



Polibits

ISSN: 1870-9044

polibits@nlp.cic.ipn.mx

Instituto Politécnico Nacional

México

García, Gustavo A.; Salcedo, Octavio  
Predicción de Fallos en Redes IP empleando Redes Neuronales Artificiales  
Polibits, vol. 41, 2010  
Instituto Politécnico Nacional  
Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=402640454009>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Predicción de Fallos en Redes IP empleando Redes Neuronales Artificiales

Gustavo A. García y Octavio Salcedo

**Resumen**—El presente artículo describe la implementación de un sistema de predicción de fallos en redes LAN (fallos de timeout y rechazo en las conexiones), utilizando redes neuronales artificiales Perceptrón Multicapa. Se describe como se implementó el sistema, las pruebas realizadas para la selección de los parámetros propios de la red neuronal, como del algoritmo de entrenamiento y los resultados de evaluación obtenidos.

**Palabras clave**—Predicción de fallos, MIB, red neuronal artificial, perceptrón multicapa, backpropagation.

## Prediction of Failures in IP Networks using Artificial Neural Networks

**Abstract**—The paper presents the implementation of a system for predicting failures in LAN (timeout failure and rejection of connections), using neural networks (multilayer perceptron). It describes the implementation of the system, experiments conducted for the selection of specific parameters of the neural network, training algorithm and evaluation results.

**Index Terms**—Prediction of failures, MIB, Artificial Neural Networks, multilayer perceptron, backpropagation.

### I. INTRODUCCIÓN

LA idea principal en la predicción de fallos es predecir fallas catastróficas en la red, de manera que se pueda garantizar fiabilidad y calidad de servicio (QoS) en tiempo real para mantener la disponibilidad y fiabilidad de la red e iniciar apropiadas acciones de restauración de la “normalidad”. Es por esto que surge la necesidad de implementar sistemas que por medio de análisis del tráfico de la red puedan predecir los fallos en servidores de archivos que se pudiesen presentar tales como time-out y rechazo de conexiones, existen diferentes técnicas de predicción que serán mencionadas en la siguiente sección, pero la utilizada en el sistema desarrollado está basada en redes neuronales artificiales, a las cuáles se les debe determinar de forma experimental y no teórica la arquitectura y los algoritmos de aprendizaje con los que se entrenará la red neuronal. A continuación se dará una breve introducción a las

herramientas de predicción, posteriormente se mostrara el sistema de predicción de fallos de time-out y de rechazo de conexiones implementado, sus partes y las diferentes pruebas realizadas para encontrar los parámetros del sistema que brinden un mejor desempeño.

### II. MARCO CONCEPTUAL

#### A. Herramientas Empleadas en la Predicción de Fallos

Existen diferentes tipos de herramientas empleadas en la predicción tales como:

##### 1) Redes Neuronales Artificiales

Según Simón Haykin [13] “Una red neuronal es un procesador masivamente paralelo distribuido que es propenso por naturaleza a almacenar conocimiento experimental y hacerlo disponible para su uso.

Este mecanismo se parece al cerebro en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso que se denomina aprendizaje.
- El conocimiento se almacena mediante la modificación de la fuerza o peso sináptico de las distintas uniones entre neuronas”.

Las neuronas artificiales se conocen también como unidades de proceso, y su funcionamiento es simple, pues consiste en recibir en las entradas las salidas de las neuronas vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes. Existen tres tipos de células o unidades [1]:

- Neuronas de entrada: reciben señales desde el entorno; estas entradas (que son a la vez entradas a la red) provienen generalmente de una serie de tiempo con datos anteriores al que se pretende predecir, resultado generalmente de preprocesamientos tales como normalizaciones, derivadas, umbralizaciones entre otros.
- Neuronas de salida: Las unidades de salida envían una señal fuera de la red; en la aplicación de predicción la salida correspondería al valor futuro o estimado.
- Neuronas ocultas: Son aquellas cuyas entradas y salidas se encuentran dentro del sistema; es decir, no tienen contacto con el exterior. Las redes neuronales pueden aprender de experiencias que son provistas como entrada-salida de la red sin necesidad de expresar la relación exacta entre la(s) entrada(s) y la salida, éstas pueden generalizar la experiencia aprendida y obtener la salida correcta cuando nuevas situaciones son encontradas [4].

Manuscrito recibido el 23 de enero de 2010. Manuscrito aceptado para su publicación el 31 de mayo de 2010.

Gustavo A. García es docente de la Universidad Central, Bogotá D.C., Colombia y candidato a Magíster en Ciencias de la información y las comunicaciones de la Universidad Distrital Francisco José de Caldas, Colombia (e-mail: gustavoandresg@gmail.com).

Octavio Salcedo es docente de la Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia y candidato a PhD en ingeniería Informática Universidad de Salamanca, Colombia (e-mail: octavionetworking@gmail.com).

