[1] SIMULACIÓN DE UN SISTEMA INTELIGENTE PARA LA DETECCIÓN DE INTRUSOS

Andrés Gómez Ramírez[1]

Juan Diego López Escudero[2]

Resumen: *En este trabajo se realiza una primera aproximación a la aplicación de la Inteligencia Artificial en el campo de la seguridad informática. Se propone la construcción de un Sistema de Detección de Intrusos basado en una red neuronal de tipo Perceptrón Multicapa con el algoritmo de Backpropagation para el proceso de aprendizaje; este sistema analizará algunas características de un conjunto de paquetes de red, con el fin de determinar cuándo la información presentada en este tráfico puede ser considerada como normal o por el contrario como un tipo de ataque informático.*

Índices: *Redes Neuronales Artificiales, Reconocimiento de patrones, Sistema de detección de intrusos, Seguridad informática.*

Abstract: *This work is a first approach to the application of Artificial Intelligence in computer security field. The construction of an Intrusion Detection System based on a Multi Layer Perceptron neural network is proposed to perform the learning process using Backpropagation algorithm; this system will analyze some features from a network packet set in order to determine whether the information submitted in this traffic can be considered as normal or as a type of computer attack instead.*

Indexes: *Artificial Neural Networks, Pattern Recognition, Intrusion Detection Systems, Information Security.*

1Estudiante Ingeniería de Sistemas, Universidad de Antioquia. Miembro del Grupo de Investigación GEPAR. andresgomezram7@gmail.com.
2Estudiante Ingeniería de Sistemas, Universidad de Antioquia. Jdiego22@gmail.com.

1. INTRODUCCIÓN

En los últimos años, la digitalización de la información ha sido una constante para la gran mayoría de las instituciones; hoy en día, se puede decir que éstas dependen de sus sistemas informáticos, por lo tanto, un aspecto prioritario es que esta información permanezca bien protegida [5]. La seguridad informática es una de las principales áreas de investigación en la actualidad en el campo de las ciencias de la computación.

Uno de los aspectos clave para que un sistema pueda garantizar un nivel de seguridad, es la capacidad de detectar en tiempo real un ataque que intente vulnerarlo y de esta forma realizar las acciones de respuesta determinadas para evitar que el intento de intrusión sea exitoso [15].

Actualmente existen numerosos Sistemas de Detección de Intrusos (IDS), siendo los más exitosos los basados en Sistemas Expertos (SE), sin embargo, el uso de Redes Neuronales Artificiales (RNA), ha sido propuesto como una poderosa alternativa que puede llegar a solucionar los problemas que se presentan en el diseño de un IDS [10].

Este artículo se divide en dos partes; en la primera sección se explican algunos conceptos fundamentales para entender el funcionamiento y las características que debe tener un IDS, también se exponen algunos antecedentes que muestran la forma en la cual ha ido evolucionando el diseño de estos sistemas.

En la segunda parte se explica cómo se obtuvieron los datos de entrenamiento y prueba,

la estructura de la red neuronal y finalmente, se presentan los resultados obtenidos y las conclusiones.

2. MARCO TEÓRICO

A. Seguridad informática

Es una disciplina que tiene como objetivo el tratar de garantizar la integridad, confidencialidad, disponibilidad e irrefutabilidad de la información en un sistema informático.

Los ataques informáticos son acciones o actividades que buscan vulnerar la seguridad de un sistema, un ataque es llevado a cabo generalmente por personas ajenas y sin acceso al sistema víctima, aunque también se presentan casos en los que el ataque es perpetuado por un usuario que cuenta con acceso y determinados permisos.

Para tratar de garantizar la seguridad, se crearon los Sistemas de Detección de Intrusos. Un IDS está diseñado para cumplir con tareas de monitorización en las redes y sistemas, siendo su principal objetivo detectar cuando las políticas de uso han sido violadas y se están realizando actividades anormales o maliciosas.

