

Faults Diagnostic using Hopfield Artificial Neural Network in front of Incomplete Data

Raquelita Torres Cabeza¹, Orestes Llanes Santiago², Egly Barrero Viciado³ and Valery Moreno Vega⁴

^{1,2,3,4} Ciudad Universitaria José Antonio Echeverría (CUJAE). Havana, Cuba.

Email: rtorresc@electrica.cujae.edu.cu, orestes@electrica.cujae.edu.cu, egly@electrica.cujae.edu.cu, valery@electrica.cujae.edu.cu

Received: December 01th, 2017.

ABSTRACT

Accepted: February 01th, 2018.

In this work, Hopfield Artificial Neural Network's performance in faults diagnostic in industrial process is evaluated when there is missing data. The diagnostic of two classes with different levels of overlapping data is done. As main result, Hopfield has a good performance in the implemented tests getting over architectures like the Probabilistic Neural Network, that's why it is a good option to use it in faults diagnostic.

Published: March 30th, 2018.

Copyright ©2016 by authors and Institute of Technology Galileo of Amazon (ITEGAM).

This work is licensed under the Creative Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Keywords: Incomplete data, Faults diagnostic, Hopfield artificial neural network.

Diagnóstico de fallos empleando una red neuronal artificial Hopfield ante datos incompletos

RESUMO

En este trabajo, el rendimiento de la Red Neuronal Artificial de Hopfield en el diagnóstico de fallas en procesos industriales se evalúa cuando faltan datos. Se realiza el diagnóstico de dos clases con diferentes niveles de datos superpuestos. Como resultado principal, Hopfield tiene un buen desempeño en las pruebas implementadas superando arquitecturas como la Red Neuronal Probabilistic, por eso es una buena opción usarlo en el diagnóstico de fallas.

Palavras Chaves: Datos incompletos, diagnóstico de fallas, red neuronal artificial Hopfield.

I. INTRODUCCIÓN

En la actualidad, el diagnóstico de fallos ha adquirido gran importancia en procesos de ingeniería debido a las ventajas potenciales que pueden obtenerse en la reducción de costos de mantenimiento y reparación, el mejoramiento de la productividad y el aumento de la seguridad y disponibilidad de los procesos industriales [16]. Una rápida detección e identificación de los fallos que afectan a un proceso puede ayudar a tomar decisiones correctas y reducir el daño que estos pueden ocasionar al sistema, por lo que las técnicas de diagnóstico permiten mejorar no sólo la eficiencia del proceso, sino además, la fiabilidad de los sistemas [11].

Frecuentemente en los procesos industriales cuando un SCADA adquiere los datos en tiempo real provenientes de cualquier subproceso, puede suceder que haya pérdida de información, o sea, los datos característicos de estos procesos se encuentran incompletos debido a problemas en el canal de medición, perturbaciones externas y otras razones [1][2][4][8]

[15][18]. Los métodos de diagnóstico de fallos basados en datos históricos se ven afectados por esta problemática que a menudo se presenta en las industrias, de ahí la necesidad que el diagnóstico sea robusto, evitando así falsas alarmas y clasificaciones erróneas del estado del sistema que conllevan la pérdida de confiabilidad en el sistema de diagnóstico [11-13].

Las redes neuronales artificiales (RNA) representan una de las herramientas más utilizadas dentro de las técnicas basadas en datos históricos, debido a sus resultados en la clasificación correcta de patrones de fallos provenientes de los procesos industriales [13] y a la robustez que presentan respecto a datos incompletos [11]. Ellas proveen una interesante y útil alternativa para los métodos clásicos porque son capaces de manejar situaciones complejas que presentan gran dificultad para los algoritmos determinísticos. Además, son capaces de operar con problemas no lineales y una vez entrenadas pueden realizar predicciones y generalizaciones con una gran confiabilidad [11].

Dentro de los diferentes modelos de redes neuronales se encuentra la RNA Hopfield desarrollada por John Hopfield en el

año 1982, basándose en los modelos de redes de McCulloch y Pitts y la Red Neuronal Probabilística desarrollada por Donald F. Specht en 1989, que se basa en la regla de clasificación Bayesiana aplicando el método de Parzen con Kernel Gaussiano [5], [10]. Una de las aplicaciones principales de estas redes está en el reconocimiento y clasificación de patrones, por lo que pueden ser empleadas en el diagnóstico de fallos [14].