### 2) Modelos autorregresivos (AR)

Son modelos comúnmente usados para describir señales de series de tiempo estocásticas no estacionarias, y su característica principal es que van más allá de medidas estadísticas como la media y la varianza [5] [2], un modelo autorregresivo como lo menciona Proakis, es un proceso de solo polos cuya función de transferencia en  $Z$  se muestra en la ecuación 1 el cual es denominado proceso autorregresivo de orden  $p$  [3].

$$H(Z) = \frac{1}{1 + \sum_{k=1}^p a_k Z^{-k}} \quad (1)$$

### 3) Autómatas de Aprendizaje

De acuerdo a Kyriakakos *et al.* [14] los autómatas de aprendizaje (LA por sus siglas en inglés), son sistemas adaptativos de estados finitos que interactúan continuamente con un ambiente general. A través de la respuesta de un proceso probabilístico de ensayo y error, los LA aprenden a escoger o a adaptarse a un comportamiento que genera la mejor respuesta. Como primer paso de un proceso de aprendizaje una entrada es provista al autómata del medio en que se encuentra, esta entrada acciona uno de los posibles candidatos (estados) del autómata, el medio recibe y evalúa la respuesta, luego provee retroalimentación al autómata la cual altera la respuesta al estímulo del autómata [6]. Los autómatas de aprendizaje son generalmente considerados sistemas robustos pero no aprendices eficientes, son fáciles de implementar y cuyo funcionamiento generalmente se basa en una matriz de estados de transición, que contiene las probabilidades de transición de un salto, estando en el estado  $i$  al estado  $j$  ( $P_{ij}$ ) [6].

### 4) Circulant Markov Modulated Poisson Process (CMMP)

Esta herramienta captura no sólo las estadísticas de segundo orden como lo hacen los procesos autorregresivos de media móvil (ARMA) sino que también las estadísticas de primer orden cuya distribución puede ser diferente a la Gaussiana, la técnica para construir dicho proceso se explica en detalle en [7]. Y Sang en su artículo [8] describe la manera de cómo emplear dicha herramienta en la predicción de tráfico comparando dicha investigación con los resultados obtenidos con un modelo ARMA.

Estos sistemas de predicción, muestran que ellos son comúnmente correlacionados [9] y el análisis de datos en redes de sistemas de gran escala, revela patrones de tiempo del día y día de la semana. Este tipo de correlación es comúnmente utilizado en proyectos de predicción como los desarrollados por Liang *et al.* [10], en el que analizaron los logs del supercomputador IBM BlueGene/L, con los que desarrollaron el sistema de predicción utilizando correlaciones temporales con los eventos de fallas presentados en las series de tiempo. Sahoo *et al.* [11], estudió las ocurrencias de eventos críticos en un cluster, realizando dos sistemas de predicción de fallos uno para cada uno, sin considerar que ellos se encontraban interrelacionados [9]. Wu *et al.* [12],

detectó fallas de nodo amplio en entre un ambiente de cluster, en donde la correlación temporal de los estados de los nodos fue usada para definir si el funcionamiento era normal.

## III. DESARROLLO METODOLÓGICO

En la figura 1 se muestra el sistema de predicción propuesto, el cual realizará predicciones de fallos de time-out y de conexiones rechazadas al servidor FTP tomando como entrada el conjunto de variables MIB (*Management Information Base*): *IpInReceives*, *IpInDelivers*, *IpOutRequests*, *tcpActiveOpens* y *tcpRetransSec*. Estas variables son tomadas utilizando un agente SNMP en el servidor FTP que se encuentra en la red mostrada en la figura 2 y son utilizadas en la etapa de preprocesamiento del sistema en donde se obtienen datos estadísticos de ellas (medias y desviaciones estándar) para posteriormente pasárselos a las neuronas de entrada de la red neuronal.

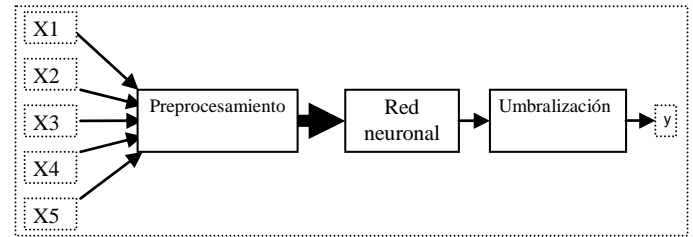


Fig 1 Sistema de predicción de fallos propuesto.

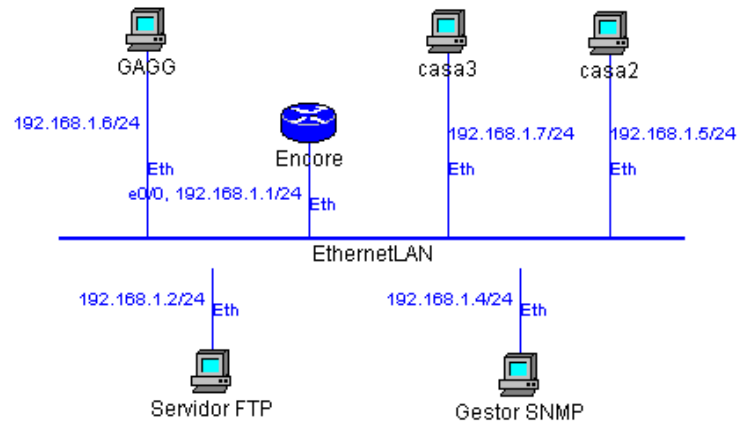


Fig 2. Red LAN de pruebas.

Para encontrar la configuración del sistema de predicción que presente el mejor desempeño en la predicción de fallos se realizaron pruebas de cada una de las siguientes variables del sistema.