B. Redes neuronales artificiales

Antes de definir este concepto es importante definir la rama de la ciencia de la computación de la cual hace parte la Inteligencia Artificial (IA). El objetivo de la IA es imitar la inteligencia de los seres humanos, crear artefactos racionales, es decir, agentes no vivos que sean capaces de percibir estímulos del medio en el que se encuentren, procesar dichos estímulos o entradas y producir una respuesta que sea la más adecuada.

Las Redes Neuronales Artificiales (RNA) son modelos matemáticos conexionistas inspirados en la forma en que funciona el sistema nervioso de los animales y especialmente de los seres humanos. Son utilizadas desde el punto de vista de la IA, para emular procesos de aprendizaje y reconocimiento asociativo; desde el punto de vista matemático, se les caracteriza más como modelos para aproximar funciones (regresión) y como métodos de clasificación de patrones. Se trata de un sistema de interconexión de elementos de cálculo o neuronas, por su símil biológico, en una red (Fig. 1) que procesa datos de entrada para producir una acción de salida [1].

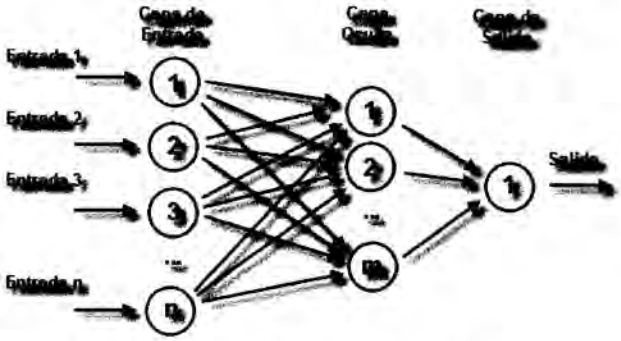

Fig. 1 Esquema general de una red neuronal.

El procedimiento por medio del cual una RNA es ajustada se llama aprendizaje o entrenamiento y se lleva a cabo utilizando ejemplos o muestras del espacio de datos. En este entrenamiento se logra ajustar los pesos Wi de las conexiones (Fig. 2) iterativamente hasta que la red demuestre un comportamiento deseado. Existen dos tipos principales de entrenamiento, el supervisado y el no supervisado. En el supervisado, cada muestra de datos viene con su respectiva salida esperada de la red neuronal. Para el tipo no supervisado, es la propia red la que decide qué salida dar para cada caso.

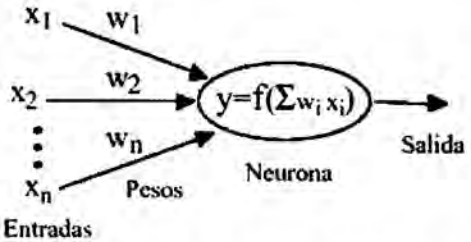

Fig. 2 Representación de una neurona artificial.

En este artículo se utiliza el algoritmo de entrenamiento supervisado conocido como Backpropagation o retropropagación, que básicamente consiste en calcular el error

obtenido por la red neuronal al pasarle cada uno de los ejemplos y distribuir este error hacia las diferentes capas de la red, modificando iterativamente los pesos hasta que se logren las salidas esperadas.

Una vez entrenada una RNA, se procede a enviar a la red un conjunto de datos diferente al de entrenamiento para comprobar que el comportamiento sea el esperado [2]. Para ilustrar lo anterior, como ejemplo de entrenamiento se puede tener un conjunto de imágenes que representen las letras del alfabeto, así la red neuronal procesa estas imágenes (ajusta los pesos) hasta que aprende a identificar correctamente las letras con su respectivo código ASCII. Finalmente se procede a realizar la prueba con otro conjunto de imágenes, esta vez con pequeñas diferencias, que pueden ser del tipo que habitualmente debería reconocer la red neuronal [1].

