de brindarnos los más asombrosos frutos de su evolución, trae consigo nuevos y más sofisticados retos, dada la enorme complejidad de estructuras y mecanismos, es cada vez más común la conjunción de varias disciplinas ingenieriles, que por medio de una estrecha relación, trabajan por la funcionalidad de dichos sistemas, en lo que respecta al campo de la ingeniería mecánica, la importancia del mantenimiento y la detección de fallas se vuelve evidente cuando entra en consideración el inmenso riesgo que conlleva el desconocimiento de las condiciones físicas de cualquier estructura.

No basta con tener plena noción de todos los medios por los cuales una estructura puede fallar (fatiga, sobrecarga, degradación química, etc.), es igual o más importante aun conocer y contar con los mejores métodos de detección de estos fenómenos. Se tiene a disposición la tecnología capaz de manejar tanto la gran cantidad de datos como las complejas interrelaciones entre los distintos componentes que toman parte en la predicción de fallas estructurales, siendo clara la necesidad de implementar la tecnología de la informática con el objetivo de gestionar un modelo virtual capaz de detectar los efectos que un fenómeno aleatorio desencadenaría en la naturaleza de una estructura.

Los desafíos que deben ser afrontados por el campo de la detección de fallas pueden llegar a ser de capital consideración, más aun si hablamos de “predicción de fallas”, campo con un componente científico-tecnológico tan profundo, que industrias y en general economías subdesarrolladas, han desistido en su implementación, conservando una filosofía correctiva propia de mediados del siglo pasado. Por lo que se hace necesaria una herramienta tecnológica capaz de suplir

estas necesidades, y además compensar las falencias de métodos actuales implementados en la detección de daños (rayos-X, ultrasonido, inspección acústica, etc.) que requieren el conocimiento a priori de la localización de la posible falla además de un grado de accesibilidad a dicha ubicación.

El sistema económico mundial amenaza la permanencia en el sector productivo de cualquier empresa reacia a la implementación de nuevas tecnologías, no solo en el campo de la automatización, organización y producción, sino también en el campo del mantenimiento. En busca de mantenerse a la vanguardia, y haciendo uso de las herramientas conceptuales y tecnológicas de que la ingeniería actual dispone, se llevo a cabo un estudio basado en dos enfoques teóricos distintos, redes de mapas auto-organizados (*Self-Organizing Map, SOM*) y redes bayesianas, que por medio de un importante componente computacional, buscará mostrar los pros y contras de cada perspectiva, esperando vislumbrar el camino más eficaz hacia la obtención de la metodología capaz de suplir toda la gama de inconvenientes y necesidades previamente planteadas en el proceso de detección de fallas.

No es esta la primera vez que técnicas metaheurísticas son empleadas en el campo de la detección de fallas, incluso son varios los avances que la Universidad Industrial de Santander ha aportado a esta rama del conocimiento, siendo estudiadas previamente metodologías como: Redes neuronales, algoritmos genéticos, optimización por enjambre de partículas (*Particle Swarm Optimization, PSO*), entre otras. La novedad que trae consigo esta investigación, no solo es la misión de desarrollar un nuevo enfoque dentro de la inmensa teoría del mantenimiento, sino también de llevar a cabo un estudio comparativo (redes SOM vs redes bayesianas), tan importante en el campo teórico, como útil en el fortalecimiento de las relaciones academia-industria.

1. OBJETIVOS DEL TRABAJO DE GRADO

1.1 OBJETIVO GENERAL

- Contribuir con la misión de la Universidad Industrial de Santander en lo que respecta a la generación de conocimiento y solución de problemáticas, específicamente en el campo de la detección y predicción de fallas en estructuras y el aprendizaje de máquinas, mediante un estudio comparativo entre dos diferentes modelos de procesamiento de datos, redes SOM y redes bayesianas, impulsando la línea de investigación del grupo DICBOT en el área de inspección y mantenimiento predictivo.

1.2 OBJETIVOS ESPECÍFICOS

- Implementar un sistema de detección de daños (localización y cuantificación) en una viga y en una armadura, mediante parámetros dinámicos de la estructura (frecuencias y formas de vibración) usando redes SOM y redes bayesianas.
- Recopilar información sobre las arquitecturas, características y desempeño, de las redes SOM y bayesianas, comparando sus cualidades en lo que respecta a: velocidad de respuesta, porcentaje de aciertos en la detección y costo computacional.
- Llevar a cabo una clara documentación de los resultados obtenidos en el estudio, con el ánimo de facilitar la futura profundización en este campo del conocimiento, en un documento tipo publicación científica con formato de la IEEE.

2. MARCO TEÓRICO

2.1 DINÁMICA DE ESTRUCTURAS

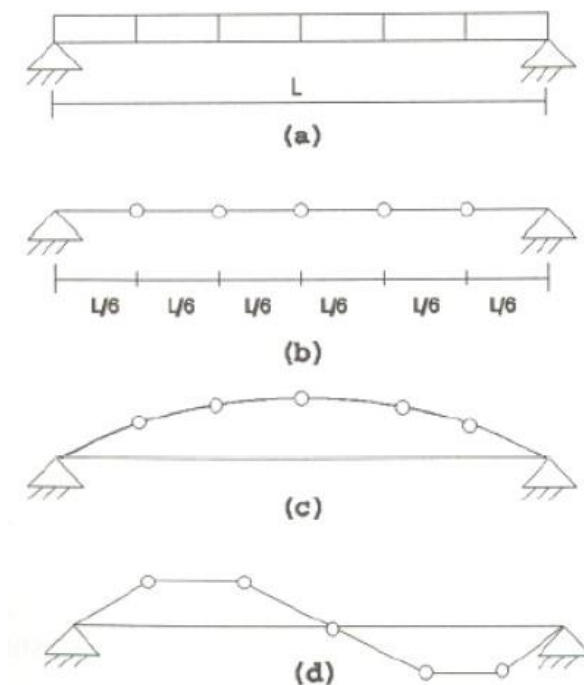
La teoría de dinámica de estructuras sometidas a cargas dinámicas, precisa que una estructura no amortiguada con múltiples grados de libertad posee un movimiento armónico simple sin cambiar la forma de deflexión. A medida que se incrementan los grados de libertad estudiados en la estructura, se incrementa el número de formas de vibración. Cuando una estructura se desplaza de una posición de equilibrio estable, la estructura tiende a volver a esta posición, debido a la acción de las fuerzas que tienden a restablecer la situación de equilibrio. Sin embargo, en el instante en que la estructura vuelve a su posición de equilibrio cuenta con un impulso que lo lleva más allá de dicha posición, presentándose una oscilación alrededor del punto de equilibrio las cuales se denominan como vibraciones mecánicas.

La dinámica estructural generalmente se dedica a las vibraciones de los cuerpos flexibles, en donde las deformaciones relativas entre los diferentes puntos que conforman la estructura son importantes; Cada forma de deflexión o vibración es llamada como los modos naturales de vibración y dependen de las características de amortiguamiento, rigidez y masa de la estructura, el proceso necesario para la determinación de las características dinámicas inherentes de un sistema en forma de frecuencias naturales, factores de amortiguamiento y formas de modos, se denomina "*análisis modal*".

La dinámica de la estructura es físicamente descompuesta por frecuencia y posición. Esto se hace claramente evidente por la solución analítica de las ecuaciones diferenciales parciales de un sistema continuo como es el caso de una

viga. El análisis modal se basa en que las respuestas de vibración de un sistema dinámico lineal que no varía con el tiempo, se puede expresar como la combinación lineal de movimientos simples harmónicos, los cuales son llamados modos naturales de la vibración, este concepto es semejante a la combinación de Fourier de las ondas de senos y cosenos para representar una complicada forma de onda. Los modos naturales de vibración son inherentes para un sistema dinámico y completamente determinados por las propiedades físicas (masa, dureza, amortiguamiento) y por la distribución espacial.

Figura 1. Modos de vibración



Fuente: René Vera Crespo (2006)

En la imagen anterior los índices (a) y (b) muestran una viga simplemente apoyada modelada con cinco puntos de masa. Los dos primeros modos de esa viga, se presentan en los índices (c) y (d).

2.2 MONITOREO DE SALUD ESTRUCTURAL

El monitoreo de salud estructural (*structural health Monitoring*, SHM) es una tecnología que combina la captura de datos de una estructura con el análisis de la misma. Los datos se consiguen desde una red sensorial no destructiva, desde la cual se obtienen indicadores que permiten detectar anomalías (daños o degradación) en la estructura. Esta información puede ser periódica o en tiempo real y continuo, y estar asociada a cambios diversos como deterioro, corrosión, fatiga, reacciones químicas, humedad, cambios en las variables del ambiente, así como a las propiedades físicas relativas a la carga, esfuerzos, desplazamientos, deformaciones, aceleraciones, vibraciones, dislocaciones entre otros.

Las técnicas aplicadas a la detección de daños son el resultado de múltiples esfuerzos de los departamentos de ingeniería de las diferentes empresas en su continuo afán de obtener una reducción en los niveles de vibración y ruido, aumentar en lo posible la expectativa de vida operacional de los diferentes sistemas así como incrementar la eficiencia y seguridad de los mismos.

Cualquier cambio en el modelo espacial (defectos), produce cambios observables en la respuesta dinámica del sistema, este es el principio en que se basan los algoritmos para la detección de daños. En términos matemáticos para dar significado al concepto de daño es necesario recurrir a la comparación entre dos diferentes estados del sistema, uno de ellos (el inicial) será en el que se asume la estructura sin daño, y el otro, aquel en que se someterá dicha estructura a múltiples escenarios, con todo tipo de alteraciones en las condiciones estructurales de dicho sistema.

2.3 REDES DE MAPAS AUTO-ORGANIZADOS (SOM)

Se observa que en muchas de las regiones del córtex de los animales superiores aparecen zonas donde las neuronas detectoras de rasgos (o características) se distribuyen topológicamente ordenadas. Por ejemplo, en el área somato sensorial, las neuronas que reciben señales de sensores que se encuentra próximos en la piel se sitúan también próximas en el córtex, de manera que se reproduce (aunque distorsionadamente) el mapa de la superficie de la piel en una zona de la corteza cerebral; lo mismo sucede, por ejemplo, en el córtex motor. Por lo que respecta al sentido del oído, existe en el cerebro un área que representa mapas tonotípicos, donde los detectores de determinados rasgos relacionados con el tono de un sonido se encuentran ordenados en dos dimensiones. En resumen, la representación de la información en la corteza cerebral aparece con frecuencia organizada especialmente, circunstancia que el modelo neurona de mapas de rasgos auto-organizados o SOM trata de reproducir.

Los SOM o mapas de Kohonen fueron desarrollados a lo largo de la década de los ochenta por el físico finlandés Teuvo Kohonen, como una continuación natural de la línea de desarrollo de las redes competitivas iniciada por von der Malsburg. Aparte de su interés como una sencilla modelización de redes neuronales naturales, los SOM poseen un gran potencial de aplicabilidad práctica. De entre las clases de problemas del mundo real en los que han demostrado su eficacia caben citar: clasificación de patrones, cuantificación vectorial, reducción de dimensiones, extracción de rasgos, monitorización de procesos, análisis exploratorio, visualización y minería de datos. Por ejemplo, los SOM han sido empleados en reconocimiento del habla, control robot, optimización combinatoria, clasificación de proteínas y de restos arqueológicos, monitorización de circuitos integrados y procesos industriales, reconocimiento de patrones financieros y minería de datos en grandes contenedores de información en internet.

El mapa puede describirse como una matriz de procesadores elementales (i,j) ordenadas en dos dimensiones , que almacenan un vector de pesos sinápticos o vector de referencia $w_{ij}(t)$, con $\{w_{ij}(t); w_{ij} \in \mathfrak{R}^m \ 1 \leq i \leq nx, \ 1 \leq j \leq ny\}$.

En “fase de ejecución” (Operación normal de la red), los pesos permanecen fijos. En primer lugar, cada neurona (i,j) calcula la similitud entre los vectores de entrada x , $\{x_k \parallel 1 \leq k \leq m\}$ y su propio vector de pesos sinápticos w_{ij} , según una cierta medida de distancia o criterio de similitud establecido. A continuación, se declara vencedora la neurona $g = (g_1, g_2)$, cuyo vector de pesos W_g es más similar al de entradas. De esta manera, cada neurona actúa como un detector de rasgos específicos, y la neurona ganadora nos indica el tipo de rasgos o patrón detectado en el vector de entradas.

$$d(w_g, x) = \min_{ij} \{d(w_{ij}, x)\} \quad (2.1)$$

En “fase de aprendizaje” cada neurona del mapa sintoniza con diferentes rangos del espacio de entrada. El proceso es el siguiente. Tras la presentación y procesamiento de un vector de entradas $x(t)$, la neurona vencedora modifica sus pesos de manera que se parezcan un poco más a $x(t)$. De hecho, ante el mismo patrón de entrada, dicha neurona responderá en el futuro todavía con más intensidad. El proceso se repite para numerosos patrones de entrada, de forma que al final de los diferentes vectores de referencia sintonizan con vectores con dominios específicos de las variables de entrada (dominio de Voronoi), y tienden a representar la función densidad de probabilidad (o función de distribución) del espacio sensorial, si dicho espacio está dividido en grupos, cada neurona se especializara en uno de ellos, y la operación esencial de la red se podrá interpretar entonces como un análisis clúster.

Lo descrito hasta el momento responde a un esquema competitivo clásico de relativa sencillez. En el que cada neurona actúa en solitario. Sin embargo, el modelo de mapa de Kohonen aporta una importante novedad, pues incorpora a este esquema relaciones entre las neuronas próximas en el mapa. Para ello introduce una “*función de vecindad*”, que define un entorno alrededor de la neurona ganadora actual (vecindad); su efecto es que durante el aprendizaje se actualizan tanto los pesos de la vencedora como los de las neuronas pertenecientes a su entorno, de esta manera, en el modelo de SOM se logra que neuronas próximas sintonicen con patrones similares, quedando de esta manera reflejada sobre el mapa una cierta imagen del orden topológico presente en el espacio de entrada (ordenación de los detectores de rasgos).

En esencia, por medio del proceso descrito las SOM realizan la proyección no lineal de un espacio multidimensional de entrada \mathfrak{R}^m sobre un espacio discreto de salida, representado por la capa de neuronas. El mapa representa una imagen del espacio sensorial, pero de menor número de dimensiones, reflejando con mayor fidelidad aquellas dimensiones del espacio de entrada de mayor varianza (que suelen coincidir con los rasgos más importantes de las entradas). La proyección se realiza de manera óptima, en el sentido de que la topología del espacio de entrada se preserva en lo posible sobre la superficie de la red (entradas “ x ” similares se corresponden con neurona próximas). Así, la distribución de las neuronas sobre el mapa resulta ser un reflejo de la función densidad de probabilidad “ $p(x)$ ”: regiones en el mayor espacio sensorial cuyos representantes “ x ” aparecen con más frecuencia ($p(x)$ mayor) serán proyectadas sobre un número mayor de neuronas en el mapa.

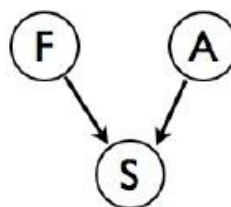
La función vecindad representa matemática de una forma sencilla el efecto global de las interacciones laterales existente entre las neuronas en el cerebro, pues en

vez de considerar en detalle que una neurona trata de activar a sus vecinas y de inhibir a las alejadas (como sucede en el córtex), modelamos esta situación mediante una sencilla función que define el tamaño de la vecindad en torno a la vencedora, dentro de la cual todas las neuronas son premiadas actualizando sus pesos, y fuera de ella son castigadas al no actualizar sus pesos al hacerlo en sentido contrario. La utilización de la función vecindad en el modelo de mapas auto-organizados aporta respecto del modelo competitivo sencillo dos ventajas adicionalmente: el ritmo efectivo de convergencia se mejora y el sistema es más robusto frente a variaciones en los valores iniciales de los pesos.

2.4 REDES BAYESIANAS

Una red bayesiana (red de Bayes) es un gráfico acíclico dirigido, donde los nodos corresponden a variables aleatorias y las flechas corresponden a la influencia directa de una variable sobre otra. Cada nodo está asociado con una tabla de probabilidad condicional (CPT), que determina la probabilidad de que una variable correspondiente adquiera un valor particular, dados los valores de sus padres. Por ejemplo, la siguiente red empleada en la representación de un diagnóstico médico:

Figura 2. Grafica de probabilidad condicional



Fuente: Olga Russakovsky (2009)

Intuitivamente, este codifica la información que los problemas nasales dependen directamente de si tiene una gripe (Flu) o alergia (Allergy). El CPT podría ser:

Figura 3. Tablas de probabilidad condicional

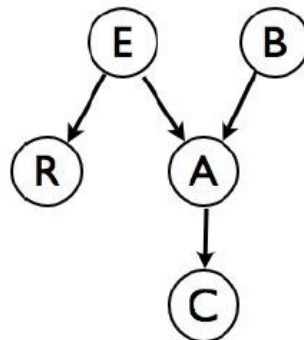
$$P(F): \begin{array}{|c|c|} \hline f & \bar{f} \\ \hline 0.3 & 0.7 \\ \hline \end{array} \quad P(A): \begin{array}{|c|c|} \hline a & \bar{a} \\ \hline 0.1 & 0.9 \\ \hline \end{array}$$

$$P(S | F, A): \begin{array}{|c|c|c|c|} \hline & & s & \bar{s} \\ \hline f & a & 0.9 & 0.1 \\ \hline f & \bar{a} & 0.6 & 0.4 \\ \hline \bar{f} & a & 0.2 & 0.8 \\ \hline \bar{f} & \bar{a} & 0.01 & 0.99 \\ \hline \end{array}$$

Fuente: Olga Russakovsky (2009)

Por ejemplo, esto indica que la probabilidad de tener alergias es del 10% y la probabilidad de tener problemas nasales si tiene alergias, pero no gripe es de 20%. Ahora veamos un ejemplo más complicado, supongamos que hay una alarma instalada en su casa, y la alarma (A) puede ser desencadenada por un terremoto (E) o un ladrón (B). Si se activa la alarma, puede hacer que su vecino llame (C). Por último, si hay un terremoto, puede ser reportado en la radio (R). En total, tenemos cinco variables aleatorias binarias: A, E, B, C y R. Vamos a representar este dominio con la siguiente red Bayesiana:

Figura 4. Red Bayesiana



Fuente: Olga Russakovsky (2009)

La variable de robo puede ser verdadera o falsa, pero no es "causada" por ninguna otra variable de la red. Lo mismo ocurre con el terremoto. La alarma depende de si hay o no un terremoto, y si es o no un robo. La llamada del vecino depende de la alarma, y la radio depende del terremoto. Los padres de un nodo, de forma intuitiva, son las únicas cosas que influyen directamente en ese nodo.

Se creará una tabla con las siguientes distribuciones de probabilidad:

- $P(B)$ - la probabilidad a priori de un robo.
- $P(E)$ - la probabilidad a priori de un terremoto.
- $P(A|B, E)$ - la probabilidad de que la alarma se active bajo cualquiera de estas circunstancias: (b^1, e^1) , (b^1, e^0) , (b^0, e^1) , (b^0, e^0) .
- $P(C|A)$ - la probabilidad de que el vecino llame para cada valor de A .
- $P(R|E)$ - la probabilidad de que la radio informe de un terremoto para cada valor de E .

Figura 5. Tablas de probabilidad condicional (CPT)

$P(B):$	b^0	b^1
	0.99	0.01

$P(E):$	e^0	e^1
	0.995	0.005

$P(A B, E):$		a^0	a^1	
	b^0	e^0	0.999	0.001
	b^0	e^1	0.7	0.3
	b^1	e^0	0.2	0.8
	b^1	e^1	0.05	0.95

$P(R E):$		r^0	r^1
	e^0	0.99999	0.00001
	e^1	0.65	0.35

$P(C A):$	c^0	c^1	
	a^0	0.95	0.05
	a^1	0.3	0.7

Fuente: Olga Russakovsky (2009)

Ahora, supongamos que estamos en el estado $(b^1, e^0, a^1, c^1, r^0)$. ¿Cuál es la probabilidad de llegar a este estado exacto? Se define de la siguiente manera:

$$\begin{aligned}
 P(b^1, e^0, a^1, c^1, r^0) &= P(b^1)P(e^0)P(a^1 | b^1, e^0)P(c^1 | a^1)P(r^0 | e^0) \\
 P(b^1, e^0, a^1, c^1, r^0) &= 0.1 \times 0.995 \times 0.8 \times 0.7 \times 0.99999 \\
 P(b^1, e^0, a^1, c^1, r^0) &= 0.05572.
 \end{aligned}$$

Acabamos de multiplicar las entradas correspondientes de todas nuestras tablas de probabilidad condicional, pudiendo observar como en toda red bayesiana es característico que cada nodo X_i asociado con un CPT $P(X_i | \text{Padres}(X_i))$, especifique una distribución de X_i para cada combinación de valores entre los padres de X_i .

Un red bayesiana representa una distribución de probabilidad conjunta entre sus variables X_1, \dots, X_n a través de la regla de la cadena para las redes de Bayes:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{Padres}(X_i)) \quad (2.2)$$

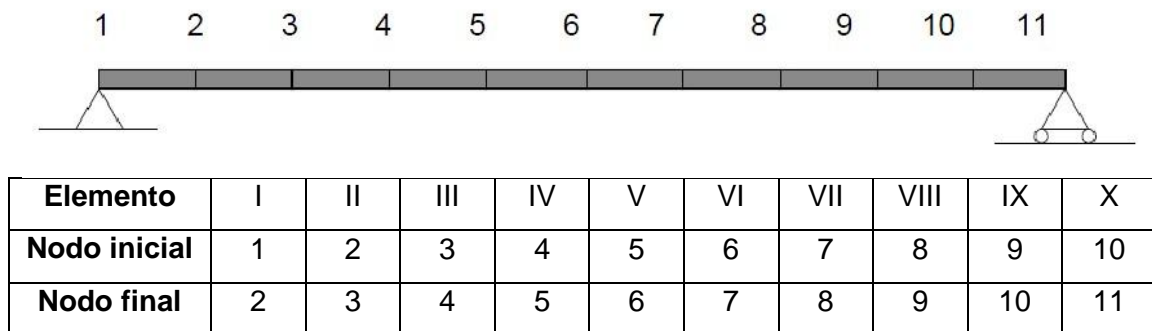
Las redes bayesianas nos brindan una forma compacta de especificar la distribución conjunta de un conglomerado de variables. En nuestra red de alarma, tenemos cinco variables, cada una capaz de tomar uno de dos valores. Por lo tanto, de manera explícita requiere que especifiquemos 32 entradas en una tabla de probabilidad condicional para esta distribución conjunta. La única restricción es la suma total debe ser igual a 1, por lo que necesitamos especificar 31 parámetros. Por otro lado, al representar el dominio como un red de Bayes, sólo es necesario especificar 10 parámetros: 1 para $P(B)$, 1 para $P(E)$, 4 para $P(A | B, E)$, 2 para $P(C | A)$, y 2 de $P(R | E)$.

3. ESTRUCTURAS ANALIZADAS

Las metodologías de detección de daño han sido estudiadas y aplicadas a dos tipos de sistemas estructurales planos: viga y armadura. Las estructuras tomadas como ejemplos son mostradas en las Figuras 6 y 7.

Viga: Muchas estructuras y/o elementos de máquinas están compuestos o se pueden modelar como una viga, elemento que a pesar de su simplicidad, es en la mayoría de los casos un componente esencial, que se debe analizar por separado para una inspección estructural minuciosa, siendo nuestro caso de estudio una viga simplemente apoyada con las siguientes características: Longitud total $L=6$ m, discretizada en 10 elementos de longitud 0.6 m, además posee una sección transversal con área $A= 0.12$ m², momento de inercia $I_z=1.6 \times 10^{-3}$ m⁴, material: concreto 3000 Psi, módulo de elasticidad $E= 2.153 \times 10^{10}$ Pa, y masa por unidad de volumen= 2402.769 kg/m³.

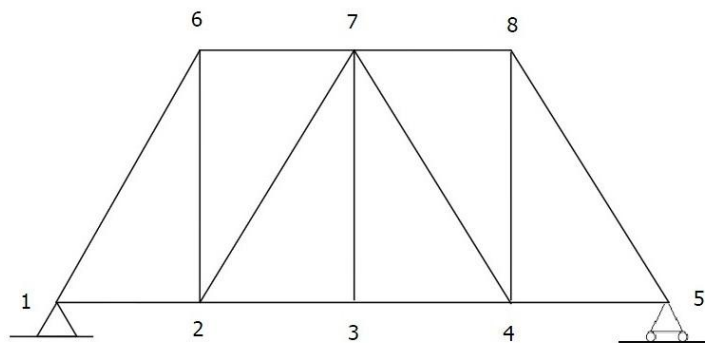
Figura 6. Viga simplemente apoyada conectividad de sus elementos



Fuente: Diego Andrés Ramírez (2010)

Armadura: Modelando una cercha, se puede comprobar el método de detección de daños en una estructura más compleja. La cercha es uno de los principales tipos de estructuras empleadas en ingeniería, especialmente en el diseño de puentes y edificios. Una armadura consta de barras rectas unidas mediante juntas o nodos. Los elementos de una cercha se unen sólo en los extremos por medio de pasadores sin fricción para formar una armazón rígida. Cada cercha se diseña para que soporte las cargas que actúan en su plano y sean aplicadas en las uniones y no en los mismos elementos; en consecuencia, una armadura puede considerarse como una estructura bidimensional, donde sus elementos están sometidos solo a fuerzas axiales directas (tracción o compresión). La cercha analizada posee las siguientes características: Armadura de 13 elementos, simplemente apoyada, con una altura de 8 ft (2.4384 m) y una longitud total de la base de 24 ft (7.3152 m), los elementos horizontales están uniformemente espaciados, es decir cada uno con una longitud de 6 ft (1.8288 m), elementos con sección transversal de área $A= 4.0 \times 10^{-4} \text{ m}^2$ y material: acero A-36 con modulo de elasticidad $E= 2.0 \times 10^{11} \text{ Pa}$.