- Número de épocas de entrenamiento,
- Variables de entrada a la red neuronal,
- Arquitectura de la red neuronal,
- Algoritmo de aprendizaje de la red neuronal,
- Parámetros del algoritmo de aprendizaje,
- Selección del nivel de umbralización.

#### A. Segmentación de los Datos de Entrada

Para encontrar los parámetros del sistema de predicción propuesto (ver figura 1), se contó con una base de datos que contenía muestras del tráfico entrante y saliente del servidor

FTP, así como sus conexiones activas y cantidad de retransmisiones (variables MIB mencionadas anteriormente), la base de datos contiene información de un mes de pruebas en la red y para motivos de entrenamiento y validación de la eficiencia del sistema propuesto, se segmentó el total de la base de datos en tres partes: 60% para entrenamiento de la red neuronal, 20% para validación del entrenamiento y el otro 20% para pruebas del sistema. Éste último fue utilizado para determinar la eficiencia del sistema implementado, ya que este segmento de datos no fue utilizado en el entrenamiento de la red neuronal, siendo desconocido por el sistema.

#### B. Selección del Número de Épocas de Entrenamiento de la Red Neuronal

La configuración de la red neuronal con la que se probó el número de épocas se encuentra en la tabla I, dicho parámetro fue variado desde un valor de 500 a 4500 en pasos de 500 épocas. Por cada época se realizaban 10 pruebas para verificar la repetitibilidad y consistencia de los resultados. Se seleccionó a 2000 épocas (ver tabla II) como el valor que mejor desempeño presentó. Como base para esta selección se tuvo en cuenta que la idea no es tener el menor error de entrenamiento sino que sea pequeño y que no requiera muchas épocas ya que esto afecta los tiempos de entrenamiento considerablemente.

TABLA I  
PRUEBA DE SELECCIÓN DEL NÚMERO DE ÉPOCAS DE ENTRENAMIENTO.

No. Neuronas ocultas	8
Algoritmo de entrenamiento	BackPropagation (traingd)
Learning rate	0.05
No. neuronas de salida	1
No. neuronas de entrada	5
Tipo de red	Perceptrón Multicapa
No. épocas	500 – 4500
No. pruebas x época	10
Total pruebas	90

TABLA II  
RESULTADOS DE PRUEBAS DE NÚMERO DE ÉPOCAS.

No Épocas	MseMin Prom	UltMse Prom
500	0,1143	0,114323222
1000	0,112908222	0,112908222
1500	0,112372556	0,112372556
2000	0,111893556	0,111893556
2500	0,110850333	0,110850333
3000	0,110573111	0,110573111
3500	0,110519444	0,110519444
4000	0,110476333	0,110476333
4500	0,110439889	0,110439889

#### C. Selección del Número de Entradas a la Red Neuronal

En esta etapa se realizaron pruebas con diferentes entradas a la red neuronal para determinar cuáles generaban un mejor desempeño en el sistema de predicción. Las entradas a la red neuronal probadas fueron: las variables MIB *IpInReceives*, *IpInDelivers*, *IpOutRequests*, *tcpActiveOpens*, *TcpRetransSec*; los valores medios, desviaciones estándar y valores anteriores de las variables *IpInReceives*, *IpInDelivers* e *IpOutRequests*. La conformación de las diversas entradas a la red neuronal, son generadas en la etapa de preprocesamiento (ver figura 1).

TABLA III  
CONFIGURACIÓN DE LA RED NEURONAL, PRUEBAS DE LAS ENTRADAS.

No. Neuronas ocultas	8
Algoritmo de entrenamiento	BackPropagation con momento (traingd)
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	5 a 26
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x conjunto de entradas	10
Total pruebas	50

En la tabla III se encuentra la configuración de la red neuronal utilizada en las pruebas para las diferentes entradas. En el proceso de verificación de resultados por ser una prueba muy importante en el sistema de predicción, se realizó la verificación visual de las salidas del sistema. La figura 2 muestra las salidas del sistema en la que la línea verde cuando toma el valor de uno, representa los puntos en los cuáles el sistema debe indicar que se va a presentar un fallo en el servidor, y las líneas azules son aquellas en las que el sistema predijo que se iba a presentar una fallo en el servidor FTP. En esta figura la imagen *DatosOriMediaDesv* correspondiente a la salida del sistema se obtuvo que las entradas dadas por: las variables MIB, los valores medios y las desviaciones estándar de los últimos veinte minutos de adquisición, fueron las que mejor comportamiento presentaron utilizando como criterio la cantidad de veces que el sistema predijo correctamente vs. la cantidad de predicciones erradas.

Los resultados de las pruebas de variables de entrada a la red neuronal se presentan en la Figura 2, en el Anexo.

#### D. Selección del Número de Neuronas Ocultas en la Red Neuronal

La configuración empleada en la selección del número de neuronas ocultas se muestra en la tabla IV donde se observa que los resultados obtenidos hasta el momento fueron incluidos en las pruebas. Se realizaron pruebas variando el número de neuronas ocultas de dos a quince, y cuyos resultados se resumen en la tabla V. En esta tabla de resultados se adicionaron parámetros que ayudan en la selección de las pruebas que presentó mejor comportamiento

en la predicción de fallos, no sólo teniendo en cuenta la cantidad de predicciones acertadas (*nOk*) sino también las erradas (*nErradas*).