La red Hopfield constituye una interesante arquitectura a utilizar para el reconocimiento y clasificación de patrones en el diagnóstico de fallos de un sistema por su rápida capacidad computacional en virtud de la naturaleza altamente paralela del proceso de convergencia y sus ventajas referentes a su funcionamiento como memoria asociativa.

En este trabajo se diseña un sistema de diagnóstico de fallos basado en estos modelos de redes neuronales artificiales para evaluar el desempeño de estas arquitecturas ante datos incompletos cuando son entrenadas con diferentes niveles de pérdida de información de los datos en diferentes situaciones, con el objetivo de lograr un sistema de diagnóstico más robusto y confiable.

El trabajo se organiza como sigue, en la sección II se presenta la RNA Hopfield, la Red Neuronal Probabilística y los aspectos necesarios para sus correspondientes diseños entrenamientos. Posteriormente en la sección III se describen las diferentes situaciones en las que se encuentran los datos a emplear y el diseño de las arquitecturas. La sección IV está dedicada al diseño de experimentos. En la sección V se discuten los resultados obtenidos. Finalmente, se presentan las conclusiones y recomendaciones para futuras investigaciones.

II. MODELOS DE REDES NEURONALES ARTIFICIALES

Las RNA constituyen un sistema de procesadores elementales interconectados, que realizan funciones tales como: aprendizaje, memorización, generalización o abstracción de características esenciales, representando de esta forma una estructura distribuida, de procesamiento paralelo [9].

En las redes neuronales se define el aprendizaje como el proceso mediante el cual se produce el ajuste de los parámetros libres de la red a partir de estimular el entorno que rodea a la misma. En la mayoría de las ocasiones el aprendizaje consiste simplemente en determinar un conjunto de pesos sinápticos que permita a la red realizar correctamente el tipo de procesamiento deseado.

Existen dos tipos básicos de aprendizaje: el supervisado y el no supervisado. Ambas modalidades pretenden estimar funciones entrada-salida multivariables o densidades de probabilidad, pero mientras que en el aprendizaje supervisado se proporciona cierta información sobre estas funciones (como la distribución de las clases, etiquetas de los patrones de entrada o salidas asociadas a cada patrón), en el no supervisado no se proporciona información alguna. Las reglas de aprendizaje supervisadas suelen ser computacionalmente más complejas, pero también más exactos sus resultados [7][9].

II.1 RED NEURONAL HOPFIELD

Uno de los modelos de redes realimentadas más populares es el modelo Hopfield, que cuenta con el interés añadido de que su análisis y aplicación fue una de las causas del renacimiento de los sistemas neuronales artificiales a principios de los años ochenta. La RNA Hopfield se divide en dos modelos: el modelo discreto y el modelo continuo [3][6][7][9][17][20].

En el caso del modelo discreto se aprovechan las ventajas que ofrece el funcionamiento de la red como memoria asociativa para el reconocimiento y clasificación de patrones. Este modelo resuelve la limitante que tienen otras arquitecturas como Perceptrón Multicapa y Redes de Base Radial. Estas requieren de un número elevado de muestras de entrenamiento para la correcta clasificación, mientras que Hopfield no necesita de un gran número de muestras para el entrenamiento, debido a su funcionamiento como memoria asociativa [20].

La arquitectura consiste en una red monocapa, ver Figura 1, donde cada neurona se conecta con todas las demás.

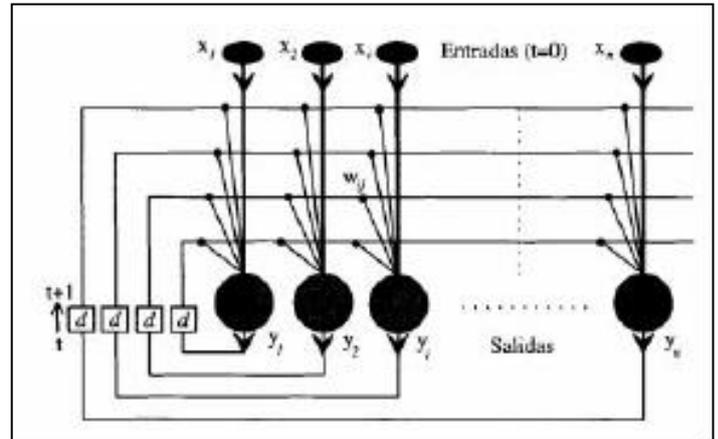


Figura 1: Arquitectura Hopfield.