Figura 7. Armadura simplemente apoyada conectividad de sus elementos



Elemento	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII
Nodo inicial	1	2	3	4	1	2	2	3	4	4	5	6	7
Nodo final	2	3	4	5	6	6	7	7	7	8	8	7	8

Fuente: Diego Andrés Ramírez (2010)

4. IMPLEMENTACIÓN Y EXPERIMENTACIÓN

Las respuestas dinámicas de las estructuras, frecuencias y formas modales, se determinan usando el software de libre distribución OpenSees[®]. Las condiciones, escenarios y consideraciones planteados para el desarrollo de la presente investigación se detallan a continuación:

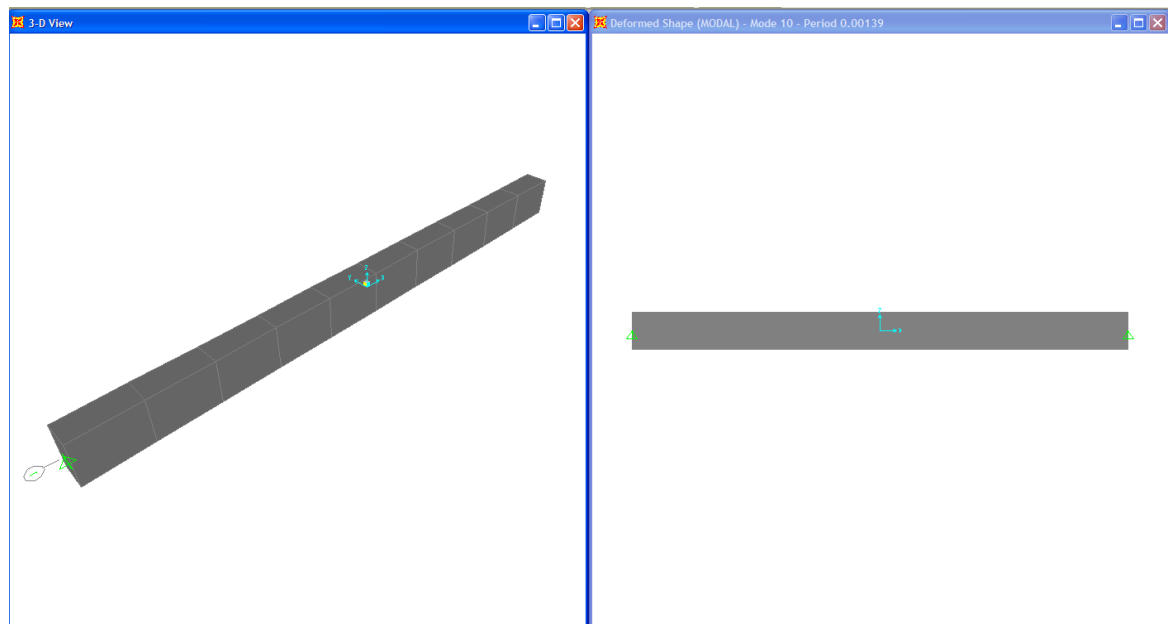
- Se considerarán reducciones entre 0 - 30% del módulo de elasticidad, debido a que variaciones más altas en el valor de dicho modulo desencadenarían en deformaciones evidentes de los elementos, por lo cual dicha situación se ubica por fuera del margen preventivo-predictivo de esta investigación.
- Se asume que la respuesta de la estructura se encuentra en un régimen lineal.
- La estructura se considera sin amortiguamiento (en estructuras civiles éste es despreciable).

Para el estudio de la condición en falla, el daño a la estructura es emulado disminuyendo el valor del módulo de elasticidad a uno o varios elementos de la estructura a través de un algoritmo de MATLAB[®]. Esta nueva versión de la estructura se introduce a OpenSees[®], el cual determina las características dinámicas. Estas características serán utilizadas, entonces, para realizar el procedimiento inverso, es decir, a partir de ellas será determinada la rigidez (módulos de elasticidad) de cada elemento usando dos enfoques diferentes (Redes SOM y Redes Bayesianas).

4.1 CONFIABILIDAD DE LOS VALORES OBTENIDOS POR OPENSEES®

La ventaja operacional en el uso de OpenSees® radica en su compatibilidad y oportunidad de fácil concatenación con la interfaz de programación de MATLAB®, pero aunque este software desarrollado por la Universidad de California, cuenta con gran prestigio y larga estancia en el entorno académico, es conveniente verificar la similitud de sus resultados con respecto a uno de los software más robustos y difundidos en el área del análisis estructural SAP2000®, debido a que este estudio ya fue llevado a cabo por el compañero DIEGO ANDRÉS RAMÍREZ RAMÍREZ integrante del grupo DICBOT en su proyecto de grado titulado: DETECCIÓN DE DAÑOS EN ESTRUCTURAS UTILIZANDO LA TÉCNICA DE OPTIMIZACIÓN PSO, a continuación se presentaran los resultados obtenidos:

Figura 8. Modelo viga usando SAP2000®



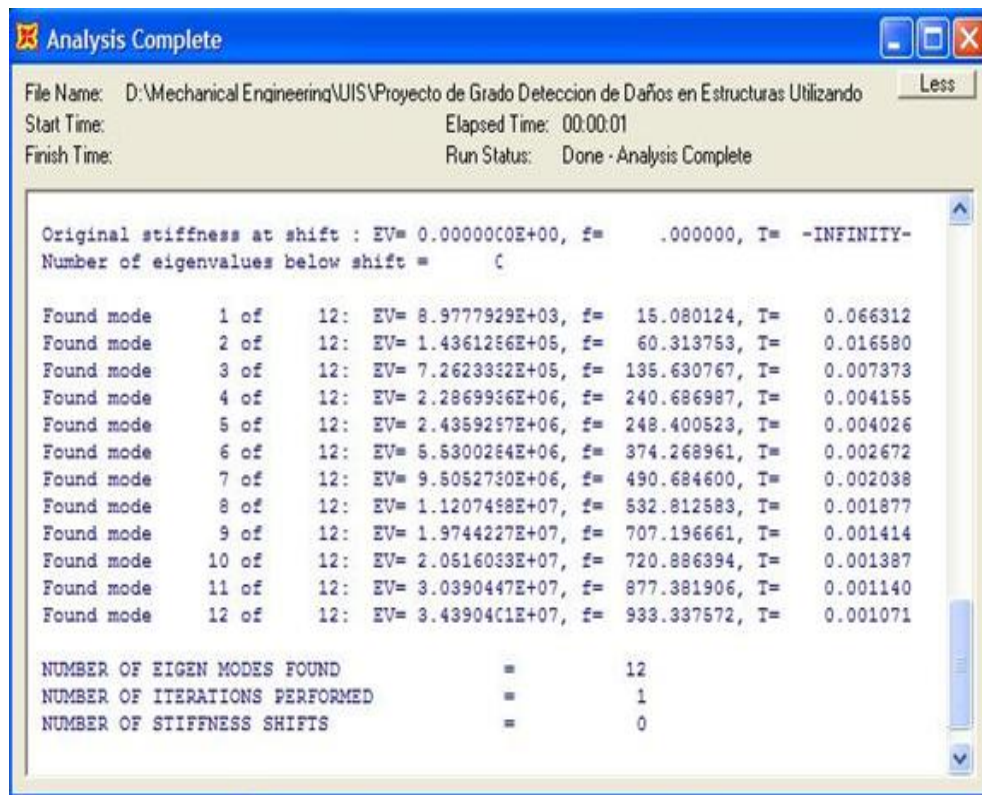
Fuente: Diego Andrés Ramírez (2010)

Tabla 1. Resultados de frecuencias y periodos OpenSees®

Elemento	Eigenvalores	Frecuencias	Periodos
1	8.98E+03	15.0816789	0.066306
2	1.44E+05	60.3199754	0.016578
3	7.26E+05	135.644743	0.007372
4	2.29E+06	240.711789	0.004154
5	2.44E+06	248.426133	0.004025
6	5.53E+06	374.307524	0.002672
7	9.51E+06	490.735188	0.002038
8	1.12E+07	532.86753	0.001877
9	1.97E+07	707.269597	0.001414
10	2.05E+07	720.960649	0.001387

Fuente: Diego Andrés Ramírez (2010)

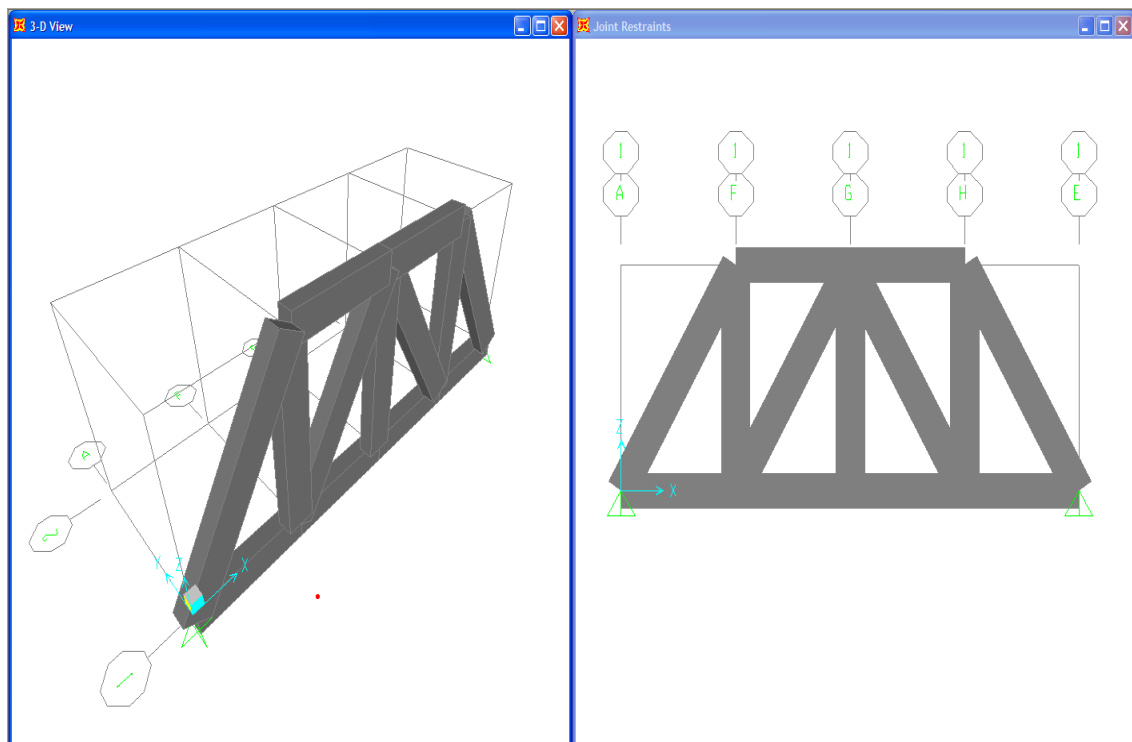
Figura 9. Resultados de frecuencias y periodos SAP2000®



Fuente: Diego Andrés Ramírez (2010)

Los resultados de frecuencias y periodos obtenidos con OpenSees[®] (Tabla 1.) son muy cercanos a los valores hallados con el análisis de SAP 2000[®] (Figura 9.) por lo que a partir de este momento se confiara en la validez de estos datos, siendo implementados como herramientas en el proceso de detección de fallas.

Figura 10. Modelo armadura usando SAP2000[®]



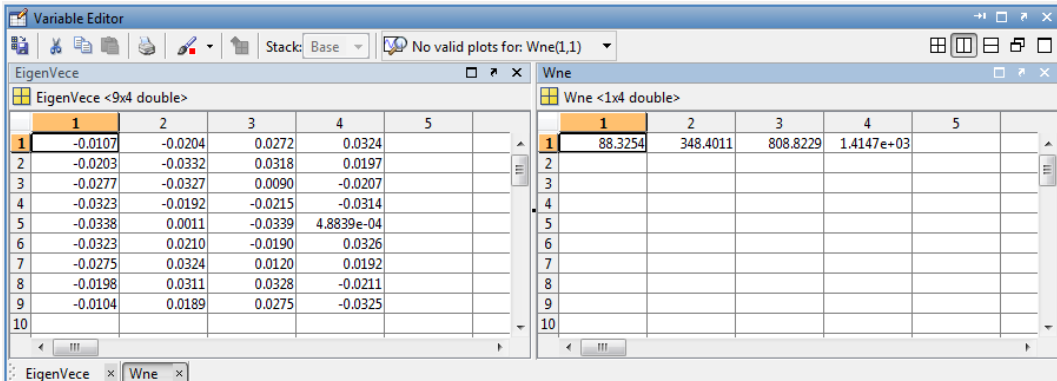
Fuente: Diego Andrés Ramírez (2010)

En el caso de la armadura el cálculo de valores propios en OpenSees[®], es muy similar al de la viga, aun teniendo en cuenta ciertas diferencias, debidas especialmente a la configuración de la estructura, por lo que el proceso de obtención de información modal con SAP2000[®] y sus características de similitud con OpenSees[®] no cuentan con diferencias apreciables, como pudo corroborarse anteriormente.

4.2 METODOLOGÍA SOM

La red SOM de forma intuitiva es en realidad un tipo de algoritmo capaz de la clasificación de escenarios, por medio de un mapa que agrupa patrones similares entre vectores de entrada, asignándole una locación específica dentro de dicho mapa en un espacio de salida, por lo que la primer interrogante a resolver consistía en como modelar los datos de entrada en forma de vectores, siendo que cada escenario de daño estaba definido por los valores y vectores propios de los cuatro primeros modos de vibración, información proveniente de OpenSees[®] luego de introducidos los datos referentes al montaje del sistema, tanto en el caso de la viga como en el caso de la armadura, se opto por desarrollar un algoritmo en MATLAB[®], capaz de reproducir escenarios de daño de forma aleatoria, requiriendo como datos de entrada únicamente el modulo de elasticidad de los elementos de la estructura y el número de elementos dañados, y brindando como dato de salida un vector organizando, cuyo primer término corresponde a la frecuencia proveniente del primer modo de vibración, seguido de cada uno de los valores del eigenvector asociados a dicha frecuencia, de manera consecutiva se anexaron a este mismo vector los valores de frecuencia y eigenvector correspondientes a cada uno de los tres modos de vibración restantes, en el mismo orden anteriormente descrito.

Figura 11. Editor de variables de MATLAB[®], eigenvectores y eigenvalores



The screenshot shows the MATLAB Variable Editor with two variables: EigenVece and Wne. EigenVece is a 9x4 double matrix, and Wne is a 1x4 double vector.

EigenVece <9x4 double>				
	1	2	3	4
1	-0.0107	-0.0204	0.0272	0.0324
2	-0.0203	-0.0332	0.0318	0.0197
3	-0.0277	-0.0327	0.0090	-0.0207
4	-0.0323	-0.0192	-0.0215	-0.0314
5	-0.0338	0.0011	-0.0339	4.8839e-04
6	-0.0323	0.0210	-0.0190	0.0326
7	-0.0275	0.0324	0.0120	0.0192
8	-0.0198	0.0311	0.0328	-0.0211
9	-0.0104	0.0189	0.0275	-0.0325
10				

Wne <1x4 double>				
	1	2	3	4
1	88.3254	348.4011	808.8229	1.4147e+03
2				
3				
4				
5				
6				
7				
8				
9				
10				

Fuente: Los autores

Figura 12. Matriz de información modal y vector de entrada

Wne	Modos de Vibración								
87,1304	0,0105	0,0200	0,0276	0,0324	0,0339	0,0323	0,0275	0,0200	0,0105
347,5150	-0,0196	-0,0322	-0,0325	-0,0200	0,0001	0,0201	0,0324	0,0324	0,0199
790,5530	0,0265	0,0323	0,0111	-0,0197	-0,0341	-0,0204	0,0103	0,0327	0,0276
1405,2665	0,0317	0,0212	-0,0193	-0,0328	-0,0008	0,0323	0,0203	-0,0201	-0,0319

Vector de entrada									
87,1304	0,0105	0,0200	...	0,0105	347,5150	-0,0196	-0,0322	-0,0325	...

Fuente: Los autores

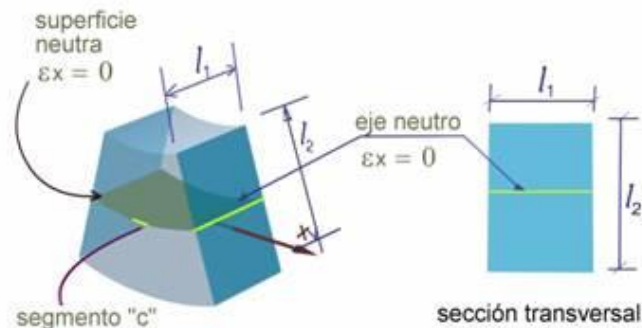
Debido a las restricciones estructurales especificadas en el modelo insertado en OpenSees[®] para la viga el vector de entrada se caracterizó inicialmente por contener la información modal de los cuatro primeros modos de vibración de la estructura, donde cada uno de estos modos contaba con un valor de frecuencia natural y 9 datos sobre la posición de los nodos en el eje vertical de la estructura, indicando la forma de vibración de dicha viga, teniendo en cuenta que se especificaron restricciones en el movimiento tanto del nodo inicial como final, obteniendo un vector de dimensiones finales [1x40]. De manera similar en el caso de la armadura, se trabajó con la información característica de los primeros cuatro modos de vibración, pero contando en esta ocasión con 12 datos de posición de los nodos luego de restringir el movimiento de 4 de ellos debido a las características estructurales del montaje, obteniendo un vector de dimensiones [1x52].

Los Programas *DataViga.m* y *DataArm.m* están encargados de generar los múltiples escenarios de daños que se toman como set de entrenamiento de la red SOM, consiste en un algoritmo que selecciona de forma aleatoria los elementos a dañar y luego genera los porcentajes de daño para cada uno de los elemento

seleccionados, considerando reducciones entre 0 - 30% del módulo de elasticidad es decir valores entre el rango (0.7-1), debido a que variaciones más altas en el valor de dicho modulo desencadenarían en deformaciones evidentes de los elementos, ubicándonos en un escenario de daño no contemplado en el enfoque preventivo de esta investigación.

Luego de la generación de escenarios de daño aleatorio por medio del proceso anteriormente descrito, el siguiente paso consistía en imprimir dichos cambios en la naturaleza de la estructura dentro de una matriz donde también se cuenta con el área de la sección transversal de cada uno de los elementos discretizados además del momento de inercia de dicha área con respecto a su eje neutro, debido a que estamos tomando en consideración solo los movimientos de la viga en el eje vertical de la misma (Ver Figura 13).

Figura 13. Área de la sección transversal y momento de inercia de la viga



Fuente: Russel C. Hibbeler (2006)

La matriz contiene en su primera columna el valor del modulo de rigidez de cada uno de los 10 elementos con un espectro de valores en la mayoría de casos menores que el inicial, lo cual evidencia una reducción en su rigidez, o en otras

palabras un daño apreciable en dicho elemento, en la siguiente columna el área de la sección transversal, seguida de aquella columna donde se almacena el valor de inercia de cada uno de los elemento, como se puede apreciar en la Figura 14.

Figura 14. Datos de entrada al programa de OpenSees®

```

Archivo Edición Formato Ver Ayuda
# 4.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit Volume=
# Section "ElasticDefault":      secTag      E      A      IZ
section Elastic 1 1.822959e+010 1.200000e-001 1.600000e-003
section Elastic 2 1.989965e+010 1.200000e-001 1.600000e-003
section Elastic 3 1.810840e+010 1.200000e-001 1.600000e-003
section Elastic 4 1.959270e+010 1.200000e-001 1.600000e-003
section Elastic 5 1.645796e+010 1.200000e-001 1.600000e-003
section Elastic 6 1.947410e+010 1.200000e-001 1.600000e-003
section Elastic 7 1.905130e+010 1.200000e-001 1.600000e-003
section Elastic 8 2.019343e+010 1.200000e-001 1.600000e-003
section Elastic 9 1.764520e+010 1.200000e-001 1.600000e-003
section Elastic 10 1.716021e+010 1.200000e-001 1.600000e-003

```

Fuente: Los autores

Luego de tener la matriz de datos cargada en un archivo .tcl, el siguiente paso consiste en la ejecución del archivo maestro (*cuervo.tcl* en el caso de la viga y *estructura.tcl* en el caso de la armadura) de OpenSees®, el cual se encarga de convocar todos aquellos archivos que contengan información relacionada con las especificaciones del problema (*1.tcl,2.tcl,3.tcl* aplicados a la viga y *Masa.tcl, Elementos.tcl, DPconstraint.tcl, Nodos.tcl* aplicados a la armadura), la matriz de datos de entrada exportada por MATLAB® (*4.tcl* en la viga y *Materiales.tcl* en la armadura), el método de procesamiento de datos (*5.tcl,6.tcl* para la viga y *Analisis3op.tcl* para la armadura), y finalmente la forma de guardar y exportar dichos datos (*7.tcl,8.tcl* referente a la viga y *Analisis2recorder.tcl* referente a la armadura), para luego capturar los vectores de datos en archivos del formato .out generados por el software OpenSees®, siendo estos los vectores encargados de almacenar los valores de frecuencias naturales y eigenvectores asociados a cada

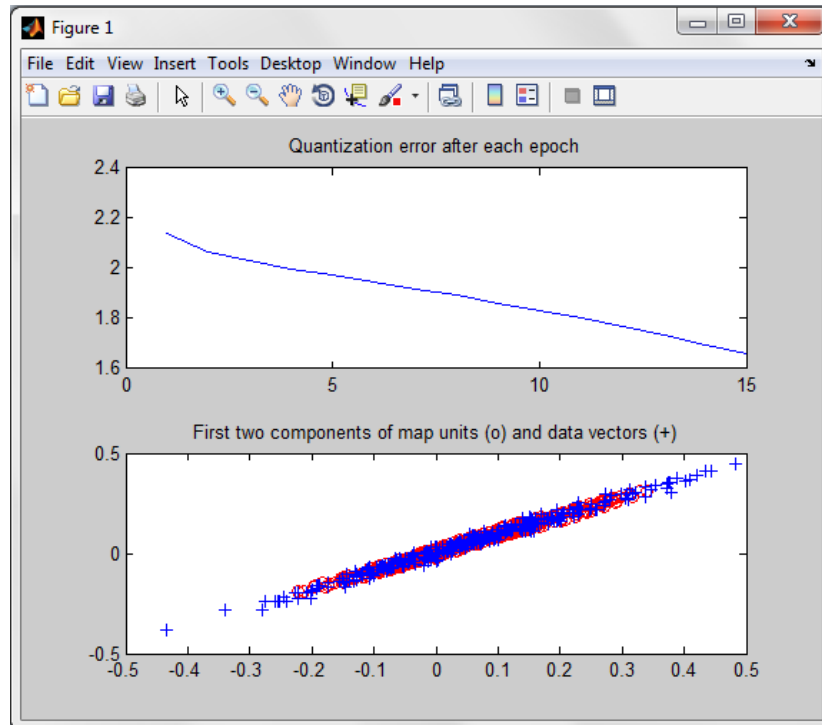
modo de vibración calculado por el programa, convirtiéndose estos luego de su previo proceso de ordenamiento en los vectores de entrada de la red SOM.

Este proceso se lleva a cabo a lo largo de múltiples repeticiones en busca de generar una base de datos capaz de ser tomada como set de entrenamiento de la red SOM, gracias a esto pudo evidenciarse como el proceso de intercomunicación entre MATLAB® y OpenSees® provoca la mayor carga computacional de todas las fases del proyecto, convirtiéndose en foco de interés debido al objetivo referente a la reducción en el costo computacional, previamente planteado en esta investigación.

Dado que en el caso de la red SOM se puede reducir el problema de clasificación al número de características observadas (número de columnas del vector de entrada) y número de grupos a clasificar (número de patrones en el mapa con una topología diferente en el espacio), ya definidas las características de cada vector se dispuso a obtener el primer entrenamiento de la red, dando paso al análisis de resultados, estableciendo en cada situación que el número de grupos a clasificar fuera el máximo posible, y en el mejor de los casos que cada neurona fuera un tipo diferente de patrón, sabiendo que cada neurona entrenada tendrá como posición el vector más próximo a ella. Es importante resaltar que en esta investigación se empleo la SOM Toolbox Versión 2.0beta, programada en MATLAB® y desarrollada por Esa Alhoniemi, Johan Himberg, Juha Parhankangas y Juha Vesanto, capaz de brindar opciones de visualización de resultados como puede apreciarse en la Figura 15, donde se observa la disminución del error de cuantización a través de las épocas o iteraciones del entrenamiento de la red SOM, en este caso las épocas se definen por la estabilización de dicha curva, además pueden apreciarse los primeros dos componentes de los vectores del set

de entrenamiento (cruses azules) y las unidades del mapa o neuronas (círculos rojos), tomando la forma del grupo de datos.