La configuración empleada en la selección del número de neuronas ocultas se muestra en la tabla IV donde se observa que los resultados obtenidos hasta el momento fueron incluidos en las pruebas. Se realizaron pruebas variando el número de neuronas ocultas de dos a quince, y cuyos resultados se resumen en la tabla V. En esta tabla de resultados se adicionaron parámetros que ayudan en la selección de las pruebas que presentó mejor comportamiento en la predicción de fallos, no sólo teniendo en cuenta la cantidad de predicciones acertadas (*nOk*) sino también las erradas (*nErradas*).

TABLA IV  
CONFIGURACIÓN DE LA RED NEURONAL  
PARA PRUEBAS DE NEURONAS OCULTAS.

No. Neuronas ocultas	2-15
Algoritmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x neurona	10
Total pruebas	140

Los resultados de las pruebas de selección de número de neuronas en la capa oculta se presentan en la Tabla V en el Anexo.

Se seleccionó la configuración de la red neuronal con 11 neuronas en la capa oculta, ya que el valor promedio de la relación entre el número de predicciones acertadas vs. el número de predicciones erradas es el cuarto más alto, pero la desviación estándar es la más baja de éstas, lo que representa una mayor homogeneidad de las predicciones, adicionalmente verificando la cantidad de predicciones acertadas vs. las erradas de la prueba en específico se ve que esta prueba tuvo el 61 % de predicciones acertadas esta configuración.

#### E. Selección del Algoritmo de Entrenamiento

Se realizaron pruebas con los siguientes algoritmos de entrenamiento:

- **traingd**: Backpropagation de gradiente descendente;
- **traingdm**: Backpropagation de gradiente descendente y momento;
- **traingda**: Backpropagation de gradiente descendente con tasa de entrenamiento adaptativa;
- **trainrp**: *Resilient* Backpropagation.

A cada uno de los algoritmos se le realizaron diez pruebas para determinar cuál de los cuatro algoritmos probados presenta el mejor comportamiento en la predicción de fallos. En la tabla VI se muestra la configuración de la red neuronal como se realizaron las pruebas y en la tabla VII el resumen de

resultados de éstas. En esta última se puede observar que el algoritmo seleccionado es el algoritmo **traingdm** (Backpropagation de gradiente descendente y momentum) aunque el que mayor índice de predicciones correctas vs predicciones erradas fue el **trainrp**. La razón obedece a que este último presenta en promedio un número elevado de predicciones erróneas como lo son 389 predicciones erradas vs. las 204 erradas que presentó el algoritmo seleccionado.

TABLA VI  
CONFIGURACIÓN DE LA RED NEURONAL,  
PRUEBAS DEL ALGORITMO DE APRENDIZAJE.

No. Neuronas ocultas	11
Algoritmo de entrenamiento	traingd, traingdm, traingda, trainrp
Learning rate	0.05
Momento	0.04
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x Algoritmo	10
Total pruebas	40

Los resultados de las pruebas de la selección del algoritmo de entrenamiento se presentan en la Tabla VII, en el Anexo.

#### F. Selección del Momento en el Algoritmo de Entrenamiento

Después de haber encontrado que el algoritmo de entrenamiento de la red neuronal es el **traingdm**, sus parámetros son: el momentum y la tasa de aprendizaje (*learning rate*). A continuación en la tabla VIII se puede ver la configuración de la red neuronal con la que se probó el parámetro momento y en la tabla IX se observa el resumen de resultados de la prueba. El parámetro momentum seleccionado fue el de 0.10 ya tiene un índice de predicción alto (del 0.68), el número de predicciones es mucho mayor al de 0.72 que sólo tuvo 72 predicciones acertadas y el número de fallos no fue muy elevado (fue menor a 250 fallos que en la práctica se observó que era un número poco eficiente para la funcionalidad del sistema).

TABLA VIII  
CONFIGURACIÓN DE LA RED NEURONAL,  
PRUEBAS DE SELECCIÓN DE MOMENTUM.

No. Neuronas ocultas	11
Algoritmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.05
Momento	0.01 - 0.15
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x momento	10
Total pruebas	150

Los resultados de las pruebas de momentum del algoritmo de aprendizaje se presentan en la Tabla IX en el Anexo.

#### G. Selección de la Tasa de Aprendizaje en el Algoritmo de Entrenamiento

Como se mencionó anteriormente el otro parámetro a definir en el algoritmo de aprendizaje es la tasa de entrenamiento, a continuación en la tabla X se encuentra la configuración de la red neuronal para las pruebas y en la tabla XI los resultados. En esta última se encuentra seleccionado la tasa de aprendizaje de 0.04 el cual comparado con los demás resultados que presentan el indicador de predicciones correctas vs. incorrectas.

TABLA X  
CONFIGURACIÓN DE LA RED NEURONAL,  
PRUEBAS DE SELECCIÓN DE LA TASA DE APRENDIZAJE.

No. Neuronas ocultas	11
Algoritmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.01 - 0.15
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x learning rate	10
Total pruebas	150

Los resultados de las pruebas de selección de la tasa de aprendizaje del algoritmo de aprendizaje se presentan en la tabla XI en el Anexo.