Fuente: Los Autores, (2018).

La ecuación que caracteriza a la red es (1) [9].

$$y_i(t) = f(h_i(t)) = f\left(\sum_j W_{ij} X_j(t) - \theta_i\right) \quad (1)$$

donde:

W_{ij} : peso sináptico entre la neurona i y la neurona j ;

X_j : entrada j de la red;

θ_i : umbral de disparo correspondiente a la neurona i ;

h_i : campo local correspondiente a la neurona i ;

$f(\cdot)$: función de activación (generalmente función signo);

y_i : salida i de la red.

Esta red presenta un aprendizaje no supervisado, de tipo Hebbiano. La regla de aprendizaje que se define como regla de Hebb debe encontrar el conjunto de pesos sinápticos W que hacen que la red pueda almacenar como estados estables un conjunto de datos dado [6][9].

El modelo Hopfield opera como una memoria asociativa o denominada también memoria direccionable por contenido (Content Addressable Memory (CAM)) pues está ideada para asociar un patrón de entrada consigo mismo ya que si se presenta en la entrada un patrón almacenado en la memoria, la red ofrece como salida el propio patrón; de lo contrario, si no está almacenado, la red evoluciona hasta generar el patrón más parecido, lo cual permite separar y clasificar en grupos los diferentes estados del sistema [9].

En general, una red Hopfield, en cada iteración $iter$, pasa de un estado $x(iter)$ a otro estado $x(iter + 1)$. El proceso finaliza cuando se alcanza un estado estable o punto fijo de la red x^* , es decir, un estado que cumpla la siguiente condición a partir de un cierto $iter$

$$x(iter + 1) = x(iter) \equiv x^* \quad (2)$$

pues ello supondrá que la salida de la red ya no cambia, es decir, que la red se ha estabilizado. En ese momento se puede suponer que la red neuronal ha acabado de procesar el patrón original procedente del exterior $x(0)$, siendo x^* la respuesta final que proporciona [9].

En la fase de ejecución existen diferentes formas de actualizar el estado de una neurona, también conocidas como dinámicas de las neuronas de la red. Una primera forma es la dinámica asíncrona o modo serie, que consiste en que en un instante de tiempo solamente una neurona actualiza su estado y la otra forma es la dinámica síncrona o modo paralelo en la cual todas las neuronas de la red actualizan su estado en el instante *iter* [9].

Se determinó utilizar la dinámica de activación asíncrona por las características que esta ofrece [6][9], por lo que en cada instante de tiempo se activa sólo una neurona y el orden seleccionado para la actualización de las neuronas es 1, 2, ...,N, donde N es la cantidad de neuronas de la red.

La función de activación empleada en esta arquitectura es la función signo [9].

II.2 RED NEURONAL PROBABILÍSTICA

La Red Neuronal Probabilística (Probabilistic Neural Network (PNN)) es una RNA semi-supervisada, perteneciente a la familia de Redes Neuronales de Base Radial, la cual opera basada en el clasificador de Parzen y su aplicación a la estadística bayesiana.

Este modelo de red neuronal, debido a su funcionamiento como clasificador, se aplica en muchas ocasiones en el reconocimiento y clasificación de patrones, obteniendo buenos resultados [10], por lo cual puede ser aplicada en el diagnóstico de fallos.

La arquitectura consta de cuatro capas, ver Figura 2, una capa de entrada, la cual consiste en d neuronas (dimensión de los datos), una capa de datos, la cual consiste en N neuronas, una por cada vector representativo de las clases, una capa sumatoria de k neuronas, donde k es el número de clases y una capa de decisión, la cual consiste en una neurona [5][10].