Figura 15. Calidad del entrenamiento (red SOM)



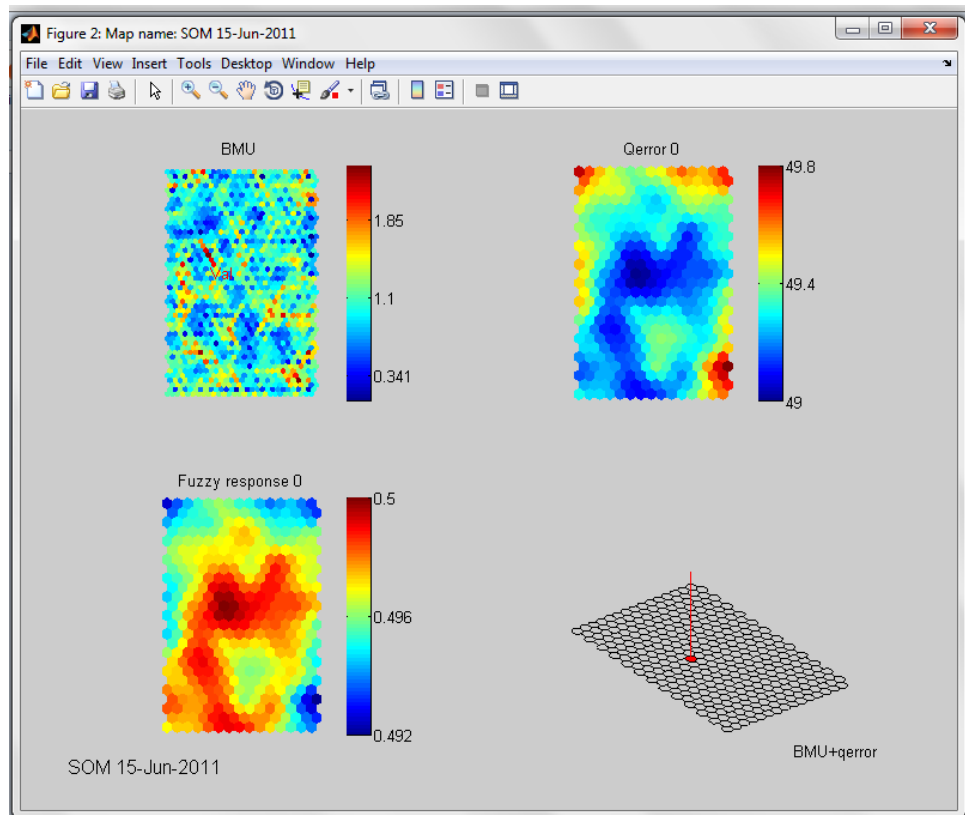
Fuente: Los autores

Usando las herramientas de análisis gráfico de la *toolbox* de redes SOM podemos verificar la calidad del entrenamiento en lo que respecta a la eficiente captura de la topología de los datos y el correcto acopio de cada neurona a un vector de entrada.

la matriz U es la Herramienta más importante de todas, debido a que indica los pesos sinápticos de la red neuronal como distancia de una neurona con respecto a

otra, mostrando la calidad del entrenamiento de la red, además se presentan dos esquemas complementarios: la respuesta *fuzzi* y el diagrama de cuantización del error encargados de indicar el porcentaje de error de cercanía (el peso de la neurona y que tan lejos se encuentra del valor más cercano del vector de entrada a comparar), por ultimo tenemos la posición de la neurona ganadora considerada más afín con respecto al vector a clasificar.

Figura 16. Herramientas de análisis gráficos (red SOM)



Fuente: Los autores

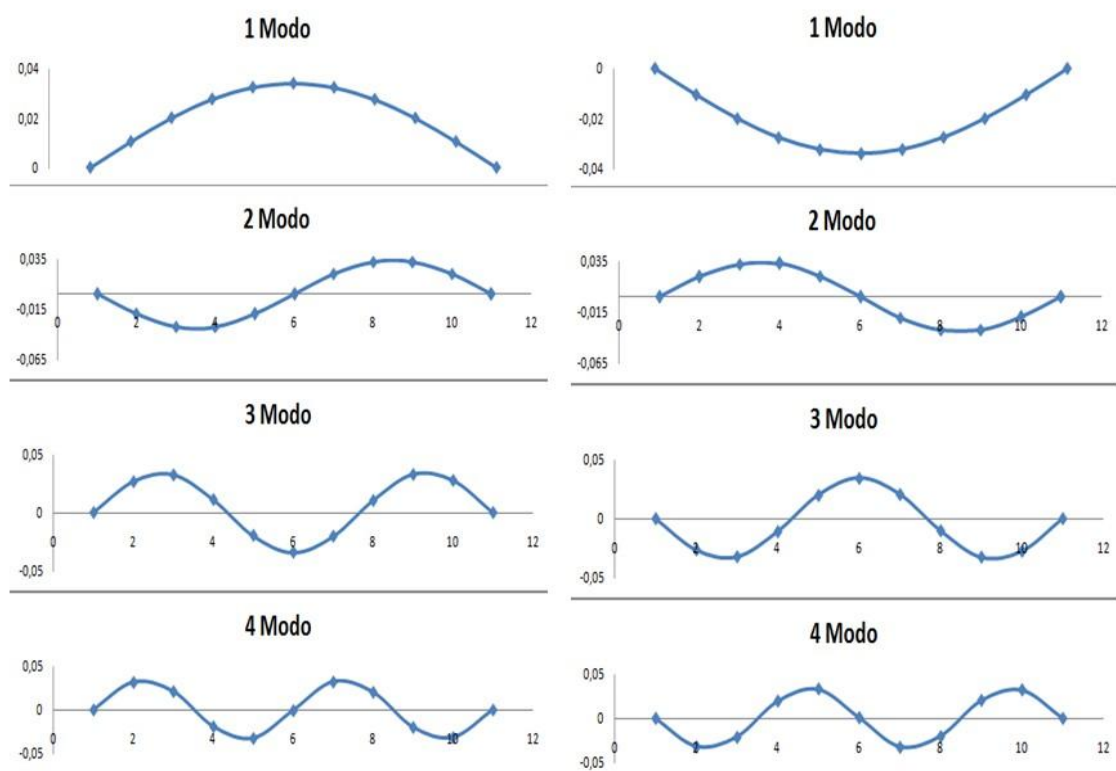
Los primeros resultado para uno, dos y tres elementos dañados y un volumen de datos de 100 escenarios como base de entrenamiento no fue demasiado contundente, se obtuvieron errores graves, en algunos casos era posible encontrar

el porcentaje de daño de alguno de los elemento con una precisión de $\pm 0 - 0.03$ pero no su ubicación correcta a lo largo de la estructura, o en el caso de dos o tres elementos las posiciones de dichos elementos dañados, se presumían en ubicaciones contiguas a las de los elementos que en realidad presentaban daño. llevándonos a la conclusión de que el vector de entrada no era lo suficientemente característico para dar una respuesta satisfactoria, o el volumen de datos no era estadísticamente lo suficientemente grande para el entrenamiento, pero en vista de que la generación y organización de los escenarios de daño y la posterior creación del set de entrenamiento de la red SOM es el punto crítico de este modelo de predicción debido al tiempo que toma el proceso computacional, se decidió primero experimentar con la reorganización del vector de entrada y la implementación de otras herramientas usadas en la caracterización de las vibraciones de una estructura dañada en comparación con una sana y a la vez reducir el tiempo de generación de dichos datos de entrada.

En busca de obtener una forma lo suficientemente característica para representar los valores de entrada, evitando así la generación de datos innecesarios, y por ende reduciendo el costo computacional, se propuso un primer planteamiento, el cual consistía en el desarrollo de una función de mejor ajuste a los puntos de los modos de vibración para encontrar las constantes de una función sinusoidal de la forma: $y = A * \sin(Bx + C) + D$ que permitiera combinar 13 datos en 4 característicos: la amplitud de la onda, la frecuencia y el desfase de la onda en el eje "y" y "x", y debido a la naturaleza de las restricciones el desfase en y es nulo. Este enfoque pretendía trabajar solo con tres variables características por modo, pero los resultados fueron negativos ya que la función no podía definir las constantes para el cuarto modo de vibración, y además de ser estadísticamente poco confiable en la determinación de las constantes para los primeros modos, no era un buen enfoque para ser aplicado en el análisis de la armadura, por lo que se decidió prescindir de esta metodología.

Un segundo enfoque para proporcionar los datos de entrada fue sumarle al vector ya obtenido otras variables que se pueden calcular de los eigenvalores y eigenvectores, además de tener en cuenta el fenómeno de la simetría en la respuesta dinámica (efecto espejo) en el caso de la viga, ya que los eigenvectores describen la misma respuesta dinámica sin importar de que nodo restringido venga la estimulación, provocando que el vector defina una situación donde la amplitud de la onda este arriba de la posición inicial y otra exactamente igual aun cuando la amplitud de la onda se encuentra por debajo de la posición inicial y debido a que las redes SOM interpretan valores positivos y negativos como escenarios con características diferentes, este era un aspecto a tener en cuenta.

Figura 17. Modos de vibración (efecto espejo)



Fuente: Los autores

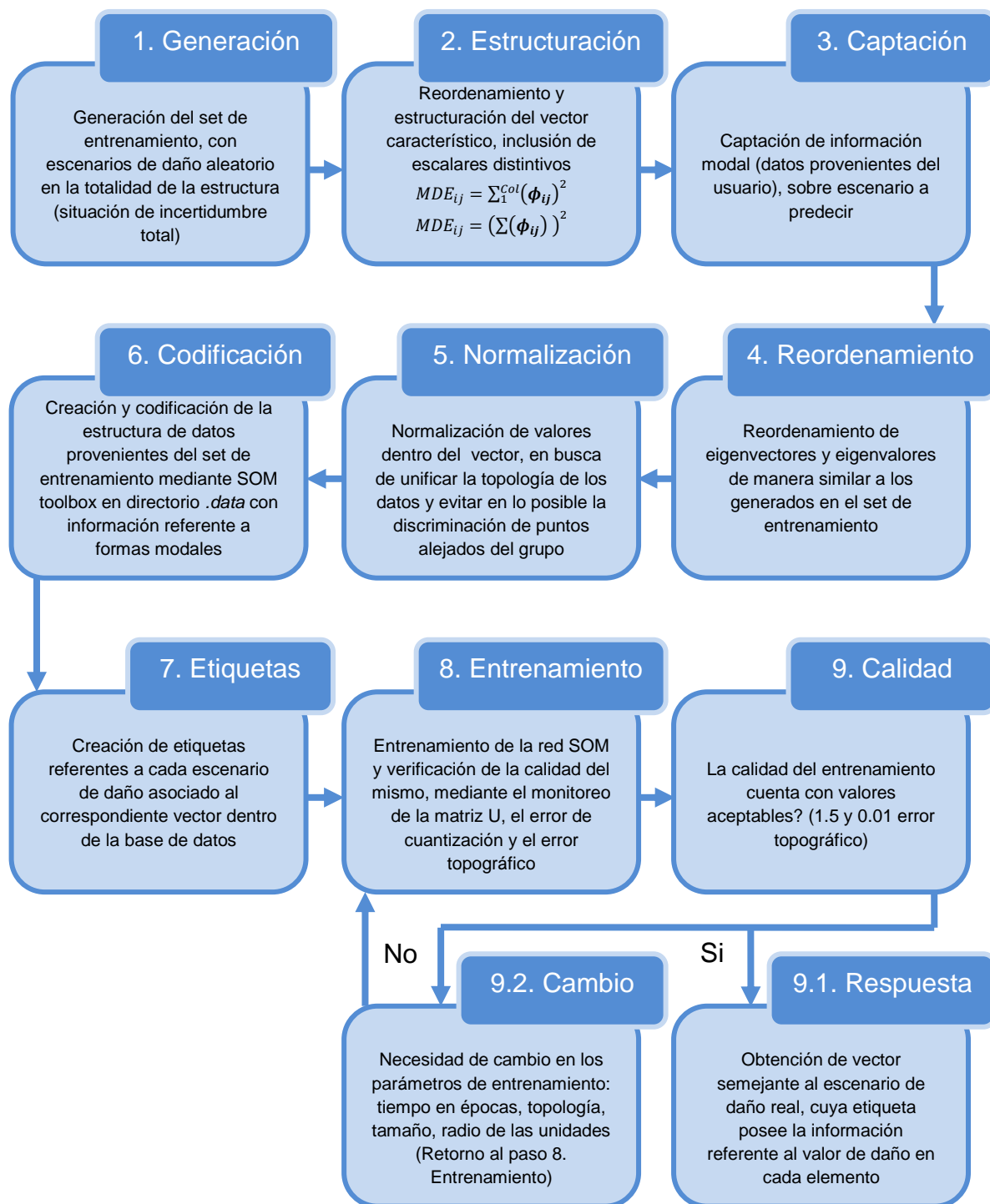
Se buscaron variables características que permitieran realizar una fácil distinción entre vectores por lo que se probaron enfoques como la mínima distancia euclidiana (MDE), y el criterio de acercamiento modal (MAC, por sus siglas en inglés).

$$MDE_{ij} = \sum_1^{col} (\phi_{ij})^2 \quad (4.1) \quad MAC_{ij} = \frac{(\phi_i^T \phi_j)^2}{(\phi_i^T \phi_i)(\phi_j^T \phi_j)} \quad (4.2)$$

Pudo comprobarse que al aplicar el criterio MAC en la distinción de modos de vibración pertenecientes al mismo escenario de daño puede ser lo suficientemente característico, lo cual no ocurre entre modos de diferentes escenarios de daño, siendo esto último precisamente el objetivo de la búsqueda, debido a las evidentes dificultades al tratar de obtener datos suficientemente característicos se llegó a la conclusión de que el porcentaje de ruido aceptado en la medición no debe ser demasiado alto, ya que esto ocasionaría una total variación de la respuesta dinámica, dificultando la correcta predicción acerca del estado original de la estructura. Por esta razón se optó por dar una aplicación diferente a las redes SOM, de forma que en el entrenamiento cada neurona buscara acercarse el máximo posible a un vector de entrenamiento, buscando heredar la etiqueta colocada a aquel vector, donde se encuentra almacenada la información del respectivo porcentaje de daño presente en cada elemento, pero en este punto la metodología se limita a obtener como resultado final una aproximación al escenario real de daño, para lo cual la veracidad de la aproximación solo será función de la capacidad del algoritmo de repetir un dato perteneciente al set de entrenamiento, encontrándose además un segundo criterio para una mayor caracterización de los vectores de entrenamiento una modificación del MDE:






$$MDE_{ij} = \left(\sum (\phi_{ij}) \right)^2 \quad (4.3)$$

Figura 18. Diagrama de funcionamiento (Metodología SOM)



Fuente: Los autores

Figura 19. Simbología y código de colores (red SOM)

x	Elemento dañado ($0,7 \leq x < 0,85$)
v	Elemento no-dañado ($0,85 < x \leq 1$)
	Predicción correcta ($\pm 0,03$ del valor real)
	Predicción aproximada ($\pm 0,05$ del valor real)
	Predicción de estado (Alta desviación del valor real, acierto en el estado del elemento)
	Falso positivo (Predicción de daño en elemento no-dañado)
	Falso negativo (Predicción de no-daño en elemento dañado)

Fuente: Los autores

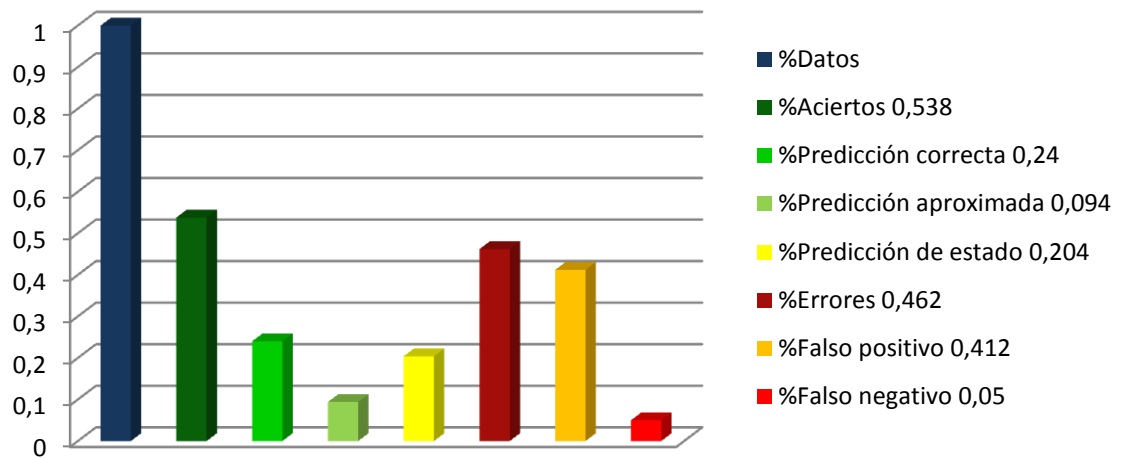
Figura 20. Resultados gráficos de las predicciones SOM

	1 Elemento					2 Elementos					3 Elementos					4 Elementos					5 Elementos				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	V	V	V	V	X	X	X	X	X	
2	X	X	X	X	X	V	V	V	V	V	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
3	X	X	X	X	X	X	X	X	X	X	V	X	V	V	X	X	X	X	X	X	X	X	X	X	
4	V	V	V	V	V	V	V	V	V	V	X	X	X	X	X	V	V	V	V	V	X	X	V	V	
5	X	X	X	X	X	V	V	V	V	V	X	X	X	X	X	X	X	X	X	V	X	V	V	V	
6	V	V	V	V	V	V	V	V	V	V	X	X	X	X	V	X	X	X	X	X	X	X	X	X	
7	V	V	V	V	V	V	V	V	V	V	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
8	V	V	V	V	V	X	X	X	X	X	V	X	V	V	V	X	X	X	X	X	V	X	X	X	
9	V	V	V	V	V	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	X	X	X	
10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	V	V	V	V	X	X	X	X	X	

	6 Elementos					7 Elementos					8 Elementos					9 Elementos					10 Elementos				
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	V	V	V	V	X	V	V	X	X	V	X	V	X	X	V	V	V	V	X	V	X	V	V	V	
2	X	X	X	X	V	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
3	X	X	X	V	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	X	X	
5	V	V	V	X	V	X	V	X	X	X	X	X	X	X	V	V	X	X	V	X	V	X	X	X	
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	V	X	X	X	X	X	X	
7	X	X	X	X	V	X	X	V	V	X	V	X	V	V	X	X	X	X	V	X	V	X	X	X	
8	X	X	X	V	X	X	X	X	V	X	V	X	X	V	X	V	X	X	X	X	V	V	V	V	
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
10	X	X	X	V	X	X	X	X	X	V	X	V	X	X	V	X	V	X	X	X	X	V	V	V	

Fuente: Los autores

Figura 21. Grafica de barras (resultados SOM)



Fuente: Los autores

Se pudo determinar que manejando un volumen cercano a los 1000 datos, podría obtenerse una respuesta estadísticamente aceptable, variando el número de elementos dañados, desde uno hasta la totalidad de los elementos pertenecientes a la estructura, renovando el número de datos para la base de entrenamiento en cada caso de estudio, contando con un tiempo de cálculo de 3 min y 45 segundos para cada escenario de daño aproximadamente.

Se pudo detectar que al emplear un set de entrenamiento en donde todos los elementos presentan daño, en la predicción de un escenario en donde el número de elementos dañados fuera menor a 6, se obtenían valores poco cercanos a los reales, al no contemplarse en dicho entrenamiento que muchos de los elementos debían considerarse en perfectas condiciones, volviéndose necesaria la implementación de un set de entrenamiento donde el número de elementos dañados fuese igual al número de daños en el escenario a predecir, enfoque descartado al no contarse con el conocimiento a priori del número de elementos

dañados en una estructura. Se obtuvieron mejores predicciones en aquellos escenarios que contemplaban daño en 7 o más elementos, debido a la mayor similitud del escenario de entrenamiento con la distribución real de daño en la estructura, luego de someter a estudio por alrededor de 6 ocasiones a cada escenario de daño, se buscaba establecer el valor de daño con mayor numero de apariciones (moda) , aunque en algunas oportunidades cada análisis realizado daba como resultado una diferente aproximación al escenario de daño, esto debido al proceso aleatorio por el cual los vectores prototipo son ubicados en la red, los ciclos de entrenamiento y la topología de los datos.

Luego de analizar, el comportamiento y alto índice de error obtenido por esta metodología, era evidente que difícilmente podría llegar a cumplirse con los estándares de confiabilidad que este estudio pretende alcanzar, pero gracias a que fueron detectados muchos de los puntos débiles de esta metodología, se obtuvieron importantes conclusiones, siendo la primera de ellas la evidente necesidad de restringir los valores que los datos de entrenamiento de la red SOM pueden tomar.

Si se espera una buena aproximación al valor de daño real, es necesario brindar a la red SOM información sobre los rangos más probables en que el valor de daño real está alojado, obteniéndose escenarios de entrenamiento de mayor similitud además de valores predictivos más confiables.

Tabla 2. Estadísticas de rendimiento (red SOM)

Viga y Armadura

Generación de datos

Tamaño del vector (Viga): [1 x 38]

Tamaño del vector (Armadura): [1 x 69]

Número de Datos	Tiempo [min: s :cs]	Tiempo [min: s :cs]
	Viga	Armadura
100	15:35	46:62
200	45:56	1:39:32
300	1:17:12	2:45:73
500	2:30:50'	4:00:47
1000	4.09:36	8:34:80
1500	6:24:11	13:01:29

Entrenamiento

Viga: Épocas de entrenamiento [7 25]

Radio: [10 3.5]

Armadura: Épocas de entrenamiento [7 50]

Radio: [10 3.5]

Número de datos		Tiempo de entrenamiento [min: s :cs]	Error de cuantización	Error topográfico	Tamaño del mapa
Viga	100	4:14	2.050	0.050	[35 3]
	200	7:12	2.260	0.010	[18 10]
	300	11:70	2.015	0.023	[18 15}
	500	21:76	1.855	0.024	[45 10]
	1000	54.64	1.645	0.016	[40 22]
	1500	1:46:32	1.608	0.028	[50 25]
Armadura	100	13:70	3.370	0.0	[35 3]
	200	10:40	3.310	0.020	[18 10]
	300	11:26	3.217	0.017	[18 15}
	500	19:44	3.030	0.016	[45 10]
	1000	57.65	2.847	0.018	[40 22]
	1500	2:31:10	2.756	0.023	[50 25]

4.3 METODOLOGÍA BAYESIANA

La aplicación de la metodología bayesiana en el campo de la detección de fallas en estructuras, cuenta con dos componentes principales, el primero de ellos es el componente computacional, aquel algoritmo capaz de dar el tratamiento matemático a datos provenientes de la red bayesiana con el fin de arrojar resultados en términos probabilísticos, en el presente trabajo esta particular tarea fue realizada gracias a la toolbox de MATLAB® especializada en la construcción y ejecución de redes bayesianas FullBNT-1.0.4 desarrollada por Kervin Murphy en la Universidad de California (Berkeley), este algoritmo de inferencia bayesiana es el encargado de obtener los valores de probabilidad que determinan la presencia o ausencia de daño en un elemento, para lo cual requieren una red bayesiana completamente estructurada, lo que nos lleva al segundo gran componente de esta metodología de detección, la identificación de los nodos, conexiones y parámetros probabilísticos necesarios para la construcción de la red bayesiana, este procedimiento fue realizado de forma totalmente experimental, ya que la búsqueda de precedentes los cuales aplicarían el enfoque bayesiano a la detección de fallas en estructuras usando información modal fue infructuosa, tratando de establecer a través de múltiples enfoques la metodología más eficiente.

Finalmente, y como resultado de un arduo proceso de experimentación se logró establecer una metodología de modelamiento para la red bayesiana buscada. El primer paso consistió en la compilación de toda información modal característica de la estructura y sus condiciones de daño en unos pocos valores numéricos capaces de permitir la identificación de dicho escenario de daño, para lo cual se empleó el concepto matemático de la mínima distancia euclidiana:

$$MDE = \sqrt{\sum_{i=1}^n (x_{ri} - x_{si})^2} \quad (4.4)$$

Se emplearon para este propósito los cuatro primeros modos de vibración, cuyos valores de frecuencia y eigenvector asociado fueron obtenidos gracias a OpenSees® y ordenados de la forma mostrada en la Figura 22, se recopiló la información de dos escenarios diferentes, un escenario sin daño como referencia, y luego un escenario de daño aleatorio, esta información se redujo aun mas al multiplicar a cada valor en el eigenvector por su respectiva frecuencia asociada.