#### H. Selección de Umbral

Como se observa en la figura 1, la umbralización es la última etapa del sistema de predicción y juega un papel muy importante en el sistema ya que es la encargada de seleccionar cuáles salidas de la red neuronal serán consideradas como un fallo en la red (Fallo en el servidor FTP) y cuáles no.

TABLA XII  
CONFIGURACIÓN DE LA RED NEURONAL,  
PRUEBAS DE SELECCIÓN DE UMBRAL.

No. Neuronas ocultas	11
Algoritmo de entrenamiento	BackPropagation con momento( traingdm)
Learning rate	0.04
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x umbral	10
Umbral	1:0.1:2.5
Total pruebas	160

Es por esto que se realizaron pruebas para determinar el nivel de umbral con el que el sistema de predicción de fallos presenta un nivel alto de desempeño, en la tabla XII se encuentra la configuración de la red neuronal de las pruebas y en la tabla XIII (ver Anexo) los resultados.

Se escogió el umbral de 2.3 ya que éste presenta un nivel alto de predicciones correctas vs. las erradas (0.56) (comparado con las demás pruebas) y adicionalmente tiene desviación estándar baja y el promedio del indicador alto.

## IV. RESULTADOS

En la tabla XIV se encuentra la configuración final del sistema de predicción de fallos propuesto en la figura 1, en ella se resume el trabajo desarrollado en la investigación para la configuración del sistema de preprocesamiento y en la tabla XIII se encuentran los resultados de las pruebas de predicción con datos de prueba diferentes a los de entrenamiento de la red neuronal. En ella se puede ver que el sistema presenta un índice de predicciones correctas vs. las erradas de un 66%, lo que nos dice que las redes neuronales perceptrón multicapa son herramientas válidas para la predicción de fallos en redes LAN aunque se debe buscar otra arquitectura de red que permita mejorar el desempeño del sistema.

TABLA XIV  
CONFIGURACIÓN FINAL DEL SISTEMA DE PREDICCIÓN.

No. Neuronas ocultas	11
Algoritmo de entrenamiento	BackPropagation con momento (traingdm)
Learning rate	0.04
Momento	0.01
No. neuronas de salida	1
No. neuronas de entrada	11
Tipo de red	Perceptrón Multicapa
No. épocas	2000
No. pruebas x learning rate	10
Umbral	2.3
Total pruebas	40

En la prueba 36 como se observa en la tabla XV se obtuvo un sistema de predicción en el que el número de predicciones incorrectas es menor a 100 y el número de predicciones correctas fue de 65. La figura 3 muestra el resultado de esta prueba siendo la línea verde (punteada) en uno el intervalo de tiempo en el cual el sistema debe realizar predicciones y las líneas azules (líneas discontinuas) los momentos en los que el sistema realiza la predicción.

TABLA XV  
RESULTADOS DE PRUEBAS DEL SISTEMA DE PREDICCIÓN.

Prueba	MseEntMin	nErradas	nOk	(nOk/nErradas)
1	0.109845	266	148	0,56
2	0.110150	225	113	0,50
3	0.111292	91	33	0,36

Prueba	MseEntMin	nErradas	nOk	(nOk/nErradas)
4	0.110363	182	110	0,60
5	0.110100	1	0	0,00
6	0.111106	2	0	0,00
7	0.110305	45	14	0,31
8	0.110122	4	2	0,50
9	0.110485	1	0	0,00
10	0.110099	176	83	0,47
11	0.110705	243	102	0,42
12	0.111991	1	0	0,00
13	0.110137	23	10	0,43
14	0.110764	107	70	0,65
15	0.110488	21	15	0,71
16	0.110766	191	77	0,40
17	0.109571	5	3	0,60
18	0.110636	244	104	0,43
19	0.110650	13	4	0,31
20	0.110410	215	103	0,48
21	0.111930	117	49	0,42
22	0.110180	185	85	0,46
23	0.110988	48	33	0,69
24	0.109985	18	12	0,67
25	0.109250	170	91	0,54
26	0.110232	21	17	0,81
27	0.110769	1	0	0,00
28	0.109736	1	0	0,00
29	0.109395	29	23	0,79
30	0.109827	1	0	0,00
31	0.109284	1	0	0,00
32	0.110412	38	28	0,74
33	0.110645	9	4	0,44
34	0.109890	35	23	0,66
35	0.110058	134	53	0,40
36	0.110605	99	65	0,66
37	0.110693	41	16	0,39
38	0.110444	22	13	0,59
39	0.111245	126	61	0,48
40	0.112721	160	71	0,44

## V. CONCLUSIONES

- Para el desarrollo de sistemas de predicción de fallos en redes se requiere que sean de baja complejidad computacional para que el tiempo utilizado en la predicción de fallos permita que el sistema sea implementable.
- Las redes neuronales perceptrón multicapa son una herramienta útil en la predicción de fallos, aunque se

deben probar otros algoritmos de entrenamiento para mejorar el desempeño del sistema de predicción obtenido.