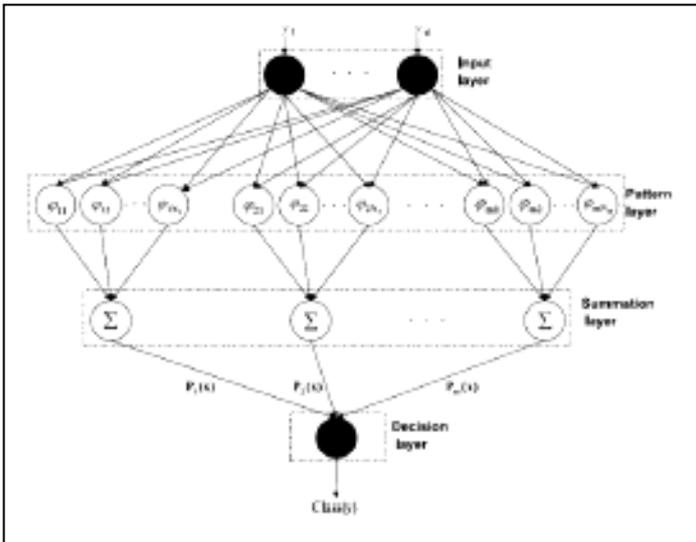


Figura 2: Arquitectura de la Red Neuronal Probabilística.

Fuente: Los Autores, (2018).

A diferencia del proceso de aprendizaje que se lleva a cabo en la mayoría de las Redes Neuronales Artificiales, en el cual se

realiza un ajuste de los parámetros: pesos y bias, en el caso de PNN no es necesario realizar ningún ajuste de pesos.

En este tipo de red sólo se determina a los patrones de salida mediante la comparación y el cálculo de las distancias de cada uno de los vectores de datos de entrada con cada uno de los datos representativos de cada una de las clases existentes. El algoritmo de entrenamiento consiste en adicionar una neurona en la capa de datos por cada dato de entrenamiento, adaptando sus pesos de entrada para que coincidan ambos, es decir, que los datos característicos de cada clase se correspondan con los datos de entrenamiento [5].

Cuando a la red se le presenta un vector a ser clasificado, la segunda capa se ocupa de calcular las distancias desde el vector de entrada hasta cada uno de los datos que caracterizan a las clases a través de la función φ_{ij} la cual funciona de acuerdo al kernel normal estándar utilizado como la función de densidad de probabilidad como se muestra en (3).

$$\varphi_{ij}(y) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} \exp \left[-\frac{(y - X_{ij})^T (y - X_{ij})}{2\sigma^2} \right] \quad (3)$$

donde σ es el parámetro de dispersión, el cual toma valor entre 0 y 1 que debe ser definido por el diseñador y X_{ij} es el dato j -ésimo característico de la clase i .

En la tercera capa el término $P_i(y)$ indica la probabilidad condicional o verosimilitud de que el dato de entrada pertenezca a la i -ésima clase y es obtenido mediante (4).

$$P_i(y) = \frac{1}{N_i} \sum_{j=1}^{N_i} \varphi_{ij}(y) \quad (4)$$

donde N_i constituye el total de datos de la clase i .

En la cuarta capa, que se corresponde con la capa de salida, el vector de entrada será asignado a la clase con mayor verosimilitud, cumpliendo con (5).

$$Clase(y) = \arg \max_i (P_i(y)) \quad (5)$$

donde $Clase(y)$ es la clase a la que el dato de entrada pertenece [5][10].

III. DESCRIPCIÓN DE LOS DATOS Y DISEÑO DE LAS ARQUITECTURAS

Los datos empleados en este trabajo se generaron a partir de los datos de Fisher que se encuentran en las librerías de Matlab y para el diseño de las arquitecturas se utilizó este mismo asistente matemático y los aspectos descritos en la sección anterior.

III.1 DESCRIPCIÓN DE LOS DATOS

Se emplearon dos conjuntos de datos en determinadas situaciones, con los objetivos de comprobar el correcto diseño de ambas arquitecturas, además, evaluar el comportamiento de las mismas ante diferentes situaciones.

El primer conjunto representa cuando las clases se encuentran totalmente separadas, ver Figura 3 y el segundo cuando están mezcladas, ver Figura 4.

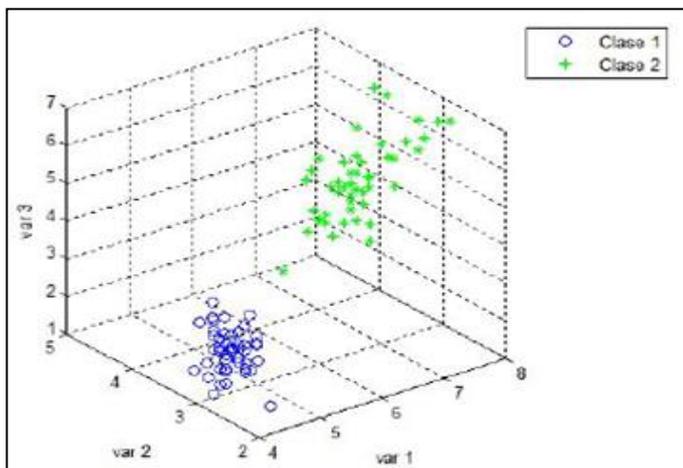


Figura 3: Clases totalmente separadas.