3. ANTECEDENTES

A. Sistemas de detección de intrusos

A medida que la importancia de los sistemas informáticos fue creciendo, la seguridad se convirtió en un tema de vital importancia; cuando comenzaba la década de los 80's, se publica "Monitoreo y vigilancia de las amenazas a la seguridad informática" [3], trabajo en el cual se describe la importancia que tienen los registros de auditoría propios de cada Sistema Operativo, se explica que a partir del análisis de estos archivos (donde se lleva un registro detallado de todas las actividades que se realizan en el sistema) se pueden identificar amenazas. Anderson, su autor, identificó tres tipos de amenazas: Los usuarios sin autorización que tratan de ganar acceso al sistema, los usuarios autorizados que utilizan el sistema en una manera no autorizada y los usuarios que usan mal sus privilegios de acceso.

A partir del trabajo de Anderson, los investigadores comenzaron a desarrollar técnicas automatizadas para el análisis de los reportes generados por los sistemas de auditoría.

Seis años más tarde, en 1886, la Doctora Dorothy Denning publicó un completo modelo para la detección de intrusos llamado Sistema Experto para Detección de Intrusos (IDES, por sus siglas en inglés) [4]; este modelo no está atado a un sistema operativo específico ni a las vulnerabilidades inherentes a éste, es decir, fue diseñado para detectar intrusiones incluso sin conocer la falla del sistema que el atacante está explotando, lo que lo hace un framework para IDS de propósito general.

Los perfiles de usuario son el principal componente de IDES; un perfil es una estructura que reúne el comportamiento de cada usuario respecto a los recursos que conforman el sistema en término de unas métricas determinadas.

El modelo de Denning ha sido ampliamente aceptado y la mayoría de las metodologías actuales se basan en este trabajo: para la detección de anormalidades se basan en el análisis de los perfiles creados a partir de los sistemas de auditoría del sistema y para la detección del mal uso (abuso) del sistema se basan en el conjunto de reglas de un Sistema Experto (SE) [5]. Un SE emula el conocimiento de una o varias personas especialistas en un determinado tema, se trata de crear una base de conocimiento que pueda responder de manera rápida y acertada a consultas masivas.

B. Detección de intrusiones con redes neuronales

Una cantidad limitada de investigaciones se han realizado acerca de la aplicación de RNA al campo de la seguridad informática. Las redes neuronales artificiales ofrecen mucho potencial para la solución de los problemas encontrados en los diseños de los IDS actuales. Las RNA han sido propuestas específicamente para identificar las características típicas de los usuarios de un sistema e identificar estadísticamente las variaciones significativas en su comportamiento.

El modelo más conocido de RNA es el Perceptron Multicapa (MLP, Multi Layer Perceptron) [6], esencialmente está compuesto por un conjunto de unidades sensoriales que conforman la capa de acceso, una o más capas medias u ocultas, y una capa de salida de nodos computacionales, la señal de entrada se propaga de una manera no recurrente, no existen conexiones periódicas, todos los nodos distintos a los de la capa de entrada tienen su propia función de activación. Las funciones de activación se usan para evitar la linealidad en la red. La Fig. 1 nos muestra su representación gráfica.

James Cannady propuso en 1998 un modelo de IDS basado en estructuras de tipo MLP [5] para el análisis de redes de computadores. La idea era extraer cierta información de los paquetes que circulan por la red, datos como el puerto de origen y de destino, la dirección IP de origen y de destino, además de algunas características de los protocolos, conforman la entrada de la RNA. Luego del proceso de entrenamiento, se obtuvo un buen resultado al detectar correctamente el 97% de los ataques informáticos.

En el mismo año se propuso otro sistema llamado NNID (Neural Network Intrusion Detection) [7] igualmente basado en redes MLP, pero en este caso lo que se hizo fue definir un conjunto de perfiles de usuarios legítimos basándose en los comandos utilizados en el equipo, el número de veces de ejecución de dichos comandos y su frecuencia en el tiempo. De esta manera se logró detectar cuando entraba al sistema un usuario no autorizado con un 96% de efectividad y un 7% de falsas alarmas.