- Las variables MIB *IpInreceives*, *IpIndelivers*, *IpOutRequests*, *TcpActiveOpens*, *tcpRetranSec* permiten determinar las fallas de una red LAN, utilizando sus valores medios y desviaciones estándar a la entrada de la red neuronal.
- Con las entradas a la red neuronal *ipInreceives*, *ipIndelivers*, *ipOutRequests*, *tcpActiveOpens*, *tcpRetranSec*, sus valores medios y desviaciones estándar, la arquitectura que mejor se comporta en la predicción es la que tiene once neuronas en la capa oculta.
- De los algoritmos backpropagation de gradiente descendente, backpropagation de gradiente descendente y momento, backpropagation de gradiente descendente con tasa de entrenamiento adaptativa y resilient backpropagation, el mejor para la predicción de fallos de una red LAN utilizando red neuronal perceptrón multicapa de once neuronas en la capa oculta, es el backpropagation de gradiente descendente y momento, con parámetros tasa de entrenamiento de 0.04 y un momentum de 0.01.
- Para la determinación del fallo de la red LAN, el mejor parámetro para la umbralización de la salida de la red neuronal es que sea de 2.3 veces la desviación estándar de las últimas 120 salidas de ésta.
- Se podría mejorar el desempeño del sistema propuesto utilizando una arquitectura de red neuronal diferente o algoritmos que permitan un mejor aprendizaje a la red neuronal.

## REFERENCIAS

- [1] J. R. Hilera y V. J. Martínez, *Redes Neuronales Artificiales*, Alfaomega, 2000.
- [2] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis, Forecasting and Control*, Holden Day Series, 1976.
- [3] J. G. Proakis y D. G. Manolakis, *Tratamiento Digital de Señales*, Prentice-Hall, 1998.
- [4] Yen-Chieh Ouyang and Li-Bin Yeh, "Predictive bandwidth control for mpeg video: A wavelet approach for self-similar parameters estimation," in *IEEE International Conference on Communications ICC 2001*, vol. 5, 2001, pp. 1551-1555.
- [5] M. Thottan and C. Ji, "Fault prediction at the network layer using intelligent agents," in *IFIP/IEEE Eighth International Symposium on Integrated Network Management*, 2003, pp. 745-759.
- [6] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547-588, Apr. 1965.
- [7] N. Frangiadakis, M. Kyriakos and L. Merakos, "Enhanced path prediction for network resources management in wireless LANs," *IEEE wireless communications*, pp.62-69, 2003.
- [8] San Qi Li and Chia Lin Hwang, "On the convergence of traffic measurement and queuing analysis: a statistical-matching and queuing (SMAQ) tool," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, 1997, pp. 95-110.
- [9] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. IEEE Int. Conf. Communications*, 1995, pp. 3-8.
- [10] Aimin Sang and San-Qi Li, "A predictability analysis of network traffic," in *IEEE INFOCOM*, 2000.
- [11] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *Proc. INTERMAG Conf.*, 1987, pp. 2.2-1-2.2-6.
- [12] Ziming Zhang and Song Fu, "Failure prediction for automatic Management of networked computer systems with availability assurance," in *15th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems*, 2010.

- [13] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette, and R. K. Shahoo, "BlueGene/L failure analysis and prediction models," in *Proceedings of international conference on dependable Systems and networks (DSN)*, 2006.
- [14] R. K. Sahoo, A. J. Oliner, and I. Rish, "Critical event prediction for proactive management in large-scale computer clusters," in *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, August 2003.
- [15] L. Wu, D. Meng, W. Gao, and J. Zhan, "A proactive fault-detection mechanism in large-scale cluster systems," in *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.
- [16] S. Haykin, *Neural Networks for pattern recognition*, Oxford University Press Inc.
- [17] N. Frangiadakis, M. Kyriakos, and L. Merakos, "Enhanced path prediction for network resources management in wireless lans," *IEEE wireless communications*, pp. 62–69. 2003.