Figura 22. Estructuración y reordenamiento de información modal

Estructura sin daño			
94,761	379,001	852,281	1512,44
0,01051	-0,02	0,02751	0,03234
0,01999	-0,0323	0,03234	0,01999
0,02751	-0,0323	0,01051	-0,02
0,03234	-0,02	-0,02	-0,0323
0,034	0	-0,034	0
0,03234	0,01999	-0,02	0,03234
0,02751	0,03234	0,01051	0,01999
0,01999	0,03234	0,03234	-0,02
0,01051	0,01999	0,02751	-0,0323

1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

A) Frecuencia X Eigenvector (Sin daño)			
0,99564	-7,5745	23,444	48,9075
1,89383	-12,256	27,5601	30,2265
2,60663	-12,256	8,95483	-30,227
3,06428	-7,5745	-17,033	-48,908
3,22198	0	-28,978	0
3,06428	7,57446	-17,033	48,9075
2,60663	12,2557	8,95483	30,2265
1,89383	12,2557	27,5601	-30,227
0,99564	7,57446	23,444	-48,908

Estructura a evaluar			
85,8535	342,631	764,316	1389,01
-0,0106	-0,0209	0,02799	0,03359
-0,02	-0,0332	0,03198	0,0185
-0,0273	-0,0323	0,00926	-0,0222
-0,0322	-0,0194	-0,0206	-0,0317
-0,034	0,00038	-0,0344	0,00221
-0,0325	0,01998	-0,0192	0,03297
-0,0276	0,03222	0,01236	0,01801
-0,0201	0,03184	0,03262	-0,0207
-0,0105	0,01923	0,02629	-0,0311

1	0,75
2	0,74
3	0,81
4	0,92
5	0,86
6	0,72
7	0,92
8	0,7
9	0,99
10	0,83

B) Frecuencia X Eigenvector (Daño)			
-0,9067	-7,1574	21,3955	46,6514
-1,7159	-11,38	24,4454	25,694
-2,3476	-11,059	7,07675	-30,836
-2,7602	-6,659	-15,767	-44,043
-2,9229	0,12962	-26,301	3,07231
-2,7864	6,84628	-14,709	45,7905
-2,372	11,0386	9,4442	25,0228
-1,722	10,9107	24,9347	-28,714
-0,9001	6,58976	20,0923	-43,223

Fuente: Los autores

Posterior a esto se dio paso a la aplicación del concepto de la mínima distancia euclidiana, por lo que fue necesario restar cada uno de los valores pertenecientes al escenario de referencia con su correspondiente en el escenario dañado, para luego elevar al cuadrado el valor de dicha diferencia, como se puede observar en la Figura 23, para finalizar se efectuaron cuatro sumatorias, una por cada columna, obteniendo un valor por cada sumatoria, se extrajo la raíz cuadrada de estos términos obteniendo los cuatro valores considerados como característicos del escenario de daño evaluado, asociando estas cantidades con el modo de vibración de donde cada uno de dichos términos provienen, por lo cual cada modo de vibración observado en la estructura aporta un valor característico, estos escalares se convertirán en los distintivos de cada escenario de daño, alejándolo completamente de otros escenarios de daño notablemente diferentes, y vinculándolo con escenarios de daño numéricamente similares.

Figura 23. Metodología de obtención de valores característicos

A - B				(A-B) ²			
1,90231	-0,4171	2,04853	2,25608	3,61878	0,17397	4,19648	5,08989
3,6097	-0,8756	3,1147	4,53253	13,03	0,7667	9,70133	20,5438
4,95425	-1,1965	1,87808	0,60971	24,5446	1,43151	3,52719	0,37175
5,82452	-0,9155	-1,2659	-4,8642	33,925	0,83812	1,6024	23,6606
6,14485	-0,1296	-2,6775	-3,0723	37,7592	0,0168	7,16917	9,43912
5,85071	0,72817	-2,3238	3,11699	34,2308	0,53024	5,40017	9,7156
4,97867	1,21712	-0,4894	5,2037	24,7872	1,48137	0,23948	27,0785
3,61584	1,34502	2,62546	-1,5123	13,0743	1,80908	6,89302	2,28697
1,89573	0,9847	3,35169	-5,6848	3,5938	0,96964	11,2338	32,3174
				188,564	8,01743	49,9631	130,504

Evidencias	$(188,56)^{1/2}$	F1	13,7318
	$(8,017)^{1/2}$	F2	2,83151
	$(49,963)^{1/2}$	F3	7,06846
	$(130,50)^{1/2}$	F4	11,4238

Fuente: Los autores

Continuando con el modelamiento de la red, y luego de conocer el proceso de obtención de valores característicos para cada escenario de daño, se procede a crear la base de datos capaz de brindar los parámetros probabilísticos requeridos por la red en su proceso de cálculo, por lo cual se debe recopilar información sobre múltiples escenarios de daño, información estadística capaz de aclarar interrogantes sobre la relación existente entre los valores característicos de una estructura y la existencia o inexistencia de daño en sus elementos, para lo cual deben ser agrupados en categorías los valores característicos más próximos numéricamente, con el ánimo de observar el espectro de escalares asociados con múltiples escenarios de daño.

Figura 24. Ejemplos de valores característicos e intervalos

Escenario 1		Escenario 2		Escenario 3		Escenario 4	
F1	13,732	F1	13,878	F1	0,7423	F1	0,6326
F2	2,8315	F2	1,9548	F2	2,6898	F2	54,59
F3	7,0685	F3	4,9109	F3	5,8635	F3	122,82
F4	11,424	F4	9,0208	F4	11,017	F4	13,203

F1	14,006	F2	3,1196	F3	5,1661	F4	9,0208
	13,938		3,1013		4,9109		9,0048
	13,878		3,0586		4,9109		8,9064
	13,878		2,9418		4,9014		8,1222
	13,832		2,931		4,7529		8,0148
	13,737		2,9091		4,6913		8,0148
	13,732		2,9047		4,6382		7,8773
	13,7		2,8975		4,6308		7,8773
	13,644		2,8844		4,3101		7,4529
	13,54		2,877		4,3101		7,4529
	13,521		2,877		3,601		7,3086
	0,8226		2,8315		3,5621		7,0457
	0,8198		2,7091		3,5207		6,9517

Fuente: Los autores

En la Figura 24 se presentaron algunos ejemplos de valores característicos en cuatro escenarios distintos de daño, con el ánimo de notar la similitud numérica entre algunos de estos valores o evidente diferencia en otros casos, también se pudo observar cómo se comporta el espectro de valores asociados a cada uno de los cuatro modos de vibración, pudiéndose establecer intervalos capaces de agrupan todos aquellos valores numéricamente no tan distantes, estableciéndose como frontera de estos intervalos aquellas pareja de valores consecutivos con una evidente diferencia numérica entre sí, teniendo como referencia la diferencia promedio obtenida entre términos consecutivo a lo largo de toda la serie de valores. Luego de aplicar este procedimiento a un volumen considerable de escenarios de daño, se podrá obtener un amplio espectro de valores característicos, que se espera contenga la mayor cantidad de escenarios posibles, para que dicha muestra de datos, pueda brindarnos importante información en el momento que se esté evaluando cualquier escenario no perteneciente a la muestra, pero que posiblemente cuenta con valores característicos similares a algunos de los escenarios de daño pertenecientes a la base de datos.

Figura 25. Selección y definición de intervalos

F17	$x-(\geq 13,8)$	F27	$x-(\geq 54,59)$	F37	$x-(\geq 122,8)$	F47	$x-(\geq 12,8)$
F16	$(\geq 13,52)-x-(<13,8)$	F26	$(\geq 2,93)-x-(<54,59)$	F36	$(\geq 6,92)-x-(<122,8)$	F46	$(\geq 12,09)-x-(<12,8)$
F15	$(\geq 0,74)-x-(<13,52)$	F25	$(\geq 2,83)-x-(<2,93)$	F35	$(\geq 6,04)-x-(<6,92)$	F45	$(\geq 9,71)-x-(<12,09)$
F14	$(\geq 0,61)-x-(<0,74)$	F24	$(\geq 2,12)-x-(<2,83)$	F34	$(\geq 4,9)-x-(<6,04)$	F44	$(\geq 8,91)-x-(<9,71)$
F13	$(\geq 0,51)-x-(<0,61)$	F23	$(\geq 1,74)-x-(<2,12)$	F33	$(\geq 4,31)-x-(<4,9)$	F43	$(\geq 7,31)-x-(<8,91)$
F12	$(\geq 0,29)-x-(<0,51)$	F22	$(\geq 1,43)-x-(<1,74)$	F32	$(\geq 3,18)-x-(<4,31)$	F42	$(\geq 6,35)-x-(<7,31)$
F11	$x-(<0,29)$	F21	$x-(<1,43)$	F31	$x-(<3,18)$	F41	$x-(<6,35)$

Fuente: Los autores

Teniendo los valores característicos agrupados por categorías, se buscara obtener cuáles son los escenarios de daño estructural predominantes en cada intervalo.

Para obtener los escenarios de daño predominantes en cada uno de los intervalos definidos se hace necesaria la implementación de tablas de probabilidad condicional, donde la información sobre el estado de cada uno de los elementos en un escenario de daño perteneciente al intervalo en cuestión alimenta dicha tabla, como lo muestra la Figura 26 , esta tabla de probabilidad representa el comportamiento estadístico de los escenarios cuyo valor característico en el primer modo de vibración es igual o se encuentra por encima del escaler 13,83, podemos ver como el escenario de daño con mayores probabilidades de contar con un valor característico que se encuentre dentro de este intervalo es aquel en el cual no existe una reducción apreciable en el modulo de elasticidad de su primer elemento (V), el segundo elemento presenta daño (X) y el elemento tercer elemento al igual que el primero se presume sin daño (V). Estas tablas de probabilidad condicional deben contemplar todos los elementos de la estructura sometidos a daño, aunque es elección del diseñador de la red con que dimensiones contara la tabla de probabilidad, ya que agregar otro valor de daño o no-daño a la tabla duplica el número de sus filas, lo cual puede dificultar el manejo de la tabla y la visualización de resultados.

Figura 26. Ejemplo de tabla de probabilidad condicional

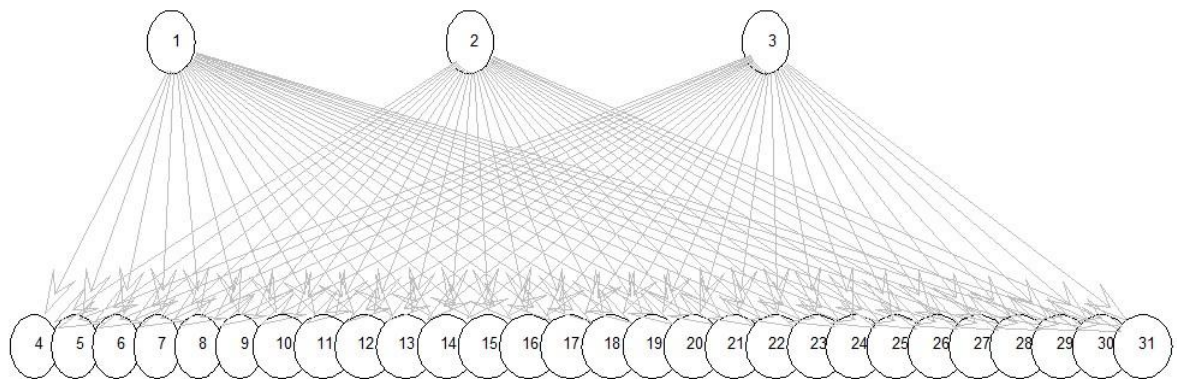
$x-(\geq 13,83)$

E1	E2	E3	SI	NO		SI%	NO%
X	X	X	0	15	15	0	1
X	X	V	2	13	15	0,13	0,87
X	V	X	0	15	15	0	1
X	V	V	4	11	15	0,27	0,73
V	X	X	0	15	15	0	1
V	X	V	6	9	15	0,4	0,6
V	V	X	2	13	15	0,13	0,87
V	V	V	1	14	15	0,07	0,93

Fuente: Los autores

Tras lograr obtener tanto los intervalos que agrupan los valores característicos de diferentes escenarios de daño, y las tablas de probabilidad asociadas a estos intervalos, no queda más que esquematizar la distribución de la red bayesiana obtenida, como se puede observar en la Figura 27 esta red está conformada por tres nodos que llamaremos los “padres”, desde los cuales se separan múltiples flechas las cuales permiten su conexión con los nodos llamados “hijos”.

Figura 27. Esquema de red Bayesiana



Fuente: Los autores

En este diagrama, los nodos “padre”, son los primeros tres elementos de la estructura analizada, los cuales están habilitados para tomar solo uno de dos estados, encendido o apagado, elemento sometido a daño o elemento en buen estado, la probabilidad de encontrarse en alguno de estos estados es el dato estadístico de interés para esta investigación, pues es esta la información que permitirá predecir el estado estructural del sistema bajo estudio, aunque nada de esto podría lograrse sin la intervención de los nodos “hijo”, los cuales comunican a sus padres la información estadística contenida en las tablas de probabilidad condicional, permitiendo la obtención de la probabilidad asociada a la presencia o ausencia de daño en los elementos de la estructura, es aquí donde se hace

evidente la agrupación según intervalos que se le dio a los valores característicos de cada escenario anteriormente, pues cada uno de estos intervalos corresponde a uno de los nodos “hijo” en la red bayesiana, dicho nodo se activara si el sistema bajo estudio cuenta con un valor característico ubicado en el intervalo representado por dicho nodo, por lo que al contar con cuatro valores característicos por escenario, en el momento de analizar el estado de un sistema en particular se deben ubicar dichos valores en alguno de los intervalos previamente establecidos e informar a la red bayesiana por medio de la activación del nodo “hijo” correspondiente, lo cual será llamado “evidencia”, disponiendo así de cuatro evidencias, una por cada dato característico del sistema.

Finalmente, después de completar el modelamiento de la estructura y conexiones de la red, y luego de establecer los parámetros estadísticos necesarios condensados en las tablas de probabilidad condicional, solo restaba someter el procedimiento de predicción de fallas a múltiples escenarios de prueba, esperando determinar su porcentaje de aciertos y desaciertos, buscando evidenciar su nivel de confiabilidad, al igual que su grado de practicidad y aplicabilidad. Debido a esto, en la etapa de experimentación se hizo necesaria la implementación de convenciones, las cuales aclaran tanto el criterio utilizado para nombrar un valor estadístico como representativo o no representativo, como el rango de valores numéricos que identifica a un elemento dañado, además del conjunto de valores causantes de errores permisibles en la predicción debido a su proximidad.

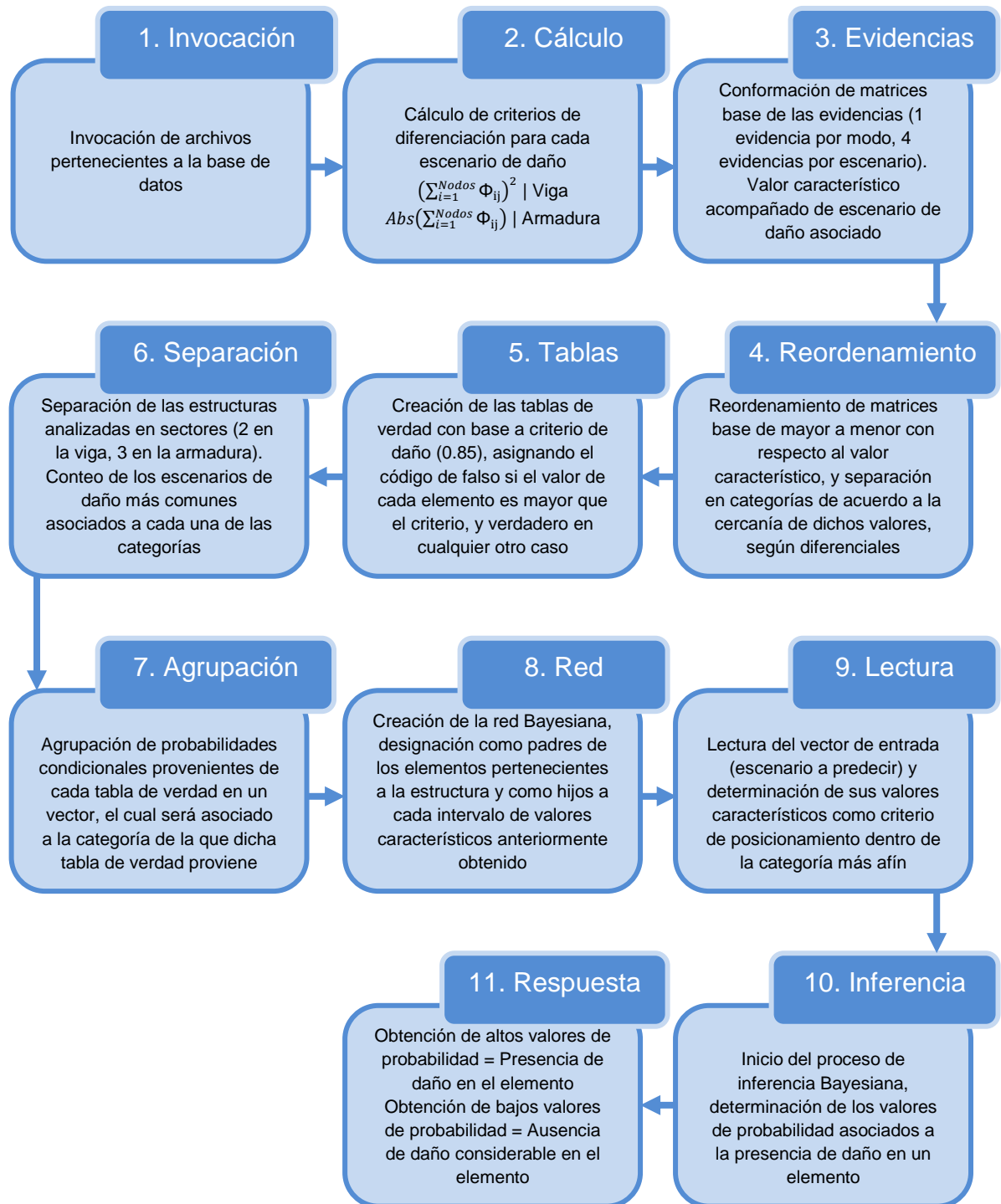
Datos estadísticamente representativos: $P(x) > 0.8$ o $P(X) < 0.2$

Datos estadísticamente no representativos: $0.2 \leq P(x) \leq 0.8$

Daño considerable: 0.7 E - 0.85 E (reducción entre un 15% y 30% en el valor del modulo de elasticidad)

Rango de tolerancia: 0.83 E - 0.88 E

Figura 28. Diagrama de funcionamiento (Metodología Bayesiana)



Fuente: Los autores

Figura 29. Simbología y código de colores (red Bayesiana)

x	Elemento dañado (Predicción en base a datos estadísticamente representativos)
v	Elemento no-dañado (Predicción en base a datos estadísticamente representativos)
	Error leve (permisible por encontrarse el valor de daño dentro del rango de tolerancia)
	Error fuerte (diagnostico incorrecto del estado del elemento en cuestión)
x̄	Elemento dañado (Predicción en base a datos estadísticamente no representativos)
v̄	Elemento no-dañado (Predicción en base a datos estadísticamente no representativos)
	Error leve (permisible por encontrarse el valor de daño dentro del rango de tolerancia)
	Error fuerte (diagnostico incorrecto del estado del elemento en cuestión)
	Acierto en la predicción

Fuente: Los autores

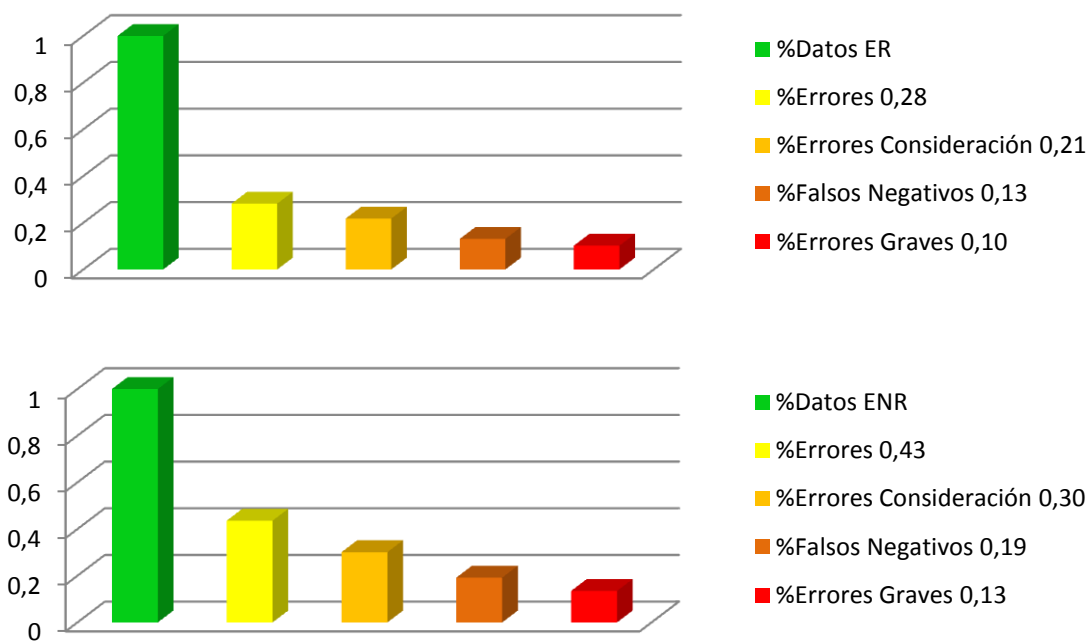
Figura 30. Resultados gráficos de las predicciones Bayesianas

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	x	x	v	v	v	v	x	v	v	v	x̄	v	x̄	v	x̄	x	x̄	v̄	x	v	x	v	v	v	x
2	v	x	x̄	x	x̄	v	x	x	v	x	x	v	x̄	v̄	x	v	v̄	v̄	x	v	v	x	v̄	v	x
3	v	x	x	v	v	x	x	v	x̄	x	x	x	v	x	x̄	v	v̄	x	x̄	v	v	x	x̄	x	x
4	v	x	x̄	v	x̄	v	x̄	v	v	x	v	v	x̄	x	v̄	v	v	x	x	v	v	v	v	v	v
5	v	x	x	x	v	v	x	x	v	v	x	x	v	x	x	v	x̄	x	x̄	v̄	v	x	x	v	x
6	v	v	v	v	x̄	v	x	x̄	v	v	v	v	x̄	x	v	v	v	x̄	v	v	v	x	x̄	v	v
7	x	v	x	x	v	v	v	x	v	x	x	v	x̄	x	v	v	x	x	v	x	x	x	v	v	x
8	v	x	x̄	x	x	v	x	x̄	x	x̄	x̄	x̄	x	v̄	x	x̄	x	x̄	x̄	v	v	x̄	v	x̄	v̄
9	v	x	x̄	v	v	v	x	v	v	x	v	v̄	x	x̄	x̄	v̄	v	x	x	v	v	v̄	v̄	v	v̄
10	x	x	x̄	x	x	x	x̄	x	x	x̄	v̄	x̄	x	x̄	x̄	x̄	x	x	x	x	x̄	v̄	v̄	x	v
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
1	v	v̄	v	v̄	v̄	v̄	v	x̄	v̄	v	v	x	v	x̄	x̄	v	v	v	x	v	v	v	x̄	x	v
2	v̄	v̄	v	x	x̄	x̄	x̄	x	x̄	v̄	v	x	v	v	x̄	x̄	v	v	v	v	x	v̄	v	x	v
3	x	v̄	x̄	x	v̄	x	x	x̄	v	x	x	x	v̄	x̄	x̄	v	x	x	x	v	v	x	v̄	x	x
4	x	v̄	x̄	x̄	x̄	x	x	v	x	v̄	x	v	v	x	v	v	v	v̄	v	v	x	v	x	v	v
5	x	v	v̄	x̄	x̄	x	x	x̄	x̄	x̄	x	x	v	x	x	v	x	x	v	x	v	x	v	x	x̄
6	v̄	x	x	v̄	v̄	v̄	x̄	x̄	v	x	v̄	x	v̄	v	x̄	x̄	x	x	v̄	v	v	x	v	x	v
7	x̄	v̄	x̄	v̄	v̄	x	v̄	v̄	v̄	v	v̄	x	v	v	v̄	v̄	x̄	v	x̄	v	x	v	v̄	x̄	v
8	x	x	x	x	x	x̄	x̄	x̄	x̄	v̄	x̄	x	v̄	v̄	x̄	x̄	x	x̄	x	v̄	v̄	v̄	v̄	v	x̄
9	x	v̄	x	v	x̄	x	x	x̄	v	x̄	x̄	x	v̄	v	x̄	v	v̄	x̄	x	v	v	x̄	v	v	v̄
10	x	x	x	x	x̄	x	x̄	x̄	x̄	x̄	x	x	x̄	x	x̄	x̄	x̄	x	x	v	x̄	x̄	x	v	x̄

Fuente: Los autores

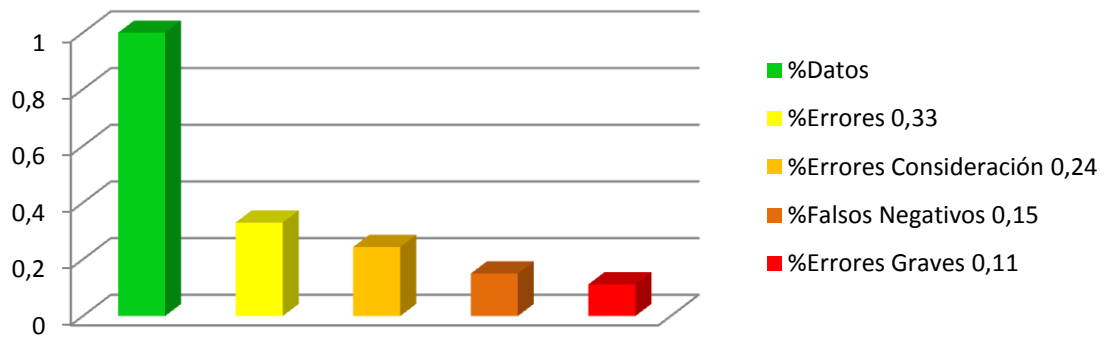
Posterior a la obtención de los datos sobre errores en la predicción, se procedió a realizar un análisis estadístico de estos resultados, buscando no solo establecer el porcentaje de errores obtenidos, sino también las características de dichos errores, ubicándolos dentro de ciertas categorías, en primer lugar se cuantifico la cantidad total de errores, a continuación el porcentaje de errores considerables, es decir errores no justificados, por no encontrarse el valor de daño del elemento dentro del rango de tolerancia, luego se estableció la cantidad de falsos negativos, aquellas predicciones erróneas que aseguraban el buen estado de un elemento aun cuando este presentaba condiciones de daño, y por último fue calculado el porcentaje de errores graves, aquellos casos de falsos negativos con altas magnitudes de daño en el elemento analizado, fuentes principales de fallas catastróficas, debido a la cancelación de un mantenimiento preventivo causada por una falsa suposición de un correcto estado estructural del elemento.