La red de Mapas Auto Organizados (SOM, Self-Organizing Map) [8,9] es uno de los modelos basados en redes neuronales más usados debido a que posee ventajas como la de necesitar una menor cantidad de tiempo en el entrenamiento, debido a que usa un aprendizaje de tipo no supervisado. De esta forma, SOM ofrece un modelo sencillo y eficaz para clasificar y procesar los datos en tiempo real mediante mapas de caracterización auto-organizativos, superando en velocidad a otras técnicas de aprendizaje. Las RNA de tipo SOM se han usado en varios IDS para aprender a identificar los patrones de uso normal de un sistema [10].

La red de Función de Base Radial (RBF) [11] es otro prototipo que se especializa por poseer un aprendizaje o entrenamiento hibrido. El diseño de estas redes se caracteriza por la presencia de 3 capas: una de entrada, una única capa oculta y una capa de salida. La red RBF se distingue de la red MPL en que las neuronas de la capa oculta calculan la distancia euclídea entre el vector de pesos y la entrada, sobre esa distancia se aplica una función de tipo radial con forma Gaussiana y para el aprendizaje de la capa oculta están estipulados diversos procedimientos, siendo muy popular el de k-means que es un algoritmo no supervisado, se establecen los valores de los centros, se precisan las anchuras (parámetros de la función Gaussiana) de cada una de las neuronas y por último, se entrena la capa de salida. En [12] se utiliza una red neuronal de este tipo para desarrollar un IDS; para esto, se analizó la información obtenido de KDDcup'99, una base de datos con información de paquetes de una red informática entre los que se encuentran diferentes tipos de ataques. En esta investigación se logró un desempeño del 89% en la clasificación correcta de las amenazas.

Por otro lado, la red Growing Grid (GG) [13] es otro de los más destacados modelos de IDS basados en redes neuronales, es una red creciente y rectangular con determinada cantidad de neuronas y que en cada número de iteraciones añade una fila o columna a la estructura hasta alcanzar la cantidad de neuronas necesarias según el criterio definido por el programador. La red clasifica cada acción como normal o anormal. En [14] se utilizan Growing Grids aplicadas a la detección de intrusos, tomando datos de las conexiones realizadas entre los clientes y los servidores, tales como el tiempo de conexión, el número de bytes enviados y el promedio de paquetes por segundo. Además, los resultados obtenidos se comparan con los de una red neuronal de tipo SOM.

4. DISCUSIÓN

A.Ventajas de los IDS basados en redes neuronales artificiales

Los SE son una buena herramienta para detectar cuando el sistema está siendo atacado, pero para que el nivel de efectividad pueda ser alto su base de conocimientos debe ser actualizada constantemente, además, se debe tener en cuenta que es imposible programar este sistema para que identifique todas y cada una de las variantes que un atacante puede usar. La primer ventaja del uso de RNA en la detección de intrusos es la flexibilidad y la capacidad de generalización que éstas pueden proveer, a diferencia de la rigidez propia de los sistemas basados en reglas, una RNA es capaz de analizar los datos de la red, incluso si los datos están incompletos, distorsionados o poseen variantes de los ataques [10]. Sin embargo, un hecho que es importante destacar es que la capacidad de generalización de la RNA depende de forma crítica en el diseño de la red y en los datos de entrenamiento, por lo tanto se requiere de la selección cuidadosa tanto de un conjunto de datos como de métodos comprobados para escoger la estructura correcta de la red; de igual forma los SE requieren de la construcción de un conjunto de reglas, tarea bastante compleja en muchas situaciones. Otra ventaja muy importante es que se puede hacer un análisis de los datos en una forma no lineal, habilidad requerida para detectar ataques que varían con el tiempo.

Siempre que se realice un correcto proceso de entrenamiento y diseño, la velocidad de las RNA puede superar en muchas ocasiones a los sistemas expertos, ya que no requieren inspeccionar bases de datos con miles de reglas. Como la salida es expresada en forma de una probabilidad, la RNA provee una capacidad predictiva de los casos de abuso del sistema. Un sistema para la detección del mal uso basado en redes neuronales identificará la probabilidad de que un evento en particular o una serie de eventos sea un indicativo de un ataque contra el sistema.