## ANEXO

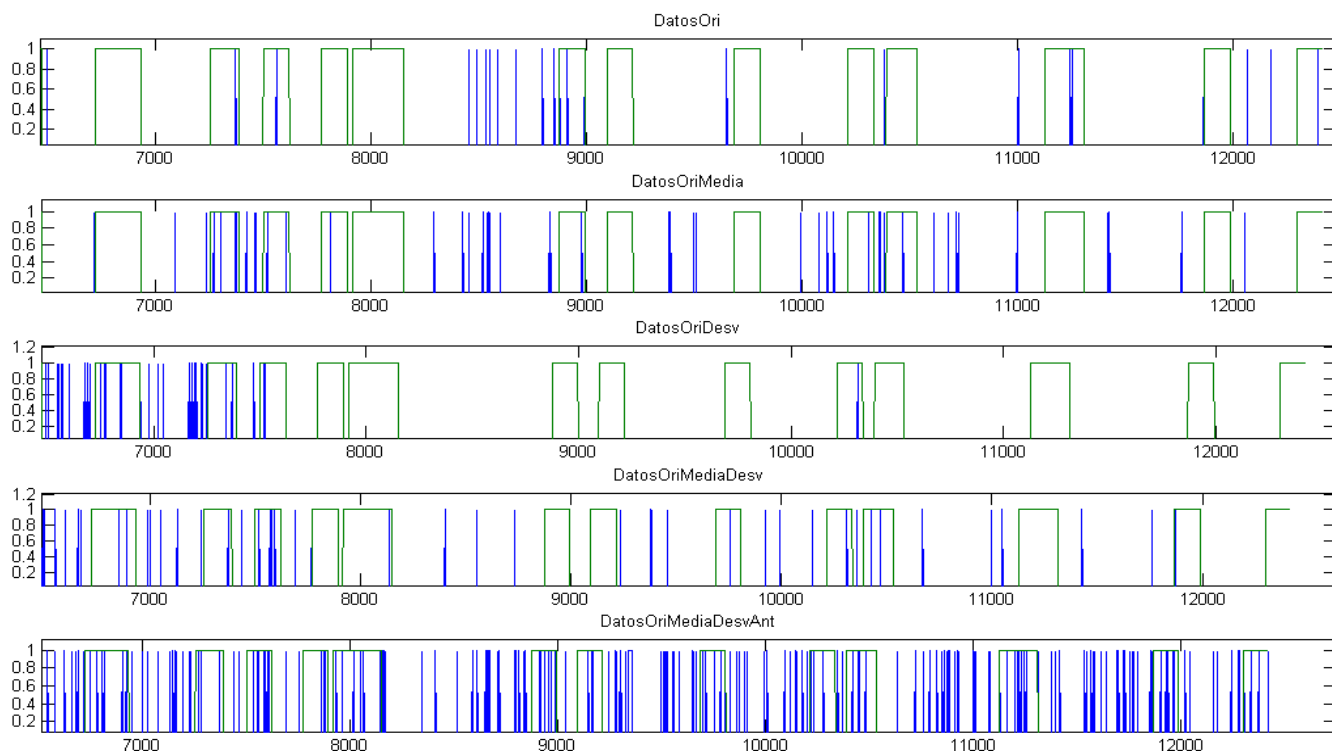


Fig 2. Resultados de pruebas de variables de entrada a la red neuronal.

 TABLA V  
 RESULTADOS DE PRUEBAS DE SELECCIÓN DE NÚMERO DE NEURONAS EN LA CAPA OCULTA.

Prueba No	No. Neuronas	Mse Ent	nErradas	nOk	nOk Prom	(nOk/nErradas)	(nOk/nErradas) prom	Desv Est nOk
9	2	0,110531	198	117	91,3	0,59	0,47	0,21
3	3	0,112086	133	93	82,8	0,70	0,44	0,21
3	4	0,110273	161	107	91	0,66	0,59	0,16
9	5	0,110105	233	122	125,5	0,52	0,44	0,16
3	6	0,111664	186	133	70,3	0,72	0,49	0,24
1	7	0,109914	227	144	63,2	0,63	0,52	0,27
3	8	0,110727	181	114	68,7	0,63	0,48	0,19
4	9	0,109966	94	69	38,5	0,73	0,32	0,31
2	10	0,109988	180	111	85,6	0,62	0,41	0,23
6	11	0,110726	219	134	96,2	0,61	0,51	0,07
3	12	0,109578	158	105	107,4	0,66	0,45	0,14
9	13	0,112075	233	139	90,4	0,60	0,62	0,16
4	14	0,110857	247	149	95,3	0,60	0,47	0,15
9	15	0,110076	171	103	58,9	0,60	0,37	0,23



TABLA VII  
RESULTADOS DE PRUEBAS DE SELECCIÓN DEL ALGORITMO DE ENTRENAMIENTO

Prueba	Algoritmo	MseEntMin	nErradas	nOk	nOk / nErradas	Prom Ind	Prom Nok	DesvEst Ind
2	traingd	0,1103	315,00	162,00	0,51	0,37	56,60	0,15
1	traingdm	0,1107	204,00	124,00	0,61	0,54	100,20	0,16
4	traingda	0,1074	306,00	161,00	0,53	0,44	86,80	0,05
5	trainrp	0,1035	389,00	241,00	0,62	0,54	234,90	0,06

TABLA IX  
RESULTADOS DE PRUEBAS DE MOMENTUM DEL ALGORITMO DE APRENDIZAJE.

PruebaNo	mc	MseEntMin	nErradas	nOk	(nOk/nErradas)	Promedio Ind	Desv Est	Error Promedio
10	0,01	0,110705	204	124	0,61	0,40	0,19	4325,3
8	0,02	0,110027	305	149	0,49	0,51	0,17	4368,7
8	0,03	0,110327	109	78	0,72	0,55	0,32	4292,6
2	0,04	0,110501	196	122	0,62	0,50	0,20	4345,1
5	0,05	0,111212	303	159	0,52	0,55	0,19	4386,6
5	0,06	0,109703	236	124	0,53	0,41	0,19	4347,1
6	0,07	0,111071	201	127	0,63	0,40	0,23	4345,2
3	0,08	0,110303	227	133	0,59	0,47	0,21	4332,4
4	0,09	0,109769	235	140	0,60	0,50	0,23	4335,7
7	0,10	0,110687	159	108	0,68	0,45	0,19	4368
8	0,11	0,110848	236	111	0,47	0,39	0,22	4334,9
4	0,12	0,109861	283	142	0,50	0,44	0,17	4333,4
9	0,13	0,110393	189	120	0,63	0,49	0,23	4332,9
4	0,14	0,110068	168	108	0,64	0,45	0,18	4363,4
7	0,15	0,109735	261	137	0,52	0,41	0,17	4384,2

TABLA 11  
RESULTADOS DE PRUEBAS DE LA TASA DE APRENDIZAJE DEL ALGORITMO DE APRENDIZAJE.