Figura 31. Datos estadísticamente representativos y no representativos



Fuente: Los autores

Figura 32. Datos estadísticos (representativos y no representativos)



Fuente: Los autores

Luego de llevar a cabo el análisis de 500 diferentes elementos presentes en 50 estructuras sometidas a diversas condiciones de daño, puede concluirse que la obtención de un 69% de altos valores de probabilidad que sugieren la presencia de daño en el elemento, o por el contrario valores demasiado bajos que contradicen esta idea, y tan solo con un 31% de datos donde cualquiera de los dos escenarios (daño y no-daño) pueden tener cabida, los índices de errores graves son evidentemente bajos para una metodología con tan poco costo computacional, pues en esta ocasión fue bastante pequeña la base de datos utilizada, al contar solo con los datos provenientes de 100 estructuras para su construcción. Pese a la obtención de interesantes resultados, en cuanto a los porcentajes y categorías de errores obtenidos, y la evidente capacidad de la metodología bayesiana de realizar predicciones acertadas acerca del nivel de daño presente en diversos elementos, por medio del análisis estadístico de la información modal brindada por la estructura, se busco el perfeccionamiento de la metodología capaz de dar solución a la problemática de la detección y cuantificación de daño, aunque eso representara la simultanea implementación de la metodología bayesiana en conjunto con las redes SOM, obteniendo como resultado el desarrollo de la metodología BAYSOM, la cual se explicara en detalle más adelante.

Tabla 3. Estadísticas de rendimiento (red Bayesiana)

Viga y Armadura

Generación de datos

Tamaño del vector (Viga): [1 x 46]

Tamaño del vector (Armadura): [1 x 61]

Número de Datos	Tiempo [min: s :cs] Viga	Tiempo [min: s :cs] Armadura
100	29:62	33:99
200	1:12:46	51:07
500	2:26:77	2:29:53
1000	4:54:01	5:06:07
2000	10:04:07	9:44:81
4000	19:39:13	17:54:34

Obtención de probabilidades

Viga: Nivel de precisión medio Diferenciales= [5 0,08 35 0,08]

Armadura: Nivel de precisión alto Diferenciales= [10 20 23 20]

Número de datos		Tiempo de procesamiento [min: s :cs]	Numero de categorías generadas
Viga	100	1:29:71	109
	200	1:54:62	148
	500	2:33:06	210
	1000	3:13:32	251
	2000	3:45:02	278
	4000	5:05:60	326
Armadura	100	1:53:62	107
	200	1:55:67	121
	500	2:33:53	137
	1000	3:13:30	153
	2000	3:37:93	160
	4000	4:40:41	170

4.4 METODOLOGIA BAYSOM

Luego de obtener los datos referentes al desempeño de cada una de las metodologías bajo estudio (redes SOM y redes Bayesianas), se pudo comprobar la notoria capacidad que tiene cada una de estas metodologías, para brindarnos información sobre el estado actual de cualquier estructura bajo estudio, evidenciando su comportamiento en cuanto a precisión en los aciertos, porcentaje de errores y presencia de falsos negativos (fuentes de fallas catastróficas) en la predicción, por otro lado, es apreciable que los porcentajes de error inherentes a la implementación de estas metodologías son considerables, por lo que extendiendo los objetivos de esta investigación y saliéndose del margen meramente comparativo, se propuso desarrollar un enfoque capaz de combinar las fortalezas de cada una de las metodologías, con el ánimo de suplir mutuamente sus falencias, esperando la obtención de resultados predictivos de mayor confiabilidad.

Luego de comprobar la evidente capacidad de las redes Bayesianas en determinar la pertenencia o no de un dato específico, dentro de un intervalo, de acuerdo a un razonamiento estadístico, se logró llevar a cabo una importante modificación en el algoritmo base de dicha predicción, dejando de lado la predicción sobre dos únicos estados posibles, con daño y sin daño, para pasar a la estimación del intervalo de valores de daño en que dicho elemento puede encontrarse, lo cual por sí mismo es un valioso indicador en el proceso de determinación del estado actual de la estructura. Gracias a la identificación de los rangos más probables en que se pueden encontrar los valores de daño de cada elemento, la metodología SOM entra en juego, pues el disponer de esta información implica limitar el espectro de valores, lo que disminuye la cantidad de aproximaciones lejanas al escenario real de daño, incrementándose las probabilidades de obtener respuestas de alta precisión.

Se dividió el bloque de programación en tres secciones, siendo la primera aquella que comprende el ingreso de los datos provenientes de la medición (frecuencias naturales y eigenvector asociado para los primeros cuatro modos de vibración), seguidamente se calculan para cada modo el criterio de diferenciación expuesto anteriormente en la metodología Bayesiana, para obtener las evidencias de daño de las múltiples posibilidades provenientes de las bases de datos disponibles (4000 escenarios de daño tanto para viga como para armadura), permitiendo la ejecución del algoritmo de inferencia bayesiana, capaz de asignar a cada elemento de la estructura bajo estudio el intervalo más probable en que puede localizarse su valor de daño real.

El segundo Bloque de programación se encarga de capturar la tabla con los rangos obtenidos por medio de la metodología Bayesiana, para luego proceder a la generación de nueva base de datos para el enfoque SOM donde el espectro de valores que pueden ser asignados al entrenamiento de la red, se encuentra limitado por los rangos anteriormente obtenidos, trabajando con un volumen de 1000 datos y un periodo de cálculo cercano a los 80 segundos

El tercer bloque de programación comprende el entrenamiento de la red SOM, donde dicha red busca la mayor afinidad con el vector de daño que se introdujo al principio, brindando información sobre la calidad del entrenamiento y la topología de los datos. Esta fase tiene como objetivo encontrar una forma satisfactoria de entrenar la red, para luego proceder a la búsqueda del vector más próximo al dato de entrada.

Figura 33. Diagrama de funcionamiento (Metodología BAYSOM)



Fuente: Los autores

Figura 34. Simbología y código de colores (BAYSOM, rangos), viga

X	X (Coeficiente del modulo de elasticidad x 100)
X Y	X (Frontera inferior del rango) Y (Frontera superior del rango)
	Predicción correcta (Valor de daño real ubicado dentro del rango)
	Predicción aproximada (Valor de daño real ubicado a $\pm 0,03$ de alguna de las fronteras)
	Error (Valor de daño real completamente fuera del rango)

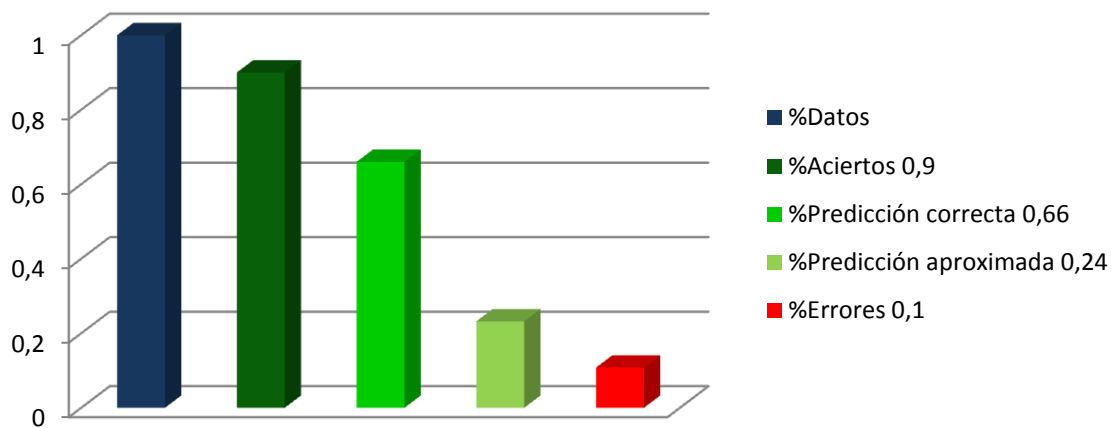
Fuente: Los autores

Figura 35. Resultados de las predicciones BAYSOM (rangos), viga

	Sin ruido															Ruido 0,005					Ruido 0,01				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	8 9	7 85	75 9	75 9	7 85	85 9	8 95	75 95	75 95	75 9	75 95	75 95	75 9	75 85	75 9	8 9	75 9	75 9	8 9	75 9	75 9	75 9	8 85	75 9	8 95
2	8 9	75 85	8 9	75 9	7 9	75 8	8 9	75 9	75 9	8 9	75 95	75 9	75 9	75 9	75 9	8 9	75 9	8 9	8 9	75 9	75 9	75 9	8 85	75 9	8 9
3	75 9	7 85	8 9	75 9	7 9	75 85	75 9	75 95	75 9	75 9	75 9	75 85	75 9	75 9	75 9	8 9	75 9	75 9	75 85	75 9	75 9	75 9	8 85	75 9	75 9
4	75 9	75 9	8 9	75 9	75 85	9 1	75 9	8 95	75 9	75 9	75 9	75 9	75 9	75 85	75 9	8 9	75 9	75 9	75 9	75 9	75 9	75 9	8 9	75 9	75 9
5	8 9	75 9	8 9	8 9	75 9	75 85	8 9	8 95	75 9	8 95	8 95	75 9	75 9	75 9	75 9	8 9	75 9	8 9	8 95	8 9	8 9	75 9	8 9	75 9	75 95
6	8 9	75 9	8 9	8 9	75 9	8 85	75 9	8 9	8 9	8 9	8 9	75 9	8 9	8 9	8 9	8 9	8 9	75 9	75 9	8 9	8 9	8 9	75 95	8 9	75 9
7	75 9	7 9	8 9	75 9	7 9	7 75	75 9	75 9	75 9	7 85	75 85	75 9	75 9	75 85	75 9	8 95	75 9	75 9	75 85	75 9	75 85	75 9	7 9	75 9	75 9
8	75 9	75 9	8 9	75 9	75 9	75 9	75 9	75 95	75 9	75 85	75 85	75 9	75 9	75 9	75 9	8 95	75 9	75 9	75 85	75 9	75 9	75 9	75 8	75 9	75 9
9	8 9	7 85	8 9	75 85	7 85	75 9	75 9	75 9	8 9	8 9	75 9	75 9	75 9	75 85	75 9	8 9	75 9	8 9	75 9	75 9	75 9	75 9	75 85	75 9	8 9
10	8 9	7 85	8 9	75 9	7 85	75 8	75 95	75 95	75 9	7 9	75 85	75 85	75 9	75 9	75 9	8 9	75 9	8 9	75 9	75 9	75 85	75 85	7 8	75 9	75 9

Fuente: Los autores

Figura 36. Grafica de barras (resultados BAYSOM, rangos), viga








Fuente: Los autores

Luego de someter a prueba la metodología BAYSOM en lo que respecta a la generación de rangos, los resultados obtenidos en el caso de estudio de la viga son altamente satisfactorios, ya que se puede estimar el rango de valores en que puede variar el porcentaje de daño en 9 de cada 10 elementos, siendo en sí mismo un importante indicador a tener en cuenta en las labores de mantenimiento, ya que mediante el análisis de estos rangos puede concebirse una primera imagen de la distribución real de daño a lo largo de la estructura, y de la urgente necesidad o no de un mantenimiento correctivo-predictivo.

Es importante resaltar que debe contarse con una base de datos capaz de agrupar los más diversos escenarios, ya sea por el volumen de escenarios con que dicha base de datos fue construida, o por la calidad de los mismos (evitar datos repetidos o demasiado similares). Debido al bajo costo computacional de esta metodología (Generación base de datos 4000 escenarios|15 min. Proceso de inferencia y obtención de rangos|8 min.), y su alto porcentaje de aciertos, se proyecta como un enfoque teórico disponible para aplicaciones reales.

Figura 37. Simbología y código de colores (BAYSOM, valores), viga

X	X (Coeficiente del modulo de elasticidad x 100)
	Predicción correcta ($\pm 0,03$ del valor real)
	Predicción aproximada ($\pm 0,05$ del valor real)
	Predicción de estado (Alta desviación del valor real, acierto en el estado del elemento)
	Falso positivo (Predicción de daño en elemento no-dañado)
	Falso negativo (Predicción de no-daño en elemento dañado)

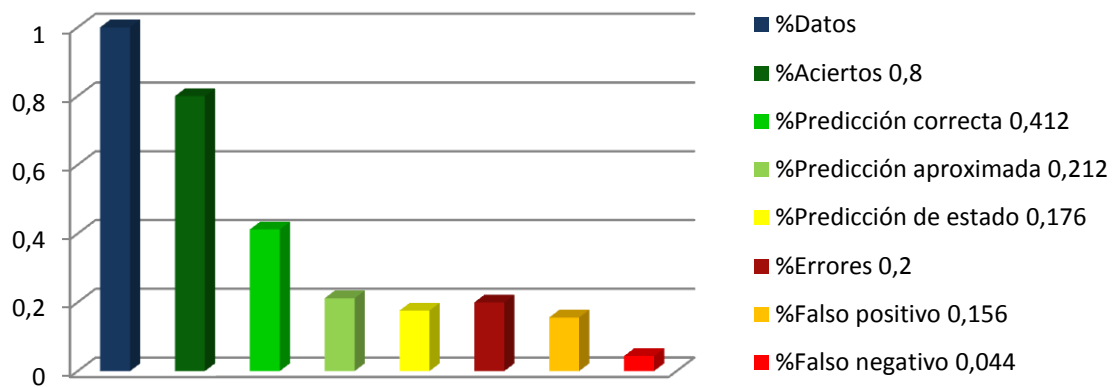
Fuente: Los autores

Figura 38. Resultados de las predicciones BAYSOM (valores), viga

	Sin ruido															Ruido 0,005					Ruido 0,01				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	84	79	76	88	85	87	85	90	77	75	85	87	88	82	89	89	90	85	86	90	76	83	81	76	81
2	87	76	83	88	86	79	83	84	80	84	75	87	85	76	79	88	81	85	81	89	83	77	81	78	85
3	86	82	90	77	82	83	81	84	83	76	76	76	83	77	77	80	80	77	77	84	79	79	81	76	76
4	83	77	82	83	75	99	90	86	89	83	78	80	76	83	81	87	78	87	77	83	86	76	81	86	78
5	89	81	83	82	75	77	86	81	77	93	82	82	80	76	81	85	75	80	93	83	81	77	88	75	79
6	89	82	83	80	79	81	90	81	81	84	83	78	83	81	83	83	87	84	85	85	80	80	80	86	89
7	80	76	84	78	72	74	78	77	87	73	78	82	85	75	77	82	83	88	82	90	75	77	71	78	84
8	77	77	82	78	77	75	79	86	84	75	76	79	83	82	88	81	76	82	83	77	76	78	78	77	76
9	80	84	85	83	82	81	79	89	83	82	88	82	79	85	80	84	83	88	80	87	76	81	77	79	84
10	86	73	87	82	81	79	77	78	77	87	78	76	80	85	80	81	81	84	78	78	82	81	71	89	76

Fuente: Los autores

Figura 39. Grafica de barras (resultados BAYSOM, valores), viga



Fuente: Los autores

Es evidente la calidad de los resultados obtenidos por BAYSOM en cuanto a la estimación de porcentajes de daño, al tener como escenario de análisis una viga discretizada en 10 elementos. Como lo muestra la grafica (Ver Figura 39.) el porcentaje de aciertos de dicha metodología es considerable, lográndose la predicción del estado de daño en 8 de cada 10 elementos, además de obtenerse el valor exacto de daño en 4 de estos 8 elementos, y aunque el porcentaje de errores alcanzo un valor del 20% es importante aclarar que el porcentaje de errores graves (Falsos negativos) no supero la barrera del 5%, ya que la mayoría de errores se debieron a predicciones que catalogaban un elemento como dañado aun cuando este contaba con buenas condiciones estructurales, lo cual traducido al lenguaje del mantenimiento tan solo ocasionaría revisiones innecesarias ya que en el momento de accesar a la región supuestamente afectada se determinaría que su porcentaje de daño no es tan considerable, a diferencia de las graves repercusiones que tiene un falso negativo. Gracias a la obtención de estos importantes resultados se puede decir que el enfoque BAYSOM puede ser considerado como una ágil metodología, capaz de brindar predicciones lo suficientemente confiables como para ser aplicadas en entornos de mantenimiento real.

Figura 40. Simbología y código de colores (BAYSOM, rangos), armadura

X	X (Coeficiente del modulo de elasticidad x 100)
X Y	X (Frontera inferior del rango) Y (Frontera superior del rango)
	Predicción correcta (Valor de daño real ubicado dentro del rango)
	Predicción aproximada (Valor de daño real ubicado a $\pm 0,03$ de alguna de las fronteras)
	Error (Valor de daño real completamente fuera del rango)

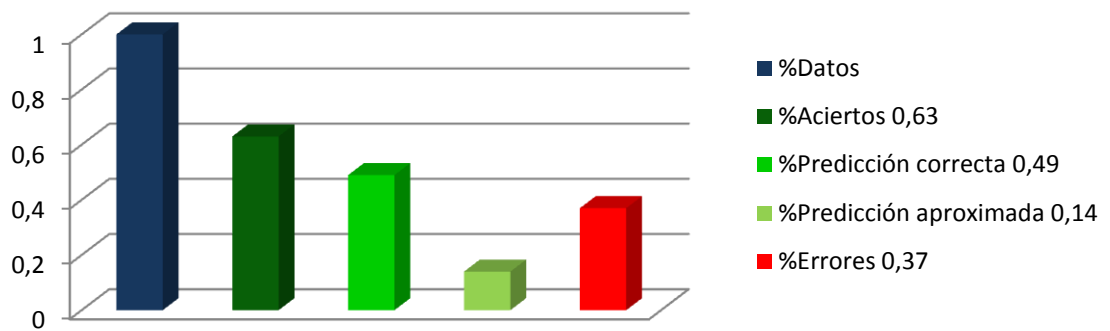
Fuente: Los autores

Figura 41. Resultados de las predicciones BAYSOM (rangos), armadura

	Sin ruido																				Ruido 0,005				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	9 1	8 9	75 9	8 1	8 85	8 1	85 1	9 1	85 1	85 1	9 1	85 9	75 1	8 1	9 1	75 85	9 1	85 1	9 1	85 1	8 1	75 1	8 1	8 85	8 1
2	75 8	75 1	7 9	7 9	75 8	7 8	7 85	8 75	7 85	85 9	75 8	85 9	7 9	75 8	9 1	8 9	7 75	7 8	7 75	7 85	7 8	7 85	7 9	7 8	7 8
3	8 9	85 1	7 1	7 9	9 1	9 1	7 9	8 85	7 9	9 1	9 1	9 1	7 1	9 1	9 1	8 1	75 9	7 9	75 9	7 9	9 1	75 1	7 9	9 1	9 1
4	7 75	75 1	7 9	7 9	7 1	7 75	7 9	8 85	7 9	7 85	75 8	85 9	7 9	7 75	75 9	8 1	7 8	7 9	75 8	7 9	7 1	7 9	7 1	7 1	7 1
5	9 1	8 1	75 9	7 9	8 9	75 1	7 9	85 1	7 9	85 1	75 8	85 1	7 9	8 1	9 1	8 1	75 85	8 9	8 85	7 9	75 1	7 9	7 1	85 1	8 1
6	9 1	85 1	75 9	7 85	9 1	9 1	85 9	9 1	85 9	9 1	9 1	9 1	8 85	9 1	9 1	85 1	9 1	9 1	9 1	85 9	9 1	85 9	8 1	9 1	9 1
7	9 1	8 85	7 9	7 9	8 1	8 9	85 9	8 9	85 9	8 1	8 85	9 1	75 9	8 1	85 1	8 1	9 1	85 1	9 1	85 9	8 9	8 1	7 1	8 1	8 1
8	75 8	8 85	75 1	75 9	8 85	7 75	85 9	7 8	85 9	7 75	8 85	85 9	75 85	7 75	8 85	8 9	7 75	75 8	7 75	85 9	7 75	75 8	75 85	7 75	8 85
9	85 9	8 9	75 9	75 1	8 85	75 8	85 1	8 9	85 1	75 8	8 85	75 85	75 1	7 8	8 85	75 85	75 9	85 9	75 9	85 1	7 8	8 9	8 9	75 8	8 85
10	85 1	8 1	75 9	75 1	85 9	85 1	75 9	7 1	75 1	85 9	9 1	8 1	75 9	85 9	85 9	8 1	8 9	75 1	75 85	75 9	85 9	75 9	75 1	85 9	85 9
11	8 1	8 1	7 1	7 9	75 8	75 1	7 1	85 1	7 1	8 85	7 75	75 85	7 9	75 9	85 9	8 9	75 9	7 9	8 9	7 1	75 9	7 9	7 1	75 9	75 9
12	8 1	8 9	7 9	7 9	8 1	7 8	7 85	85 1	7 9	9 1	7 75	75 1	7 9	7 1	9 1	8 1	7 9	75 85	75 85	7 85	7 1	7 85	75 1	75 1	7 1
13	85 1	8 1	7 9	7 9	75 8	75 1	7 9	8 9	7 9	8 85	9 1	7 1	7 1	75 9	85 9	8 1	9 1	75 1	85 1	7 9	75 9	7 9	75 9	75 9	75 9

Fuente: Los autores

Figura 42. Grafica de barras (resultados BAYSOM, rangos), armadura



Fuente: Los autores

En el caso de la armadura y totalmente opuesto al estudio llevado a cabo en la viga anteriormente por la metodología BAYSOM, no se logró obtener un porcentaje de aciertos lo suficientemente alto, como para considerar a dicha metodología como confiable en lo que respecta a la predicción de rangos más probables donde el porcentaje de daño de un elemento puede estar alojado, por lo que pueden formularse dos diferentes hipótesis que explicarían la evidente dificultad de obtener un mayor número de aciertos en la predicción, la primera de ellas está relacionada con la base de datos utilizada en el cálculo de aquellos rangos más probables, pues al contarse con una estructura compuesta de un mayor número de elementos (13) a comparación de la viga, dicha estructura puede ofrecer una cantidad inmensa de escenarios de daño de los cuales solo unos pocos toman parte en la construcción de la base de datos, corriendo el riesgo de que el escenario a predecir se encuentre notoriamente distanciado de dicho grupo de datos, errando de esta manera en la predicción del estado de sus elementos. Una segunda hipótesis es aquella que relaciona la complejidad del sistema con la dificultad en la predicción del estado de daño actual de dicha estructura, por lo cual se requiere además de un mayor volumen de información la obtención de nuevos criterios o expresiones analíticas capaces de caracterizar de una manera más rigurosa cada escenario de daño.