B. Desventajas de los IDS basados en redes neuronales artificiales

Es importante mencionar que, si bien las ventajas son muy significativas, el uso de RNA en los IDS también plantea inconvenientes a tener en cuenta. El consumo de recursos es alto en la etapa de entrenamiento, a diferencia de los sistemas basados en reglas que no requieren entrenamiento y pueden comenzar a funcionar de inmediato. Sin embargo, estos entrenamientos no deben hacerse constantemente, por lo general se hacen para extensos períodos de tiempo.

La desventaja más significativa es el hecho de que una RNA es muy sensible a los datos de entrenamiento, estos datos deben ser recolectados y preparados muy cuidadosamente ya que son precisamente los que le dan a la red la capacidad de interpretar correctamente la información que debe clasificar, en este caso los datos deben incluir la mayor cantidad de ataques que sea posible para ser efectiva. Si después del entrenamiento llega un dato totalmente diferente a los de éste, entonces la red probablemente no sea capaz de clasificarlo en forma correcta.

Otro inconveniente importante de las redes neuronales, es su característica conocida como caja negra, la cual consiste en que luego del proceso de entrenamiento es difícil modificar la estructura interna de la red neuronal para realizar pequeños ajustes. Por ejemplo, en el caso de que se detecte un nuevo tipo de ataque, en esta situación sería necesario volver a realizar todo el entrenamiento con el nuevo ataque encontrado, siempre y cuando sea totalmente diferente a los que ya pertenecían al conjunto de aprendizaje.

5. MÉTODO UTILIZADO

A. Variables de entrada

Los datos de entrada para el prototipo propuesto han sido tomados del estudio que realizó el MIT Lincoln Laboratory entre 1998 y 1999 con el patrocinio de la Agencia de Investigación de

Proyectos Avanzado de Defensa (DARPA) de Estados Unidos, para la evaluación de los IDS que existían; en ese entonces [15], se hicieron numerosas capturas de tráfico de red, el cual contiene diferentes tipos de ataques.

Para determinar cuáles paquetes contienen ataques y cuáles representan tráfico normal se ha usado Snort, el sistema de detección y prevención de intrusiones más ampliamente usado en la actualidad. Una vez separados, los paquetes han sido tratados por medio de Libpcap, librería que provee un marco de trabajo para el monitoreo de redes a bajo nivel, haciendo posible la extracción de las características que nos interesan de cada paquete.

Se decidió extraer cuatro características de cada paquete: la dirección IP de origen, dirección IP de destino, puerto de origen y puerto de destino; estos cuatro datos son la entrada de la RNA. Estos descriptores fueron elegidos principalmente porque logran conservar un compromiso entre desempeño y velocidad, mientras permiten identificar claramente las diferentes conexiones realizadas en la red. Los valores extraídos de los paquetes se agrupan en un vector, a la vez que se crea otro vector de correspondencia indicando si el paquete es normal o por el contrario se trata de un ataque. El vector de cuatro datos por paquete debe ser normalizado antes de ser pasado a la red neuronal, usando la ecuación (1),

$$x_{ij} = (x_{ij} - \min(x_j)) / (\max(x_j) - \min(x_j)) \qquad (1)$$

donde x_{ij} es el descriptor j del dato i, y $max(x_j)$, $min(x_j)$ $\boldsymbol{max(x_j), min(x_j)}$ son los valores máximos y mínimos de cada variable. Una vez se tienen todos los datos normalizados se puede empezar el proceso de entrenamiento.