Prueba	lr	MseEntMin	nErradas	nOk	(nOk / nErradas)	Promedio Ind	Desv Est	Promedio indicador	Error Promedio	NErr Promedio
5	0,01	0,110231	2	2	1,00	0,41	0,26	90,7	4386,8	206,5
7	0,02	0,109600	38	30	0,79	0,51	0,19	95,3	4368,9	193,2
10	0,03	0,109871	5	5	1,00	0,54	0,25	72,6	4335,9	137,5
4	0,04	0,109890	106	70	0,66	0,51	0,20	74,2	4337,1	140,3
7	0,05	0,110586	1	1	1,00	0,51	0,26	79,4	4356,2	164,6
4	0,06	0,110092	71	56	0,79	0,49	0,21	105,3	4373,2	207,5
10	0,07	0,110963	201	127	0,63	0,35	0,25	53,6	4321,6	104,2
2	0,08	0,110089	218	133	0,61	0,48	0,11	85,7	4348,3	163
4	0,09	0,109995	3	3	1,00	0,51	0,26	62,7	4322,5	114,2
7	0,10	0,109770	79	58	0,73	0,38	0,25	42,6	4319,5	91,1
2	0,11	0,110474	100	74	0,74	0,44	0,27	78	4340,8	147,8
7	0,12	0,109633	194	116	0,60	0,48	0,12	61,9	4333,9	124,8
8	0,13	0,109807	22	22	1,00	0,40	0,32	29,9	4297,3	56,2
4	0,14	0,110667	34	28	0,82	0,49	0,24	60,9	4333	122,9
1	0,15	0,109805	82	55	0,67	0,41	0,21	65,6	4338,8	133,4

TABLA XIII  
RESULTADOS DE PRUEBAS DE LA SELECCIÓN DEL UMBRAL.

Prueba No	Umbral	MseEntMin	nErradas	nOk	(nOk/nErradas)	Promedio Ind	Desv Est	Promedio indicador	Error Promedio	NErr Promedio
5	1,9	0,110100	5	4	0,80	0,39	0,23	90,9	4378,5	198,4
1	2	0,109845	382	198	0,52	0,38	0,17	78,3	4356,4	163,7
4	2,1	0,110363	249	134	0,54	0,35	0,20	66,6	4339,9	135,5
1	2,2	0,109845	298	161	0,54	0,31	0,24	57,5	4329,4	115,9
1	2,3	0,109845	266	148	0,56	0,33	0,24	50,3	4320	99,3
1	2,4	0,109845	228	128	0,56	0,38	0,32	42,1	4313	84,1
4	2,5	0,110363	131	79	0,60	0,33	0,24	36,8	4306,5	72,3

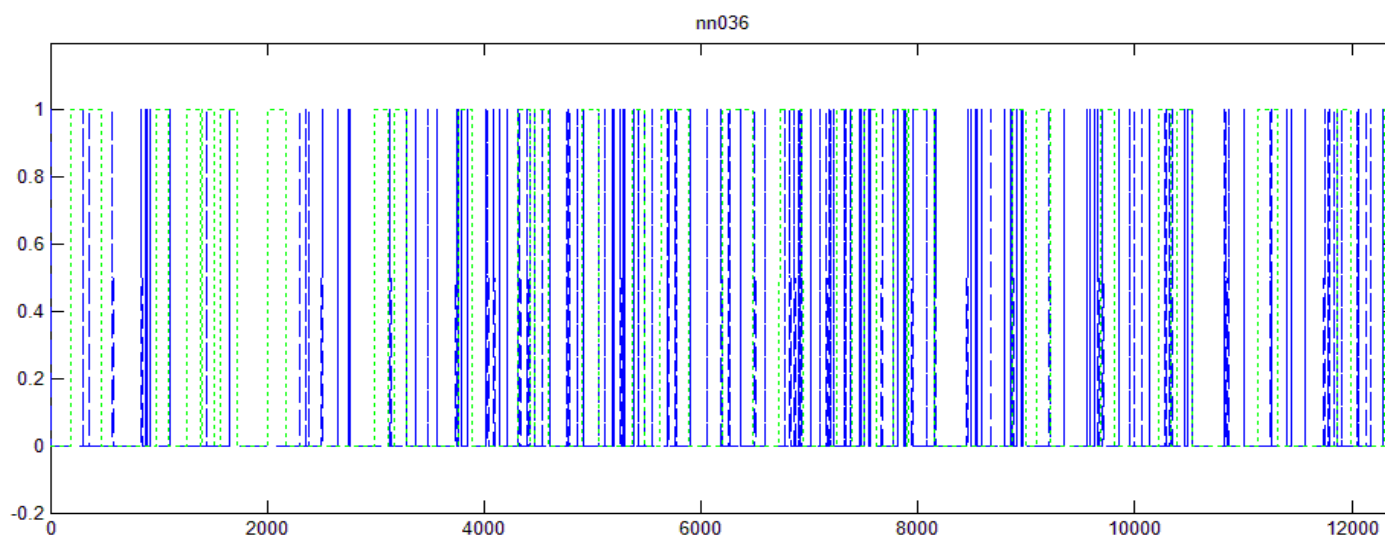


Fig. 3. Gráfica de resultados del sistema de predicción.