Fuente: Los Autores, (2018).

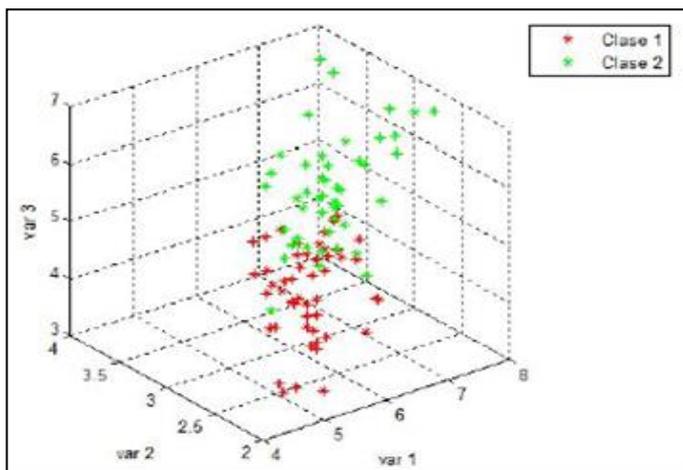


Figura 4: Clases mezcladas.

Fuente: Los Autores, (2018).

En cada caso, hay tres variables, dos clases y cada una de ellas tiene 50 datos, para un total de 100 muestras por cada conjunto de datos.

III.2 DISEÑO Y CONFIGURACIÓN DE HOPFIELD

Generalmente en el empleo de las RNA en el diagnóstico de fallos, el número de neuronas en la capa de entrada coincide con la cantidad de variables que se midan en el proceso, en este caso Hopfield es una arquitectura monocapa y tendrá tres neuronas en la capa de entrada pues los conjuntos de datos presentan tres componentes principales. En su única capa es donde se realiza el procesamiento y a la salida se brinda el dato recuperado que tiene la misma dimensión que el vector de entrada por lo que esta capa también está compuesta por tres neuronas.

Debido a que la red solamente proporciona un conjunto de datos a su salida, se necesita un mecanismo de clasificación posterior que permita agrupar los datos en las clases a las que pertenecen; para llevar a cabo este mecanismo, una vez que se obtuvieron todas las salidas de la red, se implementó el siguiente algoritmo:

- 1) Se determina la media correspondiente a los datos que representan a cada clase.
- 2) Se halla la distancia Euclídea de cada dato de salida (ofrecido por la red) con ese vector de media.
- 3) Se determina la menor distancia.

- 4) Se clasifica el dato en la clase donde se obtuvo la mínima distancia.

Se utilizó la distancia Euclídea por su sencillez y por ser la distancia más empleada en las redes neuronales artificiales [9].

III.3 DISEÑO Y CONFIGURACIÓN DE PNN

Se determinó utilizar para el cálculo de la densidad de probabilidad, el kernel Gaussiano, que constituye el kernel normal estándar más utilizado [5][10].

El parámetro de dispersión toma valor entre 0 y 1, por lo que se seleccionó $\sigma = 0.5$

Como el número de neuronas en la capa de entrada coincide con la cantidad de variables que se midan en el proceso, al igual que Hopfield, PNN tiene tres neuronas en la capa de entrada, en la capa de datos tiene 100 neuronas, debido a que son 2 clases, constituidas por 50 datos, la capa sumatoria presenta dos neuronas porque son dos clases las que se diagnostican y la capa de decisión está constituida por una sola neurona que brinda la clase a la que pertenece el dato de entrada.

IV. DISEÑO DE EXPERIMENTOS

Se diseñaron dos tipos de experimentos:

- Entrenando las redes con datos diferentes a los de la simulación.
- Entrenando las redes con datos incompletos y simulando con el conjunto original.

El diseño de estos experimentos tiene como objetivo, evaluar la capacidad de generalización de las redes ante la presencia de nuevos datos.