B. Tipo de red neuronal

Nuestra red neuronal debe reconocer los patrones que le permitan clasificar los paquetes a partir de las características extraídas a cada uno de éstos, por esta razón, se decidió hacer uso del tipo de red neuronal conocido como Perceptrón Multicapa y el algoritmo de entrenamiento llamado Backpropagation, ya que esta combinación de red y algoritmo ha demostrado ser muy efectiva en la tarea de reconocimiento de patrones [2]. La red Perceptrón Multicapa usada posee una capa de entrada con un valor constante de cuatro neuronas para las cuatro características tomadas de cada paquete y una capa de salida con dos neuronas, en donde se activará la primera cuando un paquete se reconozca como parte de un ataque y se activará la segunda en el caso contrario. El número de neuronas en la capa oculta fue escogido por medio de un proceso de prueba, en el cual se contrastó el desempeño de la red con un número desde 5 hasta 50 neuronas, encontrando a 20 como el número que mejor desempeño otorga.
El algoritmo de Backpropagation requiere de una fase de entrenamiento en la cual se le pasa a la red un conjunto de ejemplos, es decir, vectores de características de un grupo de paquetes con su respectivo valor de salida, (1,0) cuando se trata de un ataque y (0,1) cuando se trata de tráfico de red normal. De esta forma se pueden modificar progresivamente los valores de los pesos de las conexiones de la red hasta que ésta cometa la mínima cantidad de errores posible en la correcta identificación del grupo de paquetes de ejemplo [1]. Posteriormente, terminada esta fase de entrenamiento, se procede a probar la efectividad de la red en la identificación de un nuevo conjunto de paquetes. La efectividad se define como la cantidad de paquetes identificados correctamente sobre la cantidad total de éstos.

6. EVALUACIÓN

A. Simulación IDS inteligente

El objetivo en esta etapa es utilizar los datos recolectados de la base de datos del DARPA para, en primer lugar, simular el aprendizaje de la red neuronal por medio de un conjunto de paquetes de entrenamiento. Este conjunto está conformado por 1000 paquetes, 500 de ellos como parte de tráfico normal y los demás como parte de ataques. De estos datos un subconjunto escogido de forma aleatoria es enviado a la red neuronal con su respectiva etiqueta, con el fin de modificar los pesos o sinapsis de la red y lograr así que ésta logre la capacidad de clasificación. Otro conjunto de paquetes conformado por 400

elementos, 200 de ellos ataques y los demás normales, es utilizado para llevar a cabo la prueba del funcionamiento de RNA. Cada cierto tiempo un subconjunto de estos datos es seleccionado al azar y enviado a la red neuronal con fin de determinar si el aprendizaje se está logrando de forma exitosa.

Se construyeron dos prototipos de software independientes para la implementación de la simulación. El primero de ellos es el que se encarga de escoger aleatoriamente los paquetes, ya sean del conjunto de entrenamiento o el de prueba y extraer la información contenida en estos paquetes por medio de Libpcap, y realizar la normalización de estas entradas, conformadas por los puertos y las direcciones IP de los paquetes. El otro prototipo, en el que se implementó la RNA y el algoritmo de Backpropagation, fue utilizado para la clasificación de los paquetes tanto de entrenamiento como de prueba. Se escogió una tasa de aprendizaje de 0.75 que disminuye en cada iteración. La función de activación escogida fue la sigmoide.

B. Resultados

El proceso de aprendizaje y comprobación fue realizado por iteraciones. En cada iteración, 500 datos de entrenamiento son escogidos aleatoriamente y pasados a la red neuronal para su clasificación y ajuste de los pesos; luego se comprueba el desempeño obtenido con los datos de prueba. En total 200 iteraciones fueron llevadas a cabo. En la Fig. 3 se puede observar el resultado logrado.