Figura 43. Simbología y código de colores (BAYSOM, valores), armadura

X	X (Coeficiente del modulo de elasticidad x 100)
	Predicción correcta ($\pm 0,03$ del valor real)
	Predicción aproximada ($\pm 0,05$ del valor real)
	Predicción de estado (Alta desviación del valor real, acierto en el estado del elemento)
	Falso positivo (Predicción de daño en elemento no-dañado)
	Falso negativo (Predicción de no-daño en elemento dañado)

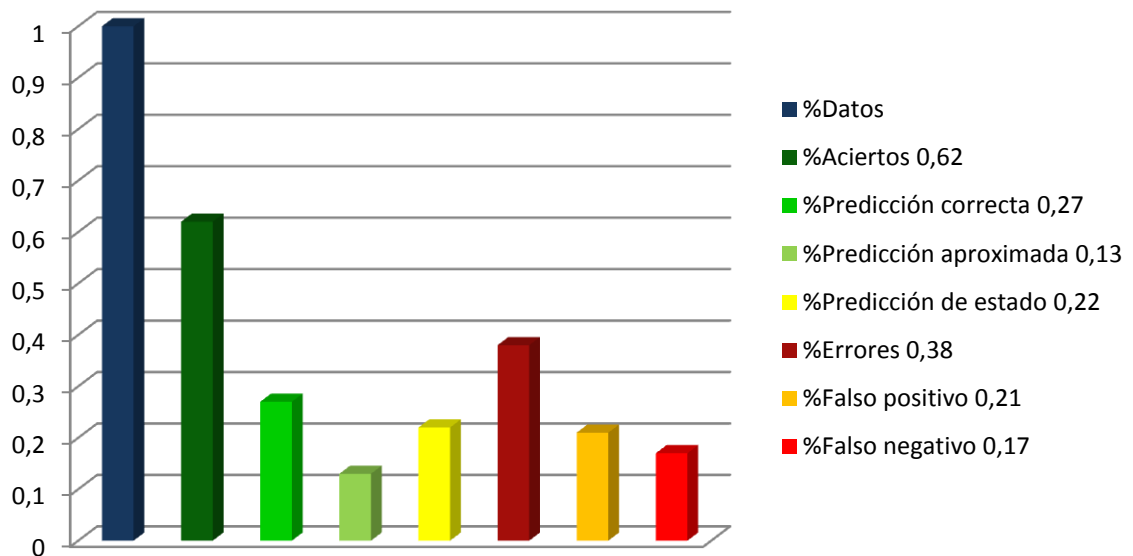
Fuente: Los autores

Figura 44. Resultados de las predicciones BAYSOM (valores), armadura

		Sin ruido																		Ruido 0,005						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1		94	82	82	86	82	97	99	99	97	85	96	88	87	90	95	81	97	89	91	99	81	92	81	81	90
2		76	99	82	83	79	74	84	78	74	86	77	88	87	80	95	81	73	75	72	84	73	70	77	75	75
3		85	95	97	70	91	94	75	82	73	99	90	95	99	96	92	87	83	84	77	75	97	90	76	92	96
4		72	86	75	73	98	71	76	81	71	77	77	90	85	72	89	76	71	80	75	76	87	77	84	86	94
5		94	80	79	87	81	87	90	96	84	87	80	87	87	87	91	85	85	89	81	90	88	80	77	85	81
6		97	88	82	71	1	90	86	93	88	93	99	91	82	97	92	84	92	95	90	86	95	86	92	95	99
7		91	83	89	87	81	81	89	89	89	81	83	90	89	82	88	1	90	89	90	89	82	89	91	82	82
8		79	80	80	85	81	75	88	78	88	72	84	87	76	70	81	79	73	76	72	88	72	76	79	71	80
9		90	83	82	76	85	77	85	89	88	80	81	85	83	77	83	96	82	87	86	85	79	86	86	79	82
10		93	96	86	86	85	92	78	92	93	88	95	98	83	89	86	96	83	91	79	78	88	86	75	85	86
11		85	84	78	83	79	85	89	97	81	85	71	79	77	86	87	94	88	73	88	89	78	85	98	85	75
12		92	87	79	76	96	72	73	88	88	97	74	95	83	79	99	89	75	83	80	73	75	84	87	77	76
13		95	85	81	71	79	80	82	87	79	81	91	87	96	82	89	81	97	82	96	82	82	74	83	87	82

Fuente: Los autores

Figura 45. Grafica de barras (resultados BAYSOM, valores), armadura



Fuente: Los autores

Tal y como se ha dejado expresado en varias secciones del presente documento la metodología SOM por si sola es demasiado susceptible a la presentación de escenarios de daño capaces de brindar respuestas equivalentes en cuanto al comportamiento modal de la estructura, pero lejanos de los valores de daño real que se pretenden predecir, por lo que su buen desempeño depende enteramente de la capacidad de limitar el rango de valores en que la red puede ser entrenada, debido a los considerables porcentajes de error obtenidos en la generación de los rangos, lo cual ya fue discutido en una sección anterior, los porcentajes de error en la cuantificación del valor de daño son igualmente considerables, por lo que los valores de confiabilidad presentes en la implementación de la metodología BAYSOM en el análisis de una armadura de 13 elementos pueden no llegar a ser muy altos, contrastando fuertemente con los resultados obtenidos en el estudio de una viga simplemente apoyada discretizada en 10 elementos donde sus porcentajes de aciertos convierten a la metodología BAYSOM en una herramienta digital capaz de desempeñarse en entornos de predicción real, logrando la consecución de uno de los objetivos primordiales de la presente investigación.

5. CONCLUSIONES

- La presente investigación pudo establecer que con un porcentaje de aciertos para la red SOM del 53.8% y de un 66% en el caso de la red bayesiana, y contando con una relación en tiempo de computo de 2 a 1 para la red bayesiana con respecto a la red SOM, la implementación de un único enfoque en la detección y cuantificación de daños en estructuras no cuenta con los valores de confiabilidad necesarios.
- Pudo concluirse que solo mediante el trabajo conjunto de los dos enfoques bajo análisis, se podrían llegar a suplir los requerimientos propuestos por la presente investigación, dando paso al desarrollo de la metodología BAYSOM, donde se combino la capacidad de inferencia bayesiana para obtener los rangos más probables en donde se aloja el porcentaje de daño, con la habilidad de las redes SOM para encontrar en una base de datos entrenada según dichos rangos, la respuesta más cercana a la real, acerca del porcentaje de daño presente en cualquier elemento bajo estudio.
- Los resultados obtenidos mediante la implementación de la metodología BAYSOM en lo referente a la predicción de los diferentes porcentajes de daño presentes en una viga discretizada en 10 elementos, cuenta con un porcentaje de acierto del 90% en la identificación de los rangos y con un 80% de aciertos en la estimación del porcentaje de daño de cada elemento, con una relación de tiempos de 4 a 1 para la inferencia bayesiana con respecto a la predicción SOM, tomando un tiempo de 25 min , si esta operación fuera llevada a cabo por un ordenador similar al implementado en la presente investigación con las siguientes características: Procesador Intel

core (TM) duo @ 2.16 GHz, memoria RAM de 2Gb, SO Windows 7 Ultimate, MATLAB® 7.8.0(R2009a).

- En lo que respecta a tiempos de cómputo se obtuvo un menor lapso de procesamiento en el caso de las redes SOM, tanto para viga como para armadura, aunque se pudo evidenciar lo susceptible que es su modelo de procesamiento al número de elementos bajo estudio:
SOM (Viga [0,3 - 8 min] – Armadura [1 – 15 min])
Bayes (Viga [2 - 25 min] – Armadura [3 – 25 min])
- En el análisis llevado a cabo en una armadura de 13 elementos se obtuvo un porcentaje de efectividad del 67%, concluyendo que: el enfoque BAYSOM por sí solo no ofrece los valores de confiabilidad que respalden su implementación en este caso de estudio, por lo que se requiere determinar nuevos criterios especializados en la obtención de valores característicos, los cuales permitan una mejor diferenciación de cada uno de los escenarios analizados y sus porcentajes de daño asociados, además de una base de datos lo suficientemente robusta, capaz de contener información estadística sobre el mayor número posible de escenarios de daño.

6. RECOMENDACIONES

- Debido a la escases de criterios característicos capaces de facilitar la labor predictiva en lo referente al análisis de una armadura de 13 elementos, se sugiere la implementación en trabajos futuros de enfoques metodológicos capaces de contemplar la disipación de energía en la estructura, ya que si se implementa el análisis modal el tener en cuenta solo las matrices de rigidez y masas es insuficiente para definir una situación característica en cualquier estructura de relativa complejidad, por lo que se aconseja tener en cuenta la matriz de amortiguamiento con el fin de disponer de un mayor número de parámetros característicos y así llegar a desarrollar el método correcto de identificación del escenario a predecir como ya se mostro posible en el caso de la viga con la metodología BAYSOM.
- Se propone la elaboración de un algoritmo capaz de evaluar la similitud de un dato de entrenamiento con respecto a otro ya existente dentro de la base de datos, omitiendo dicho dato y dando paso a la generación de un nuevo escenario, lo cual evitaría que dos escenarios de daño que pueden catalogarse como idénticos sean agrupados dentro de la misma tabla de probabilidad condicional, aumentando sus valores de probabilidad y de esta manera afectando el enfoque estadístico que se pretende brindar con la metodología bayesiana, donde se busca el análisis de escenarios mucho mas dispersos a través de la muestra.
- Se recomienda contar con una base de datos previamente almacenada, de preferencia expandible, donde sea posible alimentarla continuamente con nuevos datos, aumentando de esta manera la calidad de las predicciones.

BIBLIOGRAFÍA

ANDERSON, James. Redes neurales. 1ed. México D.F.: Alfaomega Grupo Editor, 2007. 616p. ISBN 978-0-262-01144-1

ATIENZA, Ricardo. Técnicas de actualización aplicadas a la detección de daños. Madrid: Escuela Técnica Superior de Ingenieros Aeronáuticos. Departamento de Motopropulsión y Fluidodinámica, 2004. 172p.

BARRERA, Robinson y SOTO, Oscar Mauricio. Detección de daños en estructuras utilizando la técnica de optimización algoritmos genéticos. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica, 2010. 153p.

CAICEDO, Eduardo. Mapas Auto-Organizados: El Modelo de KOHONEN [online]. 2007 [citado 3 de marzo de 2011; 10:00:00]. Disponible en Internet: http://eisc.univalle.edu.co/materias/Fundamentos_De_Computacion_Inteligente/material/RN_Kohonen.pdf

GERMANO, Tom. Self Organizing Maps [online], 1999 [citado 5 de marzo de 2011; 20:00:00]. Disponible en Internet: <http://davis.wpi.edu/~matt/courses/soms/>

GÓMEZ, Iván Darío. Caracterización dinámica experimental de puentes de hormigón simplemente apoyados a partir de mediciones de vibración ambiental.

Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Maestría de Ingeniería Civil, 2010. 214p.

HAYKIN, Simon. Neural networks: a comprehensive foundation. 2 ed. Upper Saddle River, New Jersey: Prentice-Hall, 1999. 842p. ISBN 0-13-273350-1

MARTIN DEL BARRIO, Bonifacio y SANZ, Alfredo. Redes neuronales y sistemas borrosos. 3 ed. México D.F.: Alfaomega Grupo Editor, 2007. 442 p. ISBN 84-7897-743-0

MCKENNA, Frank y FENVES, Gregory. The OpenSees quick reference guide. Berkeley: University of California. Pacific Earthquake Engineering Research Center, 2001. p. 1-5

RAMÍREZ, Diego Andrés. Detección de daños en estructuras utilizando la técnica de optimización PSO. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica, 2010.142p.

RICO, José María. Determinación numérica de eigenvalores y eigenvectores. Salamanca: Universidad de Guanajuato. Departamento de Ingeniería Mecánica, 2010. p. 1-7

SOTELO, Fredy Saúl y ACEVEDO, Javier Ernesto. Identificación de daño en pórticos 2D empleando información modal y el método Particle Swarm Optimization (PSO). Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Civil, 2009.163p.

THEODORIDIS, Sergios y KOUTROUMBAS, Konstantinos. An introduction to pattern recognition: A MATLAB® Approach. 4ed. Burlington: Academic Press Elsevier, 2010. p. 1-27

THEODORIDIS, Sergios y KOUTROUMBAS, Konstantinos. Pattern Recognition. 4ed. Burlington: Academic Press Elsevier, 2009. p. 1-86

VERA, René. Análisis de vibraciones en elementos funcionales del faro izquierdo del automóvil New Beetle 2.5 Versión USA (NB-NAR). Cholula: Universidad de las Américas Puebla. Escuela de Ingeniería y Ciencias. Departamento de Ingeniería Industrial y Mecánica, 2006. 120p.

ANEXOS

Anexo A. ANÁLISIS FÍSICO DE LA FORMA DEL MODO

Con las vibraciones libres no existe la aplicación externa de cargas por lo tanto se define como "cero", por este motivo la estructura a analizar solo tiene vibraciones por los efectos de las condiciones iniciales. Este análisis de vibración raramente es realizado, ya que los casos que se analizan la mayoría está sujeta a cargas externas y esto nos daría que tal condición es diferente de cero. Sin embargo, la solución del problema de las vibraciones con amortiguación "Cero", provee las características dinámicas más importantes de una estructura: las frecuencias naturales y las formas de los modos. Enseguida se presentaran las formulas más no la deducción de tales, con el motivo de explicar porque se hace este análisis y su entendimiento para el análisis modal:

$$[M] \{\ddot{y}\} + [K]\{y\} = 0 \quad (A.1)$$

Se realizara un análisis modal para un sistema de N-grados de libertad, por lo tanto se asume que una posible solución es la forma:

$$\{y\}_i = \{\Phi\}_i \sin(\omega_i * t - \alpha_i) \quad (A.2)$$

Donde $\{\Phi\}_i$ es la i- forma del modo con una correspondiente frecuencia natural circular ω_i y ángulo de fase α_i . Substituyendo la ecuación (A.2) en la ecuación (A.1) y se elimina $\sin(\omega_i t - \alpha_i)$ se obtiene:

$$([K] - \omega_i^2[M])\{\Phi\}_i = \{0\} \quad (A.3)$$

La ecuación (A.3) puede ser escrita como un sistema de N- ecuaciones por lo cual puede ser dada la ecuación (A.4)

$$\begin{pmatrix} k_{11} - \omega_1^2 m_1 & k_{12} & \dots & k_{1N} \\ k_{12} & k_{22} - \omega_2^2 m_2 & \dots & k_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \dots & k_{NN} - \omega_N^2 m_N \end{pmatrix} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix} \quad (A.4)$$

La enunciación de las ecuaciones (A.3) y (A.4) es un problema matemático importante, el cual se denomina “eigenproblema” (valor propio), este valor propio aparece en otros tipos de análisis de la rama de ingeniería y no solo es un segmento de los problemas dinámicos. La particularidad del problema valor propio, es que la solución no es única. Para entrar en detalle, $\{\phi\}_i$ que se calcula en análisis modal no representa las amplitudes del sistema bajo vibración libre, pero aun así, los cocientes normalizados de la amplitud, cuando están combinados correctamente, pueden proporcionar la reacción dinámica del sistema.

La función del análisis modal es el cómputo de ω_i y la reciprocidad correspondiente de $\{\Phi\}_i$, que compensa a la ecuación (A.3) obteniendo una solución no trivial, por ejemplo, una solución por el cual todos los ω_i y $\{\Phi\}_i$ no son cero, requiere que el determinante de la ecuación (A.3) sea cero, por eso:

$$\text{Det}([K] - \omega_i^2[M]) = 0 \quad (A.5)$$

La ecuación (A.5) es una ecuación polinomial de ángulo N en ω_i^2 . Este polinomial es conocido como la ecuación característica del sistema. Para cada solución ω_i^2 ($i = 1, 2, \dots, N$) de la ecuación característica podemos resolver la ecuación (A.4) para $\{\phi\}_i$. Una solución ω_i^2 de la ecuación característica es llamada un eigenvalor (valor propio). Para cada uno de los eigenvalores corresponde un eigenvector, lo cual significa, que por un sistema con N grados de libertad hay N eigenvectores y

sus correspondientes eigenvalores. En dinámica, ω_i es llamado la frecuencia natural circular y los correspondientes eigenvectores son llamados la forma del modo; esto es, en términos del eigenvector y la forma del modo son equivalentes. La frecuencia natural circular más pequeña ω_1 es llamada la *frecuencia circular fundamental*, y su correspondiente forma del modo $\{\phi\}_1$ es llamado el *modo fundamental de vibración*.

Aquí debe de ser de ayuda exhibir una exégesis física de una forma de modo. La ecuación (A.3) formula equilibrio entre la inercia y las fuerzas de dureza en una estructura en tiempo t. si se reformula la ecuación (A.3) de tal forma que:

$$[K]\{\Phi\}_i = \omega_i^2[M]\{\Phi\}_i \quad (A.6)$$

Una forma del modo puede ser vista como una deflexión estática resultante de la fuerzas del lado de la mano derecha de la ecuación (A.6).

Anexo B. ARCHIVOS OPENSEES®

Programa maestro en la viga (*cuerpo.tcl*): Este programa contiene la secuencia de activación de los subprogramas para el cálculo de una situación de daño en la viga obteniendo como resultado las frecuencias y modos de vibración, en este programa se define el número de modos a calcular, el diferencial de cálculo y la sensibilidad de este, la forma de activar cada uno es usando el comando source como se observa:

```
##### Determinando grados de libertad y dimensiones #####  
model BasicBuilder -ndm 2 -ndf 3  
  
##### Caracterizando la viga #####  
  
source 1.tcl  
source 2.tcl  
source 3.tcl  
source 4.tcl  
source 5.tcl  
  
##### Inicializando #####  
  
source 6.tcl  
  
##### Creando eigenvectores #####  
  
source 7.tcl  
  
##### Definiendo análisis y características #####  
  
source 8.tcl  
  
##### Definiendo variables #####  
  
set eigFID [open EigenVal.out w]
```

```
##### Salidas #####
```

```
puts $eigFID [eigen generalized genBandArpack 4]
close $eigFID
analyze 1 0.0001
wipe
```

Programa para definir las posiciones de los centros de masa de (posición en “x” y “y”) en una vista lateral de la viga (**1.tcl**):

```
##### Definimos los nodos en la viga #####
```

```
# Node tag xCrd yCrd
node 1 0.0 0.0
node 2 0.6 0.0
node 3 1.2 0.0
node 4 1.8 0.0
node 5 2.4 0.0
node 6 3.0 0.0
node 7 3.6 0.0
node 8 4.2 0.0
node 9 4.8 0.0
node 10 5.4 0.0
node 11 6.0 0.0
```

Programa que define las restricciones de los elementos de la viga, aquellos que se encuentran unidos a los pasadores (**2.tcl**):

```
##### Definiendo grados de libertad #####
```

```
# tag Dx Dy
fix 1 1 1 0
fix 11 1 1 0
```

```
##### Se ancla el primer y último nodo #####
```

Tipo de transformación usada en el cálculo (**3.tcl**):

```
##### Transformación lineal #####
```

```
geomTransf Linear 1
```

Programa que exporta la información que define la resistencia del material en cada nodo definido anteriormente, este archivo es modificado cada vez desde la codificación en MATLAB® (**4.tcl**):

```
# 4.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per Unit Volume=
2402.7696 Weight per Unit Volume =23563.122
```

```
# Section "ElasticDefault":  secTag  E  A  Iz
section Elastic 1 1.937180e+010 1.200000e-001 1.600000e-003
section Elastic 2 1.651170e+010 1.200000e-001 1.600000e-003
section Elastic 3 1.715881e+010 1.200000e-001 1.600000e-003
section Elastic 4 1.787906e+010 1.200000e-001 1.600000e-003
section Elastic 5 1.636996e+010 1.200000e-001 1.600000e-003
section Elastic 6 1.629619e+010 1.200000e-001 1.600000e-003
section Elastic 7 1.875177e+010 1.200000e-001 1.600000e-003
section Elastic 8 2.116915e+010 1.200000e-001 1.600000e-003
section Elastic 9 2.069574e+010 1.200000e-001 1.600000e-003
section Elastic 10 2.046846e+010 1.200000e-001 1.600000e-003
```

Este archivo .tcl se encarga de definir las posiciones de los nodos con respecto a los elementos ya definidos, también información sobre el material de forma general y sin daño (**5.tcl**):

```
##### Elements.tcl #####
```

```
# Material Concrete 3000 Psi E= 2.153E+10 Masa por Unidad de Vol= 2402.7696
Masa por unidad de long=288.335628  Peso por unidad de vol =23563.122
```

```

# Element "Element02":  eleTag  Nodel  NodeJ  NIP  secTag  geoTranTag
<-mass massDens>
element dispBeamColumn  1    1    2    10    1    1  -mass +2.883324E+002
element dispBeamColumn  2    2    3    10    2    1  -mass +2.883324E+002
element dispBeamColumn  3    3    4    10    3    1  -mass +2.883324E+002
element dispBeamColumn  4    4    5    10    4    1  -mass +2.883324E+002
element dispBeamColumn  5    5    6    10    5    1  -mass +2.883324E+002
element dispBeamColumn  6    6    7    10    6    1  -mass +2.883324E+002
element dispBeamColumn  7    7    8    10    7    1  -mass +2.883324E+002
element dispBeamColumn  8    8    9    10    8    1  -mass +2.883324E+002
element dispBeamColumn  9    9   10    10    9    1  -mass +2.883324E+002
element dispBeamColumn 10   10   11    10   10    1  -mass +2.883324E+002

```

El programa (**6.tcl**) solo contiene el comando de inicio para el cálculo de los modos de vibración indicando que se ha terminado la fase de introducir la información que define el problema.

Initialize

Este programa tan solo contiene las ubicaciones y los respectivos nombres para cada vector de salida (.out) calculado ya, organizado de forma ascendente, aunque algunos de los archivos creados se encuentren vacios (**7.tcl**):

```
##### Guardando eigenvectors #####
```

```

# Node Recorder "EigenVector":  fileName  <nodeTag>  dof  respType
recorder Node -file EigenVector_1.out -time -dof 2 eigen1
recorder Node -file EigenVector_2.out -time -dof 2 eigen2
recorder Node -file EigenVector_3.out -time -dof 2 eigen3
recorder Node -file EigenVector_4.out -time -dof 2 eigen4
recorder Node -file EigenVector_5.out -time -dof 2 eigen5
recorder Node -file EigenVector_6.out -time -dof 2 eigen6
recorder Node -file EigenVector_7.out -time -dof 2 eigen7
recorder Node -file EigenVector_8.out -time -dof 2 eigen8
recorder Node -file EigenVector_9.out -time -dof 2 eigen9
recorder Node -file EigenVector_10.out -time -dof 2 eigen10

```

Definimos relaciones entre elementos y parámetros de resolución del problema **(8.tcl)**:

```
# AnalysisOptn "EigenDefault": Type: Eigen Definiendo análisis y características #
# Construyendo el objeto Constraint Handler relación entre los grados de libertad #
constraints Plain

##### Convergence Test (Prueba de tolerancia) #####
test NormUnbalance +1.000000E-006 25 0 2

# Integrator (Determina significado de los términos en el sistema de ecuaciones) #
integrator Newmark +5.000000E-001 +2.500000E-001 +0.000000E+000
+0.000000E+000 +0.000000E+000 +0.000000E+000

##### Solution Algorithm (Construye el objeto para determinar
la secuencia de pasos para resolver las ecuaciones) #####
algorithm Newton

##### DOF Numberer (Determina la asignación entre el
número de ecuaciones y los grados de libertad) #####
numberer RCM

##### System of Equations (Almacena y resuelve el
sistema de ecuaciones en el análisis) #####
system ProfileSPD

##### Analysis Type (Comando que tiene actualmente todo
disponible para implementar una estrategia de solución) #####
analysis Transient
```

Programa maestro de la estructura (*armadura.tcl*):

```
##### ARMADURA1.tcl #####
##### Inicio #####
model BasicBuilder -ndm 2 -ndf 2
##### Geometría #####
source Nodos.tcl
##### Puntos de restricción #####
source DPConstraint.tcl
##### Masas modales #####
source Masas.tcl
##### Materiales #####
source Materiales.tcl
##### Elementos #####
source Elementos.tcl

initialize
```

Como observamos al igual que en la viga este archivo es el encargado de definir el tipo de modelo, además de activar los demás programas de OpenSees que contienen la definición del problema, así como las restricciones, modelos de cálculo e información del material.

Posteriormente la primera acción es definir las posiciones de los nodos en el plano que contiene a la armadura en una vista lateral (“x” y “y”).

```
# Node tag xCrd yCrd
node 1 +0.0 +0.0
node 2 +1.83 +0.0
node 3 +3.66 +0.0
node 4 +5.49 +0.0
node 5 +7.32 +0.0
node 6 +1.83 +2.44
node 7 +3.66 +2.44
node 8 +5.49 +2.44
```

Se definen los nodos anclados debido a su posición (***DPConstraint.tcl***):

```
##### DPConstraint.tcl #####
```

```
# SPC tag Dx Dy
fix 1 1 1
fix 5 1 1
```

```
##### Anclamos el primer nodo y el quinto nodo #####
```

Posiciones de los centros de masa de elementos en el plano (***Masas.tcl***):

```
##### Masas #####
```

```
# Mass tag mx my
mass 1 8.39 8.39
mass 2 15.97 15.97
mass 3 10.45 10.45
mass 4 15.97 15.97
mass 5 8.39 8.39
mass 6 13.1 13.1
mass 7 21.48 21.48
mass 8 13.1 13.1
```

```
##### Sección transversal #####
```

Programa que reescrito desde MATLAB® importa la información de la situación de daño de cada elemento (***Materiales.tcl***):

```
##### MATERIALES #####
```

```
# Material "ElasticDefault": matTag E eta
uniaxialMaterial Elastic 1 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 2 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 3 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 4 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 5 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 6 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 7 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 8 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 9 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 10 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 11 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 12 2.000000e+011 0.000000e+000
uniaxialMaterial Elastic 13 2.000000e+011 0.000000e+000
```

Luego se definen las relaciones de posición entre los nodos y los elementos (***Elementos.tcl***):

```
##### Elementos #####
```

```
#Element"Element01":eleTag Nodel NodeJ A matTag
element truss 1 1 2 +4.000000E-4 1
element truss 2 2 3 +4.000000E-4 2
element truss 3 3 4 +4.000000E-4 3
element truss 4 4 5 +4.000000E-4 4
element truss 5 1 6 +4.000000E-4 5
element truss 6 2 6 +4.000000E-4 6
element truss 7 2 7 +4.000000E-4 7
element truss 8 3 7 +4.000000E-4 8
element truss 9 4 7 +4.000000E-4 9
element truss 10 4 8 +4.000000E-4 10
element truss 11 5 8 +4.000000E-4 11
element truss 12 6 7 +4.000000E-4 12
```

element truss 13 7 8 +4.000000E-4 13

El archivo que se encarga de guardar los vectores de salida (.out), calculados en la secuencia del programa (**Analysis2Recorder.tcl**):

```
##### Analysis_2_Recorder.tcl #####  
  
# Node Recorder "EigenVector": fileName <nodeTag> dof respType  
recorder Node -file EigenVec_1.out -time -dof 1 2 eigen1  
recorder Node -file EigenVec_2.out -time -dof 1 2 eigen2  
recorder Node -file EigenVec_3.out -time -dof 1 2 eigen3  
recorder Node -file EigenVec_4.out -time -dof 1 2 eigen4  
recorder Node -file EigenVec_5.out -time -dof 1 2 eigen5  
recorder Node -file EigenVec_6.out -time -dof 1 2 eigen6  
recorder Node -file EigenVec_7.out -time -dof 1 2 eigen7  
recorder Node -file EigenVec_8.out -time -dof 1 2 eigen8  
recorder Node -file EigenVec_9.out -time -dof 1 2 eigen9  
recorder Node -file EigenVec_10.out -time -dof 1 2 eigen10  
recorder Node -file EigenVec_11.out -time -dof 1 2 eigen11  
recorder Node -file EigenVec_12.out -time -dof 1 2 eigen12
```

Por último se tiene el programa que define la forma de cálculo del modo de vibración (**Analysis3opt.tcl**):

```
##### AnalysisOptn_3.tcl #####  
  
##### AnalysisOptn "EigenDefault": Type: Eigen #####  
  
##### Constraint Handler #####  
  
constraints Plain  
  
##### Convergence Test #####  
  
test NormUnbalance +1.000000E-006 25 0 2
```

Integrator

integrator Newmark +5.000000E-001 +2.500000E-001 +0.000000E+000
+0.000000E+000 +0.000000E+000 +0.000000E+000

Solution Algorithm

algorithm Newton

DOF Numberer

numberer RCM

System of Equations

system ProfileSPD

Analysis Type

analysis Transient

Anexo C. ARCHIVOS MATLAB®

Aclaraciones: Los títulos sombreados de color oscuro corresponden a los nombres de los programas principales. Los títulos sombreados con colores claros son los nombres de funciones específicas activadas dentro de un programa maestro o desde otra función.