Para la realización del primer experimento se entrenan las arquitecturas empleando el método de validación cruzada, dividiendo cada subconjunto en 10 particiones [19].

Para realizar el segundo experimento se entrenan las arquitecturas con el conjunto original de datos pero introduciendo varios porcentajes (10%, 20% y 30%) de pérdida de información en determinadas variables y determinados tiempos de muestreo y se simulan las redes con todo el conjunto de datos (100 muestras).

En [1][8][18] reflejan diferentes métodos para estimar los datos incompletos debido a la pérdida de los mismos como son: (1) reemplazar los valores perdidos por cero, método no recomendable pues el cero indica la ausencia de características esenciales en determinados procesos y en este caso puede representar un fallo; (2) reemplazarlos por la media de la variable que se encuentra incompleta del conjunto de datos original, este método es el que comúnmente se utiliza por los resultados que brinda y el empleado en este trabajo; (3) eliminar los datos donde al menos exista un valor incompleto de algunas de las variables, en este caso se trabajará con menos datos de los disponibles y (4) desarrollar un modelo que reconstruya los valores perdidos atendiendo a la relación entre las variables.

V. RESULTADOS Y DISCUSIÓN

Los resultados obtenidos aplicando el método de validación cruzada se muestran en la Tabla 1, donde para cada uno de los conjuntos de datos, se obtiene el porcentaje de error de clasificación de las arquitecturas empleadas.

Tabla 1: % de error de clasificación n con entrenamiento con validación Cruzada.

Arquitecturas	Conjunto 1	Conjunto 2
Hopfield	0%	10%
PNN	0%	18%

Fuente: Los Autores, (2018).

Los resultados para el experimento con datos incompletos se muestran en las Tablas 2 para cuando las clases están separadas y en III cuando están mezcladas, donde para cada uno de estos casos, se obtiene el porcentaje de error de clasificación de las arquitecturas ante el 10%, 20% y 30% de pérdida de información.

Para el conjunto 1:

Tabla 2: % de error de clasificación n con diferentes % de datos Incompletos con clases separadas.

Arquitecturas	10%	20%	30%
Hopfield	0%	0%	0%
PNN	0%	0%	0%

Fuente: Los Autores, (2018).

Tabla 3: % de error de clasificación n con diferentes % de datos incompletos con clases mezcladas.

Arquitecturas	10%	20%	30%
Hopfield	16%	16%	16%
PNN	19%	20%	19%

Fuente: Los Autores, (2018).

En los resultados obtenidos se muestra la capacidad de generalización de las arquitecturas de redes neuronales ante datos nuevos y su desempeño ante la presencia de varios porcentajes de datos incompletos en el entrenamiento. En el caso de los conjuntos de datos donde las clases están separadas, el porcentaje de error de clasificación es cero como era de esperarse, por lo que se considera que el diseño de las redes es correcto. Se puede observar que en todos los casos el porcentaje de error de Hopfield es menor o igual que el de PNN, teniendo mejores resultados en la clasificación de las clases, por lo que Hopfield representa una arquitectura eficiente con resultados satisfactorios en el reconocimiento y clasificación de patrones y se considera adecuada para emplearla en el diagnóstico de fallos en procesos industriales.

VI. CONCLUSIONES

Teniendo en cuenta los resultados obtenidos en cada experimento, el desempeño de Hopfield ante las diferentes situaciones que se presentaron es mejor que el de la Red Neuronal Probabilística, presentando buena robustez ante datos incompletos, siendo esta una característica deseada en un sistema de diagnóstico de fallos. Por lo cual la red neuronal artificial Hopfield puede ser aplicada con éxito al diagnóstico de fallos.

Para futuras investigaciones se desea utilizar otras distancias además de la Euclídea en el mecanismo de clasificación de Hopfield con el objetivo de comprobar si la distancia empleada afecta o no la clasificación, utilizar otros métodos para estimar datos incompletos y continuar comparando con otras arquitecturas.