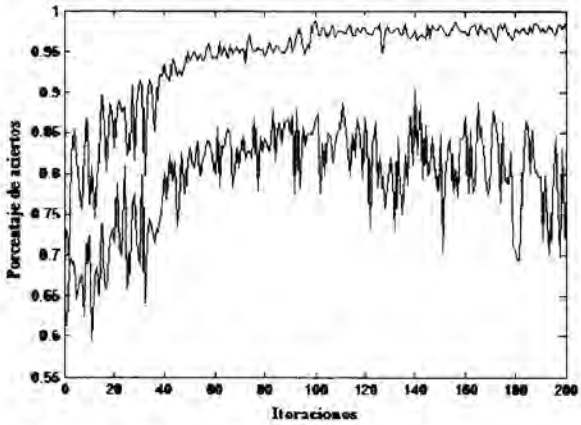

Fig. 3. Comportamiento de la RNA durante el entrenamiento (línea superior) y prueba (línea inferior).

La figura representa el desempeño de la red con respecto al número de iteraciones; la línea superior ilustra el porcentaje de clasificación correcta obtenido para los datos de entrenamiento, mientras que la línea inferior representa el porcentaje obtenido para los datos de prueba. En este punto es importante mencionar que existe un fenómeno conocido como sobre entrenamiento [2], que consiste en que entre más iteraciones se realicen, mejor es el desempeño de la red en la clasificación de los datos de entrenamiento, pero disminuye la capacidad de generalización, y por lo tanto el desempeño con los datos de prueba comienza a verse afectado. Este hecho, claramente evidenciado en el gráfico, requiere de la búsqueda del número de iteraciones para el cual se logra un máximo de rendimiento con los datos que sirven de prueba. Se puede observar este máximo cerca de la época 140, luego el rendimiento comienza a reducirse. En este punto, 91.5% de las paquetes de prueba fue clasificado correctamente. De éstos el 11% de los paquetes de tráfico normal fueron categorizados erróneamente como ataques, lo que se conoce como falsas alarmas. En total el sistema identificó correctamente el 90.04% de los paquetes.
En la tabla I se resumen los resultados:

TABLA I
RESULTADOS OBTENIDOS CON EL PROTOTIPO DE RED NEURONAL

ATAQUES CORRECTAMENTE IDENTIFICADOS	91.5%
FALSAS ALARMAS	11%
TOTAL PAQUETES	90.04%

Finalmente se decidió utilizar una técnica conocida como validación cruzada, que consiste básicamente en dividir el conjunto de datos de entrenamiento en n subconjuntos – 10 en nuestro caso–, y realizar el proceso de entrenamiento con n-1 subconjuntos dejando 1 para probar el desempeño (etapa de prueba). Este procedimiento se repite n veces utilizando cada vez un subconjunto diferente. Al final se obtiene un promedio del desempeño obtenido

en todas las iteraciones. El objetivo de esta técnica es determinar qué tan sensible es el algoritmo utilizado con respecto a los datos de entrenamiento que se utilizan en el proceso de aprendizaje.

TABLA II
RESULTADOS DE VALIDACIÓN CRUZADA

SUBCONJUNTO	PORCENTAJE
1	86%
2	90%
3	92%
4	95%
5	92%
6	95%
7	92%
8	92%
9	91%
10	89%
PROMEDIO	**91.4%**

En la tabla II se puede observar el resultado obtenido. El porcentaje de aciertos es muy similar en todas las iteraciones, con un promedio final cercano a lo que se obtuvo con los datos de prueba en la tabla I. Esto sugiere que los datos de entrenamiento fueron correctamente escogidos y se encuentran bien distribuidos aleatoriamente en los diferentes grupos. Además, el algoritmo de aprendizaje aquí utilizado demuestra un comportamiento estable con diferentes conjuntos de datos.

7. CONCLUSIONES

En este artículo se planteó un método para simular el comportamiento de un sistema inteligente para la detección de intrusos. Se utilizó una base de datos conformada por paquetes TCP/IP, algunos de los cuales eran maliciosos y otros normales, y se implementó un algoritmo de red neuronal conocido como Backpropagation, con el que fue posible simular un proceso de aprendizaje con el objetivo de que la RNA adquiriera la capacidad de clasificar dichos paquetes.