Secuencia de uso:

1. Insertamos en la matriz de **Wne.mat** las frecuencias naturales de los 4 primeros modos de vibración y en la matriz llamada **EigenVece.mat** los modos de vibración de los 4 primeros modos de vibración sin tener en cuenta las posiciones del primer y último elemento en la viga debido a que están restringidos sus movimientos y su posición es cero, y en el caso de la armadura la posición de los elementos restringidos por los pasadores.
2. Si queremos crear una base de datos nueva, simplemente corremos el programa **BysGua.m** definiendo el tamaño de la base de datos nueva y también si se desea cambiarle el nombre de la nueva base de datos. (tener en cuenta que al crear una base de datos nueva, el nombre de esta debe coincidir con el utilizado en los programas de cálculo, sino se estará utilizando la base de datos predeterminada).
3. Los diferenciales de separación se encuentran dentro de la función **BayesFinal.m** y por defecto están establecidos en sus valores medios (son tres valores comprobados por el autor y se encuentran en forma de comentario), en busca de modificar la sensibilidad de la respuesta.

4. Corremos el programa **BayesMaestro.m** y esperamos la respuesta de las matrices **Ran.mat**, observando los rangos de daño en los que se encuentra cada elemento.
5. Abrimos la carpeta de **SOM_Practica_Viga** o **SOM_Practica_Armadura**, y abrimos el programa de **SetEntreViga.m** o **SetEntreArma.m**, si se desea cambiar el número de datos del set de entrenamiento.
6. Corremos el programa de **SOMVIGA.m** o **SOMARMA.m**, y observamos los resultados obtenidos.
7. Si desea realizar algún muestreo puede abrir los programas de validación de la viga o armadura y modificar el número de repeticiones para entrenar la red SOM o las combinaciones de tamaños de dicha red, para luego obtener la respuesta por medio de la matriz **resp.mat**.

Primer bloque de procesamiento (metodología Bayesiana)

BysGua.m

```
%Programa para construir la estructura de la data para el entrenamiento de la Red Bayesiana
```

```
clear all  
close all  
clc
```

```
fid=fopen('Bay.data','wt');
```

```
NDatos=4000;
```

```
N=10;
```

```
NElemeD=10;
```

```
Modos=4;
```

```

for k=1:NDatos;
Dan=BysDan;
for j=1:9*Modos;
    fprintf(fid, '%12.5g\t', Dan(1, j));
end
for j=9*Modos+1:9*Modos+10;
    fprintf(fid, '%8.2g\t', Dan(1, j));
end
fprintf(fid, '\n');
end

fclose(fid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Dan=BysDan

cd      'C:\Users\Usuario\Desktop\INGENIERIA\REDES    SOM    vs    BAYES
NET\Trasteo1\EIGENS\10eigen'
clear all
close all
clc

%% ESCRITURA DE DATOS

NElemeD = 10; % Numero de elementos dañados
N = 10; % Numero de elementos
con = 1;      %Contador para el generador de daño aleatorio
Ei = 2.153E10; %Modulo de rigidez del material

Ruido1 = 0.02;
Ruido2 = 0.02;
D = ones(1, NElemeD);
Dano = ones(1, N);
Elem = 1:1:N;

%% GENERACIÓN DE LOS DISTINTOS ESCENARIOS DE DAÑOS PARA CADA ELEMENTO
DAÑADO

while con-1<NElemeD && con-1 < N
    Elem(con) = con;
    D(con) = random('unif', 0.7, 1, 1);
    if D(con)<=0.05
        D(con) = 0.05;
    end
    con = con+1;
end

fila = length(Elem);

```

```

%% ORGANIZACIÓN DEL VECTOR DE DAÑOS (POR ORDEN DE ELEMENTOS) PARA
INTRODUCIR EN OPENSSES

for j=1:N
    for i=1:fila
        if j == Elem(i)
            Dano(j)=D(i);
        end
    end
end

E=Dano.*Ei;

%% CREAMOS EL ARCHIVO DE ESCRITURA PARA OPENSSES

[~,nmat] = size(E);

A = 0.12;          %% ÁREA DE LA SECCIÓN TRANSVERSAL DE LA VIGA
I = 0.0016;       %% INERCIA DE LA SECCIÓN TRANSVERSAL

fid = fopen('4.tcl', 'wt');

fprintf(fid,'# 4.tcl Material Concrete 3000 Psi E= 2.153E+10 Mass per
Unit Volume= 2402.7696 Weight per Unit Volume =23563.122');
fprintf(fid,'\n');

fprintf(fid,'# Section "ElasticDefault":      secTag      E      A      Iz ');
fprintf(fid,'\n');

for i=1:nmat
    fprintf(fid,'section Elastic %1.0f %12.6e %12.6e %12.6e',i,E(i),A,I);
    fprintf(fid,'\n');
end

fclose(fid);

%% %% Llamando el programa de OpenSees viga masa distribuida

!OpenSees cuerpo.tcl

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LECTURA DE DATOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Frecuencias %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Eigen = load('EigenVal.out');
Wne = sqrt(Eigen);
Modos=length(Wne);%%%%%%%%%% Cargando modos # frecuencias Wne
Wne = sort(Wne);

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RUIDO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ruido=ones (Modos,2);

```

```

for i = 1:Modos
    ru = round(random('unif',-1,1,1,1));
    if ru ==0
        ru = -1;
    end
    ruido(i,1) = ru*Ruido1;
    ruido(i,2) = ru*Ruido2;
end

Wne = Wne + Wne.*ruido(:,1)';
T = 2*pi./(Wne);

%% MODOS DE VIBRACIÓN
EigenVece=ones(12,Modos);

for i=1:Modos
EigenVece(:,i)=load(['EigenVector_',num2str(i),'.out'])+load(['EigenVecto
r_',num2str(i),'.out']).*ruido(i,2)';
end

EigenVece([1 2 12],:) = [];

%% ORGANIZACIÓN Y PRESENTACIÓN DE LA INFORMACIÓN

R=ones(9,Modos);
for i=1:Modos;
    R(:,i)=Wne(1,i).*EigenVece(:,i);
end

R=R';
Dan=ones(1,9*Modos+10);

for n=0:Modos-1;
    z=n*9+1;
    for i=0:8;
        Dan(1,z+i)=R(n+1,i+1);
    end
end

for i=0:NElemeD-1;
    Dan(1,9*Modos+1+i)=Dano(i+1);
end

clear all
close all
clc
%% 5% FIN DEL PROGRAMA

```

BysMaestro.m

```
%% Definimos el tope máximo del rango de daño para calcular la
probabilidad de que se encuentre en ese rango cada elemento dañado

e=ones(7,10);
C=[0.7 0.75 0.80 0.85 0.9 0.95 1];
Linf=0.12;
Lsup=0.92;

for j=1:7;

Criterio=C(1,j);
BysFinal(Criterio);

e1=Bayese12345;
e2=Bayese678910;

for i=1:5
    e(j,i)=e1(:,i);
    e(j,i+5)=e2(:,i);
end
end

%% Agrupamos los rangos de permisibilidad de daño en función de la
probabilidad de que se encuentre dañado bajo del criterio

Ran=ones(2,10);

for i=1:10
    for j=1:7
        if e(j,i) <= Linf
            Ran(1,i) = C(1,j);
        elseif e(j,i) >= Lsup && Ran(2,i)==1
            Ran(2,i) = C(1,j);
        end
    end
end

%% Guardamos las matrices de probabilidades y Los rangos de daño para
cada elemento

save('e','e');
save('Ran','Ran');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%
```

```

function BysFinal(Criterio)
%% PROGRAMA PARA CREAR LAS TABLAS DE VERDAD A PARTIR DE LAS FRECUENCIAS Y
%%MODOS DE VIBRACIÓN%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%clear all
%close all
%clc

Mac=load('Bayperfect');

M=Mac;
[~,NCol]=size(Mac);
Modos=4;
M(:, [NCol-9 NCol-8 NCol-7 NCol-6 NCol-5 NCol-4 NCol-3 NCol-2 NCol-1
NCol])=[];

Ref=M(1, :);
M(1, :)=[];
Mac(1, :)=[];

[NDatos,~]=size(Mac);
%%%%%%%%%% 2.5 5 10
Dif1=5;
%%%%%%%%%% 0.04 0.08 0.16
Dif2=0.08;
%%%%%%%%%% 17.5 35 70
Dif3=35;
%%%%%%%%%% 0.04 0.08 0.16
Dif4=0.08;

Dif=[Dif1 Dif2 Dif3 Dif4];

Exc=ones (NDatos,Modos*9);

for i=1:NDatos;
    Exc(i, :)=M(i, :);
end

MDE=ones (NDatos,Modos);

for k=1:NDatos
    for i=0:Modos-1;
        S=0;
        for j=0:8;
            z=i*9+1;
            S=S+Exc(k, z+j);
        end
        MDE(k, i+1)=(S)^2;
    end
end
end

```



```

%% Agregamos la información de los valores máximos y mínimos del criterio
de separación del grupo

C=max(A1(:,1));
D=min(A1(:,1));

%% %%%%%%%%%%% FORMAMOS LAS TABLAS DE VERDAD DE DAÑO %%%%%%%%%%%
if k~=0
    [fil,col]=size(A1);
    F1=zeros(fil,col);
    F1(:,1)=A1(:,1);
    for i=1:fil
        for j=2:col
            if A1(i,j)<=Criterio
                F1(i,j)=1;
            end
        end
    end
end

%% %%%%%%%%%%%Oreganización de las tablas preverdad%%%%%%%%%%
%% %%%%%%%%%%%

[TabProb12345, TabProb678910]=Tabs55(F1);

    for i=1:32
        Evidencial2345(d,i)=TabProb12345(i,3);
        Evidencial2345(d,i+32)=TabProb12345(i,4);
    end
    Evidencial2345(d,65)=c;
    Evidencial2345(d,66)=C;
    Evidencial2345(d,67)=D;

    for i=1:32
        Evidencia678910(d,i)=TabProb678910(i,3);
        Evidencia678910(d,i+32)=TabProb678910(i,4);
    end
    Evidencia678910(d,65)=c;
    Evidencia678910(d,66)=C;
    Evidencia678910(d,67)=D;

    d=d+1;
end
A1(:,:)=[];
end
end
%%%%%%%%%%5%% FIN DEL PROGRAMA %%%%%%%%%%%

```

```
function [TabProb12345,TabProb678910]=Tabs55 (F1)
```

```
%%%%%%%%%%EVIDENCIAS ELEMENTOS 1 2 3 4 5%%%%%%%%%%
```

```
%DEFINIR CONTADORES
```

```
c=0;  
d=0;  
e=0;  
f=0;  
g=0;  
h=0;  
i=0;  
j=0;  
k=0;  
l=0;  
m=0;  
n=0;  
o=0;  
p=0;  
q=0;  
r=0;  
s=0;  
t=0;  
u=0;  
v=0;  
w=0;  
y=0;  
z=0;  
C=0;  
D=0;  
E=0;  
F=0;  
G=0;  
H=0;  
I=0;  
J=0;  
K=0;
```

```
[a,b]=size(F1);
```

```
%EVIDENCIAS ELEMENTOS 1 2 3 4 5
```

```
for x=1:a  
if F1(x,2)==1 && F1(x,3)==1 && F1(x,4)==1 && F1(x,5)==1 && F1(x,6)==1  
c=c+1;  
elseif F1(x,2)==1 && F1(x,3)==1 && F1(x,4)==1 && F1(x,5)==1 && F1(x,6)==0  
d=d+1;  
elseif F1(x,2)==1 && F1(x,3)==1 && F1(x,4)==1 && F1(x,5)==0 && F1(x,6)==1  
e=e+1;  
elseif F1(x,2)==1 && F1(x,3)==1 && F1(x,4)==1 && F1(x,5)==0 && F1(x,6)==0  
f=f+1;  
elseif F1(x,2)==1 && F1(x,3)==1 && F1(x,4)==0 && F1(x,5)==1 && F1(x,6)==1  
g=g+1;
```



```
end  
end
```

```
%ORGANIZACION ETIQUETAS
```

```
TabProb12345 (1,1)=5.111115;  
TabProb12345 (2,1)=5.111105;  
TabProb12345 (3,1)=5.111015;  
TabProb12345 (4,1)=5.111005;  
TabProb12345 (5,1)=5.110115;  
TabProb12345 (6,1)=5.110105;  
TabProb12345 (7,1)=5.110015;  
TabProb12345 (8,1)=5.110005;  
TabProb12345 (9,1)=5.101115;  
TabProb12345 (10,1)=5.101105;  
TabProb12345 (11,1)=5.101015;  
TabProb12345 (12,1)=5.101005;  
TabProb12345 (13,1)=5.100115;  
TabProb12345 (14,1)=5.100105;  
TabProb12345 (15,1)=5.100015;  
TabProb12345 (16,1)=5.100005;  
TabProb12345 (17,1)=5.011115;  
TabProb12345 (18,1)=5.011105;  
TabProb12345 (19,1)=5.011015;  
TabProb12345 (20,1)=5.011005;  
TabProb12345 (21,1)=5.010115;  
TabProb12345 (22,1)=5.010105;  
TabProb12345 (23,1)=5.010015;  
TabProb12345 (24,1)=5.010005;  
TabProb12345 (25,1)=5.001115;  
TabProb12345 (26,1)=5.001105;  
TabProb12345 (27,1)=5.001015;  
TabProb12345 (28,1)=5.001005;  
TabProb12345 (29,1)=5.000115;  
TabProb12345 (30,1)=5.000105;  
TabProb12345 (31,1)=5.000015;  
TabProb12345 (32,1)=5.000005;
```

```
%ORGANIZACION CONTADORES
```

```
TabProb12345 (1,2)=c;  
TabProb12345 (2,2)=d;  
TabProb12345 (3,2)=e;  
TabProb12345 (4,2)=f;  
TabProb12345 (5,2)=g;  
TabProb12345 (6,2)=h;  
TabProb12345 (7,2)=i;  
TabProb12345 (8,2)=j;  
TabProb12345 (9,2)=k;  
TabProb12345 (10,2)=l;  
TabProb12345 (11,2)=m;  
TabProb12345 (12,2)=n;  
TabProb12345 (13,2)=o;  
TabProb12345 (14,2)=p;  
TabProb12345 (15,2)=q;
```

```

TabProb12345 (16, 2) =r;
TabProb12345 (17, 2) =s;
TabProb12345 (18, 2) =t;
TabProb12345 (19, 2) =u;
TabProb12345 (20, 2) =v;
TabProb12345 (21, 2) =w;
TabProb12345 (22, 2) =y;
TabProb12345 (23, 2) =z;
TabProb12345 (24, 2) =C;
TabProb12345 (25, 2) =D;
TabProb12345 (26, 2) =E;
TabProb12345 (27, 2) =F;
TabProb12345 (28, 2) =G;
TabProb12345 (29, 2) =H;
TabProb12345 (30, 2) =I;
TabProb12345 (31, 2) =J;
TabProb12345 (32, 2) =K;

```

```

x=c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+y+z+C+D+E+F+G+H+I+J+K;

```

```

%PORCENTAJE

```

```

TabProb12345 (1, 3) =c/x;
TabProb12345 (2, 3) =d/x;
TabProb12345 (3, 3) =e/x;
TabProb12345 (4, 3) =f/x;
TabProb12345 (5, 3) =g/x;
TabProb12345 (6, 3) =h/x;
TabProb12345 (7, 3) =i/x;
TabProb12345 (8, 3) =j/x;
TabProb12345 (9, 3) =k/x;
TabProb12345 (10, 3) =l/x;
TabProb12345 (11, 3) =m/x;
TabProb12345 (12, 3) =n/x;
TabProb12345 (13, 3) =o/x;
TabProb12345 (14, 3) =p/x;
TabProb12345 (15, 3) =q/x;
TabProb12345 (16, 3) =r/x;
TabProb12345 (17, 3) =s/x;
TabProb12345 (18, 3) =t/x;
TabProb12345 (19, 3) =u/x;
TabProb12345 (20, 3) =v/x;
TabProb12345 (21, 3) =w/x;
TabProb12345 (22, 3) =y/x;
TabProb12345 (23, 3) =z/x;
TabProb12345 (24, 3) =C/x;
TabProb12345 (25, 3) =D/x;
TabProb12345 (26, 3) =E/x;
TabProb12345 (27, 3) =F/x;
TabProb12345 (28, 3) =G/x;
TabProb12345 (29, 3) =H/x;
TabProb12345 (30, 3) =I/x;
TabProb12345 (31, 3) =J/x;
TabProb12345 (32, 3) =K/x;

```

TabProb12345 (1, 4)=1-c/x;
TabProb12345 (2, 4)=1-d/x;
TabProb12345 (3, 4)=1-e/x;
TabProb12345 (4, 4)=1-f/x;
TabProb12345 (5, 4)=1-g/x;
TabProb12345 (6, 4)=1-h/x;
TabProb12345 (7, 4)=1-i/x;
TabProb12345 (8, 4)=1-j/x;
TabProb12345 (9, 4)=1-k/x;
TabProb12345 (10, 4)=1-l/x;
TabProb12345 (11, 4)=1-m/x;
TabProb12345 (12, 4)=1-n/x;
TabProb12345 (13, 4)=1-o/x;
TabProb12345 (14, 4)=1-p/x;
TabProb12345 (15, 4)=1-q/x;
TabProb12345 (16, 4)=1-r/x;
TabProb12345 (17, 4)=1-s/x;
TabProb12345 (18, 4)=1-t/x;
TabProb12345 (19, 4)=1-u/x;
TabProb12345 (20, 4)=1-v/x;
TabProb12345 (21, 4)=1-w/x;
TabProb12345 (22, 4)=1-y/x;
TabProb12345 (23, 4)=1-z/x;
TabProb12345 (24, 4)=1-C/x;
TabProb12345 (25, 4)=1-D/x;
TabProb12345 (26, 4)=1-E/x;
TabProb12345 (27, 4)=1-F/x;
TabProb12345 (28, 4)=1-G/x;
TabProb12345 (29, 4)=1-H/x;
TabProb12345 (30, 4)=1-I/x;
TabProb12345 (31, 4)=1-J/x;
TabProb12345 (32, 4)=1-K/x;

%%%5%#####EVIDENCIAS ELEMENTOS 6 7 8 9 10#####

c=0;
d=0;
e=0;
f=0;
g=0;
h=0;
i=0;
j=0;
k=0;
l=0;
m=0;
n=0;
o=0;
p=0;
q=0;
r=0;
s=0;
t=0;
u=0;

```

v=0;
w=0;
y=0;
z=0;
C=0;
D=0;
E=0;
F=0;
G=0;
H=0;
I=0;
J=0;
K=0;

```

```

%EVIDENCIAS ELEMENTOS 6 7 8 9 10

```

```

for x=1:a
if F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==1 && F1(x,11)==1
    c=c+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==0
    d=d+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==1
    e=e+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==0
    f=f+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==1
    g=g+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==0
    h=h+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==1
    i=i+1;
elseif F1(x,7)==1 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==0
    j=j+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==1
    k=k+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==0
    l=l+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==1
    m=m+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==0
    n=n+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==1
    o=o+1;

```

```

elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==0
    p=p+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==1
    q=q+1;
elseif F1(x,7)==1 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==0
    r=r+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==1
    s=s+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==0
    t=t+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==1
    u=u+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==0
    v=v+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==1
    w=w+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==0
    y=y+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==1
    z=z+1;
elseif F1(x,7)==0 && F1(x,8)==1 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==0
    C=C+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==1
    D=D+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==1 &&
F1(x,11)==0
    E=E+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==1
    F=F+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==1 && F1(x,10)==0 &&
F1(x,11)==0
    G=G+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==1
    H=H+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==1 &&
F1(x,11)==0
    I=I+1;
elseif F1(x,7)==0 && F1(x,8)==0 && F1(x,9)==0 && F1(x,10)==0 &&
F1(x,11)==1
    J=J+1;

```

```

else    F1(x,7)==0    &&    F1(x,8)==0    &&    F1(x,9)==0    &&    F1(x,10)==0    &&
F1(x,11)==0;
        K=K+1;
end
end

```

```

%ORGANIZACIÓN ETIQUETAS

```

```

TabProb678910(1,1)=5.111115;
TabProb678910(2,1)=5.111105;
TabProb678910(3,1)=5.111015;
TabProb678910(4,1)=5.111005;
TabProb678910(5,1)=5.110115;
TabProb678910(6,1)=5.110105;
TabProb678910(7,1)=5.110015;
TabProb678910(8,1)=5.110005;
TabProb678910(9,1)=5.101115;
TabProb678910(10,1)=5.101105;
TabProb678910(11,1)=5.101015;
TabProb678910(12,1)=5.101005;
TabProb678910(13,1)=5.100115;
TabProb678910(14,1)=5.100105;
TabProb678910(15,1)=5.100015;
TabProb678910(16,1)=5.100005;
TabProb678910(17,1)=5.011115;
TabProb678910(18,1)=5.011105;
TabProb678910(19,1)=5.011015;
TabProb678910(20,1)=5.011005;
TabProb678910(21,1)=5.010115;
TabProb678910(22,1)=5.010105;
TabProb678910(23,1)=5.010015;
TabProb678910(24,1)=5.010005;
TabProb678910(25,1)=5.001115;
TabProb678910(26,1)=5.001105;
TabProb678910(27,1)=5.001015;
TabProb678910(28,1)=5.001005;
TabProb678910(29,1)=5.000115;
TabProb678910(30,1)=5.000105;
TabProb678910(31,1)=5.000015;
TabProb678910(32,1)=5.000005;

```

```

%ORGANIZACIÓN CONTADORES

```

```

TabProb678910(1,2)=c;
TabProb678910(2,2)=d;
TabProb678910(3,2)=e;
TabProb678910(4,2)=f;
TabProb678910(5,2)=g;
TabProb678910(6,2)=h;
TabProb678910(7,2)=i;
TabProb678910(8,2)=j;
TabProb678910(9,2)=k;
TabProb678910(10,2)=l;
TabProb678910(11,2)=m;
TabProb678910(12,2)=n;

```

```

TabProb678910 (13, 2)=o;
TabProb678910 (14, 2)=p;
TabProb678910 (15, 2)=q;
TabProb678910 (16, 2)=r;
TabProb678910 (17, 2)=s;
TabProb678910 (18, 2)=t;
TabProb678910 (19, 2)=u;
TabProb678910 (20, 2)=v;
TabProb678910 (21, 2)=w;
TabProb678910 (22, 2)=y;
TabProb678910 (23, 2)=z;
TabProb678910 (24, 2)=C;
TabProb678910 (25, 2)=D;
TabProb678910 (26, 2)=E;
TabProb678910 (27, 2)=F;
TabProb678910 (28, 2)=G;
TabProb678910 (29, 2)=H;
TabProb678910 (30, 2)=I;
TabProb678910 (31, 2)=J;
TabProb678910 (32, 2)=K;

```

```
x=c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+y+z+C+D+E+F+G+H+I+J+K;
```

```
%PORCENTAJE
```

```

TabProb678910 (1, 3)=c/x;
TabProb678910 (2, 3)=d/x;
TabProb678910 (3, 3)=e/x;
TabProb678910 (4, 3)=f/x;
TabProb678910 (5, 3)=g/x;
TabProb678910 (6, 3)=h/x;
TabProb678910 (7, 3)=i/x;
TabProb678910 (8, 3)=j/x;
TabProb678910 (9, 3)=k/x;
TabProb678910 (10, 3)=l/x;
TabProb678910 (11, 3)=m/x;
TabProb678910 (12, 3)=n/x;
TabProb678910 (13, 3)=o/x;
TabProb678910 (14, 3)=p/x;
TabProb678910 (15, 3)=q/x;
TabProb678910 (16, 3)=r/x;
TabProb678910 (17, 3)=s/x;
TabProb678910 (18, 3)=t/x;
TabProb678910 (19, 3)=u/x;
TabProb678910 (20, 3)=v/x;
TabProb678910 (21, 3)=w/x;
TabProb678910 (22, 3)=y/x;
TabProb678910 (23, 3)=z/x;
TabProb678910 (24, 3)=C/x;
TabProb678910 (25, 3)=D/x;
TabProb678910 (26, 3)=E/x;
TabProb678910 (27, 3)=F/x;
TabProb678910 (28, 3)=G/x;
TabProb678910 (29, 3)=H/x;
TabProb678910 (30, 3)=I/x;