VII. REFERENCIAS

- [1] Baraldi, P.; Di Maio, F; Genini, D. and Zio, E., "Reconstruction of missing data in multidimensional time series by fuzzy similarity," Applied Soft Computing, vol. 26, pp. 1-9, 2015.
- [2] Chen, Haibo; Grant-Muller, Susan; Mussone, Lorenzo and Montgomery, Frank, "A study of Hibrid Neural Network Approaches and the Effects of Missing Data on Traffic Forecasting," Neural Computing & Applications, vol. 10, pp. 277-286, 2001.
- [3] Elhadef, Mourad, "A Modified Hopfield Neural Network for Diagnosing Comparison Based Multiprocessor Systems Using Partial Syndromes," 17th International Conference on Parallel and Distributed Systems, vol. 11, Abu Dhabi, Emiratos Arabes Unidos, 2011.
- [4] He, Xiao; Wang, Zidong and Zhou, D.H, "Robust fault detection for networked systems with communication delay and data missing," Automatica, vol. 45, pp. 2634-2639, 2009.
- [5] Khashei, Mehdi and Bijari, Mehdi, "Hybridization of the probabilistic neural networks with feed-forward neural networks for forecasting," Engineering Applications of Artificial Intelligence, vol. 25, pp. 1277- 1288, 2012.
- [6] Kumar, Somesh and Pratap Singh, Manu, "Study of Hopfield neural network with sub-optimal and random GA for pattern recalling of English characters," Applied Soft Computing, vol. 12, pp. 2593-2600, 2012.
- [7] Liukkonen, Mika; Havia, Elina and Hiltunen, Yrjo, "Computational intelligence in mass soldering of electronics - A survey," Expert Systems with Applications, vol. 39, pp. 9928-9937, 2012.
- [8] Markey, Mia K.; Tourassi, Georgia D.; Margolis, Michael and DeLong, David M., "Impact of missing data in evaluating artificial neural networks trained on complete data," Computers in Biology and Medicine, vol. 36, pp. 516-525, 2006.
- [9] Martin del Brio, Bonifacio and Sanz Molina, Alfredo, "Redes Neuronales y Sistemas Difusos," RA-MA, 2da edicion, 2001.
- [10] Miguez, Roberto; Georgiopoulos, Michael and Kaylani, Assem, "GPNN: A genetically engineered probabilistic neural network," Nonlinear Analysis, vol. 73, pp. 1783-1791, 2010.
- [11] Patan, Krzysztof, "Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes," 2008.
- [12] Patan, Krzysztof; Witczak, Marcin and Korbicz, Josef, "Towards Robustness in Neural Network Based Fault Diagnosis," Int. J. Appl. Math. Comput. Sci, vol. 18, no. 4, pp. 443-454, 2008.
- [13] Simani, Silvio; Fantuzzi, Cesare and Patton, Ron J., "Model Based Fault Diagnosis in Dynamic Systems using Identification Techniques," 2002.

- [14] Tatem, A.J.; Lewis, H.G.; Atkinson, P.M. and Nixon, M.S., “**Superresolution land cover pattern prediction using a Hopfield neural network**,” Remote Sensing of Environment, vol. 79, pp. 1-14, 2002.
- [15] van Lint, J.W.C.; Hoogendoorn, S.P. and van Zuylen, H.J., “**Accurate freeway travel time prediction with state-space neural networks under missing data**,” Transportation Research Part C, vol. 13, pp. 347-369, 2005.
- [16] Venkatasubramanian, Venkat; Rengaswamy, Raghunathan; Yin, Kewen and Kavuri, Surya N., “**A review of process fault detection and diagnosis**,” Computers and Chemical Engineering, 2003.
- [17] Wen, Ue-Pyng; Lan, Kuen-Ming and Shih, Hsu-Shih, “**A review of Hopfield neural networks for solving mathematical programming problems**,” European Journal of Operational Research, vol. 198, pp. 675-687, 2009.
- [18] Wu, Xuedong; Wang, Yaonan; Mao, Jianxu; Du, Zhaoping and Li, Chunhua, “**Multi-step prediction of time series with random missing data**,” Applied Mathematical Modelling, 2014.
- [19] Yaghini, Masoud; Khoshraftar, Mohammad M. and Fallahi, Mehdi, “**A hibrid algorithm for artificial neural network training**,” Engineering Applications of Artificial Intelligence, vol. 26, pp. 293-301, 2013.
- [20] Zhiqiang, Liu; Li, Zhang; Xue, Lv and Jie, Chen, “**Evaluation Method about Bus Scheduling Based on Discrete Hopfield Neural Network**,” Journal of Transportation Systems Engineering and Information Technology, vol. 11, no. 2, pp. 77-83, 2011.