Los resultados obtenidos muestran que la clasificación se logró de forma exitosa obteniendo buenos resultados, especialmente en lo que a la identificación de paquetes maliciosos se refiere. Además estos resultados se lograron con un muy reducido número de características obtenidas de los paquetes, sólo 4, lo que favorece en gran medida la implementación de un sistema en tiempo real, gracias al menor tiempo de procesamiento requerido. Por otro lado, es importante notar que a pesar de los buenos resultados, con un 90.04% de correcta clasificación, y un 11% de falsas alarmas, para un sistema comercial y en producción se requiere avanzar mucho más en estos resultados, ya que aún se tendría un sistema con un rango bastante alto de vulnerabilidad.

Los antecedentes encontrados y la discusión realizada en torno a ellos, sugieren que los sistemas expertos y las redes neuronales por sí solos no proveen una solución completa para el problema de la detección de intrusiones, lo que podría derivar en propuestas de sistemas mixtos, que utilicen las mejores características de los dos paradigmas. También es muy importante, como trabajo futuro, determinar qué otros datos tomados de los paquetes pueden ser de utilidad para obtener mejores resultados, para lo cual será necesario investigar más a fondo los distintos protocolos de red. Se pretende implementar otros modelos de redes neuronales para determinar cuál muestra mejor desempeño en nuestro problema de interés.

REFERENCIAS

[1] Lahoz-Beltra, R. "Bioinformática: Simulación, vida artificial e inteligencia artificial". Editorial Díaz de Santos, 2004.
[2] Haykin, S. "Neural networks: A comprehensive Foundation". Prentice Hall, 1994.
[3] Anderson, J.P. "Computer Security Threat Monitoring and Surveillance". Technical Report, J.P. Anderson Company, Fort Washington, Pennsylvania. 1980.

[4] Denning, Dorothy. "An Intrusion-Detection Model". IEEE Transactions on Software Engineering, Vol. SE-13, No. 2. 1986.
[5] Cannady, J. "Artificial Neural Networks for Misuse Detection". Proceedings of the 1998 National Information Systems Security Conference, pp. 443-456. Arlington, USA. 1998.
[6] Zhang, C.; Jiang J. & Kamel, M. "Intrusion Detection using hierarchical neural networks". Pattern Recognition Letters, Volume 26, Issue 6, pp. 779-791. 2005.
[7] Ryan, J; Meng-Jang L. & Miikkulainen R. "Intrusion Detection with Neural Networks". The University of Texas. Austin, TX, USA. 1998.
[8] Wang, G.; Hao, J.; Ma, J. & Huang, L. "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering". Department of Information Systems, University of Hong Kong. 2010.
[9] Lichodzijewski, P.; Zincir-Heywood, A. & Heywood, M. "Host-Based Intrusion Detection Using Self-Organizing Maps". IEEE Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada. 2002.
[10] Wu, S. & Banzhaf, W. "The use of computational intelligence in intrusion detection systems: A review". Department of Computer Science, Memorial University of Newfoundland. Canada. 2008.
[11] Yang, Z.; Wei, X.; Bi, L.; Shi, D. & Li, H. "An Intrusion Detection System Based in RBF Neural Network". Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design Proceedings. pp. 873-875. Coventry, UK. 2005
[12] Bi, J.; Zhang Z. & Cheng X. "Intrusion Detection Based on RBF Neural Network". IEEC, pp.357-360. 2009 International Symposium on Information Engineering and Electronic Commerce. 2009.
[13] Marsland, S., Shapiro, J., & Nehmzow, U. "A self-organizing network that grows when required. Neural Networks", vol. 15, pp. 1041–1058. 2002.
[14] Mora, F. J.; Maciá F.; García, J.M. & Ramos, H. "Intrusion Detection System Based on Growing Grid Neural Networks". IEEE Mediterranean Electrotechnical, Benalmádena, Málaga. 2006.
[15] Haines, J; Lippman, R; Fried, D; Zissman, M; Tran, E. & Boswell, S. "1999 DARPA intrusion detection evaluation: design and procedures". MIT Lincoln Laboratory Technical Report, TR-1062, Massachusetts, USA. 2001.