```

```
TabProb678910 (31, 3)=J/x;  
TabProb678910 (32, 3)=K/x;
```

```
TabProb678910 (1, 4)=1-c/x;  
TabProb678910 (2, 4)=1-d/x;  
TabProb678910 (3, 4)=1-e/x;  
TabProb678910 (4, 4)=1-f/x;  
TabProb678910 (5, 4)=1-g/x;  
TabProb678910 (6, 4)=1-h/x;  
TabProb678910 (7, 4)=1-i/x;  
TabProb678910 (8, 4)=1-j/x;  
TabProb678910 (9, 4)=1-k/x;  
TabProb678910 (10, 4)=1-l/x;  
TabProb678910 (11, 4)=1-m/x;  
TabProb678910 (12, 4)=1-n/x;  
TabProb678910 (13, 4)=1-o/x;  
TabProb678910 (14, 4)=1-p/x;  
TabProb678910 (15, 4)=1-q/x;  
TabProb678910 (16, 4)=1-r/x;  
TabProb678910 (17, 4)=1-s/x;  
TabProb678910 (18, 4)=1-t/x;  
TabProb678910 (19, 4)=1-u/x;  
TabProb678910 (20, 4)=1-v/x;  
TabProb678910 (21, 4)=1-w/x;  
TabProb678910 (22, 4)=1-y/x;  
TabProb678910 (23, 4)=1-z/x;  
TabProb678910 (24, 4)=1-C/x;  
TabProb678910 (25, 4)=1-D/x;  
TabProb678910 (26, 4)=1-E/x;  
TabProb678910 (27, 4)=1-F/x;  
TabProb678910 (28, 4)=1-G/x;  
TabProb678910 (29, 4)=1-H/x;  
TabProb678910 (30, 4)=1-I/x;  
TabProb678910 (31, 4)=1-J/x;  
TabProb678910 (32, 4)=1-K/x;
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function e=Bayese12345
```

```
%clear all  
%close all  
%clc
```

```
%%%%DATOS DE ENTRADA
```

```
[F_1,F_2,F_3,F_4]=BysInfo;
```

```
A1=load('Evidencial2345');  
A_1(:, :)=A1.Evidencial2345(:, :);
```

```

[NFila, NColu]=size(A_1);

x=0;
y=0;
w=0;
z=0;

for i=1:NFila
    if A_1(i,65)==1
        x=x+1;
        B_1(x,:)=A_1(i,:);

    elseif A_1(i,65)==2
        y=y+1;
        B_2(y,:)=A_1(i,:);

    elseif A_1(i,65)==3
        w=w+1;
        B_3(w,:)=A_1(i,:);

    elseif A_1(i,65)==4
        z=z+1;
        B_4(z,:)=A_1(i,:);

    else
    end
end

B_1(:,65)=[];
B_2(:,65)=[];
B_3(:,65)=[];
B_4(:,65)=[];

C_1(:,:)=A_1(:,:);
C_1(:,[65 66 67])=[];

N = NFila+5; %numero nodos
dag = zeros(N,N);

Vec1=6:1:NFila+5;

dag(1,Vec1) = 1; %establecer descendencia
dag(2,Vec1) = 1;
dag(3,Vec1) = 1;
dag(4,Vec1) = 1;
dag(5,Vec1) = 1;

discrete_nodes = 1:N;
node_sizes = 2*ones(1,N);

bnet = mk_bnet(dag, node_sizes, 'discrete', discrete_nodes);

G = bnet.dag;

```

```

draw_graph(G);

bnet.CPD{1} = tabular_CPD(bnet, 1, [0.5 0.5]); %tablas de verdad
bnet.CPD{2} = tabular_CPD(bnet, 2, [0.5 0.5]);
bnet.CPD{3} = tabular_CPD(bnet, 3, [0.5 0.5]);
bnet.CPD{4} = tabular_CPD(bnet, 4, [0.5 0.5]);
bnet.CPD{5} = tabular_CPD(bnet, 5, [0.5 0.5]);

for i=1:Nfila      %% Asignación de los nombres para cada hijo de la red
    bnet.CPD{i+5} = tabular_CPD(bnet, i+5, [C_1(i,:)]);
end

engine = jtree_inf_engine(bnet); %establecer evidencias 1 Verdad 2 Falso
evidence = cell(1,N);

for i=1:x-1 %Asignación del lugar del hijo de la Primera evidencia(1modo)
    if F_1>A_1(1,66)
        evidence{6} = 1;
    elseif F_1<A_1(x,66)
        evidence{x+5} = 1;
    elseif(F_1<=A_1(i,66) && F_1>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_1<A_1(i,67) && F_1>A_1(i+1,66))&& ((A_1(i,67)-F_1)<(F_1-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_1<A_1(i,67) && F_1>A_1(i+1,66))&& ((A_1(i,67)-F_1)>=(F_1-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

for i=x+1:x+y-1%Asignación del lugar del hijo de la 2da evidencia (2modo)
    if F_2>A_1(x+1,66)
        evidence{x+6} = 1;
    elseif F_2<A_1(x+y,66)
        evidence{x+y+5} = 1;
    elseif(F_2<=A_1(i,66) && F_2>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_2<A_1(i,67) && F_2>A_1(i+1,66))&& ((A_1(i,67)-F_2)<(F_2-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_2<A_1(i,67) && F_2>A_1(i+1,66))&& ((A_1(i,67)-F_2)>=(F_2-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

for i=x+y+1:w+x+y-1 %Asignación hijo de la 3ra evidencia(3modo)
    if F_3>A_1(x+y+1,66)
        evidence{x+y+6} = 1;
    elseif F_3<A_1(x+y+w,66)
        evidence{x+y+w+5} = 1;
    elseif(F_3<=A_1(i,66) && F_3>=A_1(i,67))

```

```

        evidence{i+5} = 1;
    elseif (F_3<A_1(i,67)  &&  F_3>A_1(i+1,66)) &&  ((A_1(i,67)-F_3)<(F_3-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_3<A_1(i,67)  &&  F_3>A_1(i+1,66)) &&  ((A_1(i,67)-F_3)>=(F_3-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

for i=w+x+y+1:z+w+x+y-1 %Asignación hijo de la 4ta evidencia(4modo)

    if F_4>A_1(x+y+w+1,66)
        evidence{x+y+w+6} = 1;
    elseif F_4<A_1(x+y+w+z,66)
        evidence{x+y+w+z+5} = 1;
    elseif(F_4<=A_1(i,66)  &&  F_4>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_4<A_1(i,67)  &&  F_4>A_1(i+1,66)) &&  ((A_1(i,67)-F_4)<(F_4-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_4<A_1(i,67)  &&  F_4>A_1(i+1,66)) &&  ((A_1(i,67)-F_4)>=(F_4-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

[engine, loglik] = enter_evidence(engine, evidence); %Probabilidad
marg = marginal_nodes(engine, 1);
e(1,1) = marg.T(1);
marg = marginal_nodes(engine, 2);
e(1,2) = marg.T(1);
marg = marginal_nodes(engine, 3);
e(1,3) = marg.T(1);
marg = marginal_nodes(engine, 4);
e(1,4) = marg.T(1);
marg = marginal_nodes(engine, 5);
e(1,5) = marg.T(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
function e=Bayese678910
```

```

%clear all
%close all
%clc

```

```
%%%%%DATOS DE ENTRADA
```

```
[F_1,F_2,F_3,F_4]=BysInfo;
```

```
A1=load('Evidencia678910');  
A_1(:,:)=A1.Evidencia678910(:,:);  
[Nfila,NColu]=size(A_1);
```

```
x=0;  
y=0;  
w=0;  
z=0;
```

```
for i=1:Nfila  
    if A_1(i,65)==1  
        x=x+1;  
        B_1(x,:)=A_1(i,:);  
  
    elseif A_1(i,65)==2  
        y=y+1;  
        B_2(y,:)=A_1(i,:);  
  
    elseif A_1(i,65)==3  
        w=w+1;  
        B_3(w,:)=A_1(i,:);  
  
    elseif A_1(i,65)==4  
        z=z+1;  
        B_4(z,:)=A_1(i,:);
```

```
    else  
    end
```

```
end
```

```
    B_1(:,65)=[];  
    B_2(:,65)=[];  
    B_3(:,65)=[];  
    B_4(:,65)=[];
```

```
C_1(:,:)=A_1(:,:);  
C_1(:,[65 66 67])=[];
```

```
N = Nfila+5; %numero nodos  
dag = zeros(N,N);
```

```
Vec1=6:1:Nfila+5;
```

```
dag(1,Vec1) = 1; %establecer descendencia  
dag(2,Vec1) = 1;  
dag(3,Vec1) = 1;  
dag(4,Vec1) = 1;  
dag(5,Vec1) = 1;
```

```

discrete_nodes = 1:N;
node_sizes = 2*ones(1,N);

bnet = mk_bnet(dag, node_sizes, 'discrete', discrete_nodes);

G = bnet.dag;
draw_graph(G);

bnet.CPD{1} = tabular_CPD(bnet, 1, [0.5 0.5]); %tablas de verdad
bnet.CPD{2} = tabular_CPD(bnet, 2, [0.5 0.5]);
bnet.CPD{3} = tabular_CPD(bnet, 3, [0.5 0.5]);
bnet.CPD{4} = tabular_CPD(bnet, 4, [0.5 0.5]);
bnet.CPD{5} = tabular_CPD(bnet, 5, [0.5 0.5]);

for i=1:Nfila           %% Asignación de los nombres para cada hijo de la red
    bnet.CPD{i+5} = tabular_CPD(bnet, i+5, [C_1(i,:)]);
end

engine = jtree_inf_engine(bnet); %establecer evidencias 1 Verdad 2 Falso
evidence = cell(1,N);

for i=1:x-1 %Asignación del lugar del hijo de la Primera evidencia(1modo)
    if F_1>A_1(1,66)
        evidence{6} = 1;
    elseif F_1<A_1(x,66)
        evidence{x+5} = 1;
    elseif(F_1<=A_1(i,66) && F_1>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_1<A_1(i,67) && F_1>A_1(i+1,66))&& ((A_1(i,67)-F_1)<(F_1-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_1<A_1(i,67) && F_1>A_1(i+1,66))&& ((A_1(i,67)-F_1)>=(F_1-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

for i=x+1:x+y-1%Asignación del lugar del hijo de la 2da evidencia (2modo)
    if F_2>A_1(x+1,66)
        evidence{x+6} = 1;
    elseif F_2<A_1(x+y,66)
        evidence{x+y+5} = 1;
    elseif(F_2<=A_1(i,66) && F_2>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_2<A_1(i,67) && F_2>A_1(i+1,66))&& ((A_1(i,67)-F_2)<(F_2-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_2<A_1(i,67) && F_2>A_1(i+1,66))&& ((A_1(i,67)-F_2)>=(F_2-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end
end

```

```

for i=x+y+1:w+x+y-1 %Asignación hijo de la 3ra evidencia(3modo)

    if F_3>A_1(x+y+1,66)
        evidence{x+y+6} = 1;
    elseif F_3<A_1(x+y+w,66)
        evidence{x+y+w+5} = 1;
    elseif(F_3<=A_1(i,66) && F_3>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_3<A_1(i,67) && F_3>A_1(i+1,66))&& ((A_1(i,67)-F_3)<(F_3-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_3<A_1(i,67) && F_3>A_1(i+1,66))&& ((A_1(i,67)-F_3)>=(F_3-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

for i=w+x+y+1:z+w+x+y-1 %Asignación hijo de la 4ta evidencia(4modo)
    if F_4>A_1(x+y+w+1,66)
        evidence{x+y+w+6} = 1;
    elseif F_4<A_1(x+y+w+z,66)
        evidence{x+y+w+z+5} = 1;
    elseif(F_4<=A_1(i,66) && F_4>=A_1(i,67))
        evidence{i+5} = 1;
    elseif (F_4<A_1(i,67) && F_4>A_1(i+1,66))&& ((A_1(i,67)-F_4)<(F_4-
A_1(i+1,66)))
        evidence{i+5} = 1;
    elseif (F_4<A_1(i,67) && F_4>A_1(i+1,66))&& ((A_1(i,67)-F_4)>=(F_4-
A_1(i+1,66)))
        evidence{i+6} = 1;
    end
end

[engine, loglik] = enter_evidence(engine, evidence); %Probabilidad
marg = marginal_nodes(engine, 1);
e(1,1) = marg.T(1);
marg = marginal_nodes(engine, 2);
e(1,2) = marg.T(1);
marg = marginal_nodes(engine, 3);
e(1,3) = marg.T(1);
marg = marginal_nodes(engine, 4);
e(1,4) = marg.T(1);
marg = marginal_nodes(engine, 5);
e(1,5) = marg.T(1);
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
function [F_1,F_2,F_3,F_4]=BysInfo

clear all
close all
clc

load('Wne');
load('EigenVece');
EigenVece=EigenVece';
Wx=ones(4,9);
Wf=ones(1,4);

for i=1:4;
    Wx(i,:)=Wne(1,i).*EigenVece(i,:);
    S=0;
    for j=1:9;
        S=S+Wx(i,j);
    end
    Wf(1,i)=(S)^2;
end

F_1=Wf(1,1);
F_2=Wf(1,2);
F_3=Wf(1,3);
F_4=Wf(1,4);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Segundo bloque de procesamiento (metodología SOM)

SetEntreViga.m

```
%Programa para construir la estructura de la data para el entrenamiento de la Red SOM viga
```

```
clear all
close all
clc
```

```
cd 'C:\Users\Usuario\Desktop\INGENIERIA\REDES SOM vs BAYES
NET\Trasteo1\EIGENS\10eigen'
load('Ran.mat');
load('Wne.mat');
load('EigenVece.mat');
load('Dano.mat');
```

```
cd 'C:\Users\Usuario\Desktop\INGENIERIA\REDES SOM vs BAYES
NET\Som_practica_Viga'
fid=fopen('DatViga.data','wt');
```

```
Modos=4;
NDatos=1000;
Ruido1=0.00;
Ruido2=0.00;
```

```
for k=1:NDatos
    j=1;
    D = DataViga1(Ran,Ruido1,Ruido2);
    for j=1:7*Modos;
        fprintf(fid,'%12.5g\t',D(1,j));
    end
    for j=7*Modos+1:7*Modos+10;
        fprintf(fid,'%8.2g\t',D(1,j));
    end
    fprintf(fid,'\n');
end
```

```
fclose(fid);
```

```
save ('EigenVece', 'EigenVece')
save ('Wne', 'Wne')
save ('Dano', 'Dano')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

function D = DataViga (ElemeIns,Ruido1,Ruido2)

cd      'C:\Users\Usuario\Desktop\INGENIERIA\REDES      SOM      vs      BAYES
NET\Som_practica_Viga'

%% Generación de datos (Daños aleatorios)

[~,NElemeIns]=size(ElemeIns);

N = 10; % Numero de elementos
Ei = 2.153E10;

Dani = ones(1,NElemeIns);

Elem=ones(1,N);
Dan = ones(1,N);

%% GENERACIÓN DE PORCENTAJES DE DAÑOS EN ELEMENTOS CON INCERTIDUMBRE DE
DAÑOS
con=1;
if NElemeIns ~= 0
    while con-1<NElemeIns && con-1 < N
        false = sum(ElemeIns(con) == ElemeIns(1:con));
        if false==1
            Dani(con) = random('unif',0.75,1,1);
            if Dani(con)<=0.05
                Dani(con) = 0.05;
            end
            con = con+1;
        else
            Elem(con) = [];
        end
    end
end

%% ARREGLO DEL VECTOR DE PORCENTAJES DE DAÑO
for j=1:N
    for i=1:NElemeIns
        if j == ElemeIns(i)
            Dan(j)=Dani(i);
        end
    end
end

%% ESCRITURA DE DATOS

E=Dan.*Ei;

[filas,nmat] = size(E);

```



```

%% APLICACIÓN DEL RUIDO A LA MEDIDA DE LOS MODOS

EigenVece=ones(Modos,12);
for i=1:Modos
EigenVece(i,:) = load(['EigenVector_',num2str(i),'.out'])+
load(['EigenVector_',num2str(i),'.out']).*ruido(i,2)';
end

EigenVece(:,[1 2 12]) = [];
[M,Elem]=size(EigenVece);

%% ORGANIZACIÓN Y PRESENTACIÓN DE DATOS

Nv1=zeros(Modos,Elem);

for i=1:Modos;
    Nv1(i,:)=Wne(1,i).*EigenVece(i,:);
end
MDE1=ones(1,Modos);
MDE=ones(1,Modos);

%% CALCULO DE LA MÍNIMA DISTANCIA EUCLIDIANA

for k=1:Modos;
    S=0;
    for j=1:9;
        S=S+(Nv1(k,j))^2;
    end
MDE(1,k)=sqrt(S);
end

%% CALCULO DEL SEGUNDO CRITERIO

load('VecRef');
for i=1:Modos
    if abs(Nv1(i,1))< abs(Nv1(i,1)-VecRef(i,1))
        Nv1(i,:)=Nv1(i, :)*(-1);
    end
end

k=1;
for i=1:Modos;
    S=0;
    for j=1:9;
        S=S+Nv1(i,j);
    end
MDE1(k)=(S)^2;
k=k+1;
end

```

```

%% ORGANIZACIÓN DE LA DATA
j=1;
D=zeros(1,Modos*7+N);
for n=0:Modos-1;
    z=n*7+1;
    for i=0:4;
        D(1,z+i)=Nv1(n+1,i+1);
        j=j+1;
    end

    D(1,z+5)=MDE(n+1);
    D(1,z+6)=MDE1(n+1);
    j=j+2;
end

%% ORGANIZACIÓN DE LAS ETIQUETAS
for i=0:N-1;
    D(1,j+i)=Dan(i+1);
end
j=j-1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%

```

SOMViga.m

```

clear all
close all
clc

echo off

D=load('DatViga.data');
NEleme=10;
EntradaData;
P=load('ValViga.data');
[NDatos,col]=size(D);

A=zeros(NDatos,col-10);

for i=1:col-10
    A(:,i)=D(:,i);
end

[a,b]=size(P);
V=ones(a,b-10);

for i=1:col-10
    V(:,i) = P(:,i);
end

```

```

A(NDatos+1,:) = V(1,:);

sData = som_data_struct(A, 'name', 'Frec & Modos (simulated)');

sData=som_label(sData, 'clear', 'all');

for i=1:NDatos;
    sData=som_label(sData, 'add', i, [num2str(D(i, col-9)), ' ', ...
        num2str(D(i, col-8)), ' ', num2str(D(i, col-7)), ' ', num2str(D(i, col-6)), '
', num2str(D(i, col-5)), ' ', num2str(D(i, col-4)), ' ', ...
        num2str(D(i, col-3)), ' ', num2str(D(i, col-2)), ' ', num2str(D(i, col-1)), '
', num2str(D(i, col))] );
end

EtVal=ones(1,NEleme);
j=1;

for i=col-9:col
    EtVal(:,j)=P(:,i);
    j=j+1;
end

%% DATA NORMALIZATION

sData = som_normalize(sData, 'var');
Vn=sData.data(NDatos+1,:);
sData.data(NDatos+1,:)=[];

%% MAP TRAINING

sMap = som_make(sData, 'init', 'randinit', 'algorithm', 'batch', 'msize',
[50 25], 'training', [7 25], 'radius', [10 3.5], 'trailen', 80,
'tracking', 0.9, 'neigh', 'gaussian');

%% HERENCIA DE ETIQUETAS:

sMap = som_autolabel(sMap, sData, 'vote');

%% CLASIFICACIÓN DE UN VECTOR

figure
[bm, q] = som_bmus(sMap, Vn, 'all'); % distance to all map units
[dummy, ind] = sort(bm(1,:)); d0 = q(1, ind)';
bmu=som_bmus(sMap, Vn);

h0 = zeros(prod(sMap.topol.msize), 1); h0(bmu) = 1; % crisp hits

lab = cell(prod(sMap.topol.msize), 1);
lab{bmu} = 'Val';

hf0 = som_hits(sMap, Vn, 'fuzzy'); % fuzzy response

```

```

som_show(sMap,'umat',{ 'all', 'BMU'},...
'color',{d0,'Qerror 0'}, 'color',{hf0,'Fuzzy response 0'},...
'empty','BMU+qerror');
som_show_add('label',lab,'Subplot',1,'Textcolor','r');
som_show_add('hit',h0,'Subplot',4,'MarkerColor','r');
hold on
Co = som_vis_coords(sMap.topol.lattice,sMap.topol.msize);
plot3(Co(bmu,[1 1]),Co(bmu,[2 2]),[0 10*q(1,1)/q(1,end)], 'r-')
view(3), axis equal

%% PRESENTACIÓN DE LA RESPUESTA

size(sMap.labels(bmu));
Resp2=sMap.labels(bmu);
Resp21=char(Resp2);
R2=str2num(Resp21);

if R2 ~= 0
    figure
    bar(R2,0.7), colormap(cool);
    Error=ones(1,NEleme);
    e=0;
    for i=1:NEleme
        Error(:,i)=abs(R2(:,i)-EtVal(:,i));
        if Error(:,i) <= 0.055
            e=e+1;
        end
    end
    disp(' La aproximación mas cercana es:');
    disp(Resp2);
else
    disp(' No se encuentra en la base de datos una aproximación
pertinente');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function EntradaData

clear all
close all
clc

N=10;
Ei = 2.153E10;
load('Wne');
load('EigenVece');
EigenVece=EigenVece';

[Modos,Nodos]=size(EigenVece);

```

```

%% ORGANIZACIÓN DEL VECTOR DE LA DATA

Nv1=zeros(Modos,Nodos);
for i=1:Modos;
    Nv1(i,:)=Wne(1,i).*EigenVece(i,:);
end

MDE1=ones(1,4);
MDE=ones(1,4);

%% CALCULO DE LA MÍNIMA DISTANCIA EUCLIDIANA

for k=1:Modos;
    S=0;
    for j=1:9;
        S=S+(Nv1(k,j))^2;
    end
MDE(1,k)=sqrt(S);
end

%% CALCULO DEL MAC

RefViga(N,Ei);
load('VecRef');

for i=1:Modos
    if abs(Nv1(i,1)) < abs(Nv1(i,1)-VecRef(i,1))
        Nv1(i,:)=Nv1(i, :)*(-1);
    end
end

k=1;
for i=1:Modos;
    S=0;
    for j=1:9;
        S=S+Nv1(i,j);
    end
MDE1(k)=(S)^2;
k=k+1;
end

%% GUARDAR EL VECTOR FORMADO EN UN .DATA
j=1;
D=ones(1,Modos*7+10);
for n=0:Modos-1;
    z=n*7+1;
    for i=0:4;
        D(1,z+i)=Nv1(n+1,i+1);
    end
    j=j+1;
end

D(1,z+5)=MDE(n+1);

```

```

        D(1,z+6)=MDE1(n+1);
        j=j+2;
end

fid=fopen('ValViga.data','wt');
j=1;
    for j=1:7*Modos;
        fprintf(fid,'%8.5g\t',D(1,j));
    end
    for j=7*Modos+1:7*Modos+10
        fprintf(fid,'%8.2g\t',D(1,j));
    end

fclose(fid);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Validacion.m

```

Resp=zeros(6,10);

C=[80 80 40 40 25 25; 11 11 22 22 35 35];
for k=1:6

D=load('DatViga.data');
NEleme=10;

P=load('ValViga.data');
[NDatos,col]=size(D);
A=zeros(NDatos,col-10);

for i=1:col-10
    A(:,i)=D(:,i);
end

[a,b]=size(P);
V=ones(a,b-10);

for i=1:col-10
    V(:,i) = P(:,i);
end

A(NDatos+1,:)=V(1,:);

sData = som_data_struct(A,'name','Frec & Modos (simulated)');

sData=som_label(sData,'clear','all');

```

```

for i=1:NDatos;
    sData=som_label(sData,'add',i,[num2str(D(i,col-9)),' ',...
        num2str(D(i,col-8)),' ',num2str(D(i,col-7)),' ',num2str(D(i,col-6)),'
',num2str(D(i,col-5)),' ',num2str(D(i,col-4)),' ',...
        num2str(D(i,col-3)),' ',num2str(D(i,col-2)),' ',num2str(D(i,col-1)),'
',num2str(D(i,col))]);
end

EtVal=ones(1,NEleme);
j=1;

for i=col-9:col
    EtVal(:,j)=P(:,i);
    j=j+1;
end

%% DATA NORMALIZATION

sData = som_normalize(sData,'var');
Vn=sData.data(NDatos+1,:);
sData.data(NDatos+1,:)=[];

%% MAP TRAINING

sMap = som_make(sData, 'init', 'randinit', 'algorithm','batch','msize',
[C(1,k) C(2,k)], 'training', [7 25], 'radius', [10 3.5], 'trailen', 80,
'tracking', 0.9, 'neigh', 'gaussian');

%% HERENCIA DE ETIQUETAS:

sMap = som_autolabel(sMap,sData,'add');

%% CLASIFICACIÓN DE UN VECTOR

bmu=som_bmus(sMap,Vn);

%% PRESENTACION DE LA RESPUESTA

size(sMap.labels(bmu));
Resp2=sMap.labels(bmu);
Resp21=char(Resp2);
R2=str2num(Resp21);

if R2 ~= 0
    Resp(k,:)=R2;
else
    disp(' No se encuentra en la base de datos una aproximación
pertinente');
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FIN DEL PROGRAMA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```