Modelo de detección de intrusiones en sistemas de red, realizando selección de características con FDR y entrenamiento y clasificación con SOM[1]

Artículo de Investigación - Fecha de recepción: 8 de agosto de 2012 - Fecha de aceptación: 28 de septiembre de 2012

Emiro De la Hoz
Magíster en Ingeniería de Computadores y Redes, Corporación Universidad de la Costa - CUC. Barranquilla, Colombia, edelahoz@cuc.edu.co

Eduardo Miguel De la Hoz
Magíster en Ingeniería de Computadores y Redes, Corporación Universidad de la Costa - CUC. Barranquilla, Colombia, edelahoz6@cuc.edu.co

Andrés Ortiz
Doctor en Tecnologías de la Información y las Comunicaciones, Universidad de Málaga. Madrid, España, aortiz@ic.uma.es

Julio Ortega
Doctor en Tecnologías de la Información y las Comunicaciones, Universidad de Granada. Granada, España, julio@atc.ugr.es

RESUMEN

Los Sistemas de Detección de Intrusos (IDS, por sus siglas en inglés) comerciales actuales clasifican el tráfico de red, detectando conexiones normales e intrusiones, mediante la aplicación de métodos basados en firmas; ello conlleva problemas pues solo se detectan intrusiones previamente conocidas y existe desactualización periódica de la base de datos de firmas. En este artículo se evalúa la eficiencia de un modelo de detección de intrusiones de red propuesto, utilizando métricas de sensibilidad y especificidad, mediante un proceso de simulación que emplea el dataset NSL-KDD DARPA, seleccionando de éste las características más relevantes con FDR y entrenando una red neuronal que haga uso de un algoritmo de aprendizaje no supervisado basado en mapas auto-organizativos, con el propósito de clasificar el tráfico de la red en conexiones normales y ataques, de forma automática. La simulación generó métricas de sensibilidad del 99,69% y de especificidad del 56,15% utilizando 20 y 15 características, respectivamente.

Palabras clave

IDS (Sistema de Detección de Intrusos), FDR (Razón Discriminante de Fisher), SOM (Mapas Auto-organizativos), dataset NSL-KDD DARPA.

1. Artículo derivado del proyecto de investigación titulado: *Preprocesamiento en sistemas de detección de intrusos (IDS) con Mapas Auto-organizativos (SOM)*, gestado desde la labor investigativa realizada a nivel de la maestría en Ingeniería de Computadores y Redes, Universidad Nueva Granada, España.

Intrusion detection model in network systems, making feature selection with FDR and classification-training stages with SOM

ABSTRACT

Current commercial IDSs classify network traffic, detecting both intrusions and normal connections by applying signature-based methods. This leads to problems since only intrusion detection previously known is detected and signature database is periodically outdated. This paper evaluates the efficiency of a proposed network intrusion detection model, using sensitivity and specificity metrics through a simulation process that uses the dataset NSL-KDD DARPA, selecting from this, the most relevant features with FDR and training a neural network that makes use of an unsupervised learning algorithm based on SOMs, in order to automatically classify network's traffic into normal and attack connections. Metrics generated by simulation were: sensitivity 99.69% and specificity 56.15%, using 20 and 15 features respectively.

Keywords

IDS (Intrusion Detection System), FDR (Fisher Discriminant Ratio), SOM (Self-Organizing Map), dataset NSL-KDD DARPA.

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

INTRODUCCIÓN

Las organizaciones requieren proteger la información contenida en sus redes informáticas; sin embargo, el hecho de que en la actualidad tanto usuarios internos como externos al contexto de la red puedan conectarse de forma local o remota, incrementa considerablemente la probabilidad de que ésta sea atacada, razón por la cual se han desarrollado diferentes herramientas y estrategias, tanto de hardware como de software, para detectar y prevenir accesos intrusivos a la red con intenciones maliciosas.

Para evitar ataques procedentes de fuentes externas existen cortafuegos (firewalls) y Redes Privadas Virtuales (VPNs); tales herramientas restringen el tráfico de servicios desconocidos, en el caso de los cortafuegos, mediante el bloqueo de puertos. Queda aún un hueco de seguridad desde el exterior y es encapsular los ataques en el tráfico de servicios permitidos por el dispositivo. Aunada a esta situación, estas herramientas no controlan los ataques que se generan desde el interior de la red. Para subsanar este inconveniente se han desarrollado Sistemas de Detección de Intrusos (IDS) que identifican tráfico malicioso en la red y Sistemas de Prevención de Intrusos (IPS) que una vez detectado y aprendido el ataque por parte del sistema, lo bloquean, documentan e incluso contrarrestan tomando represalias contra el posible atacante. En el presente artículo se mencionará IDS para referirse indistintamente a los sistemas de detección y prevención de intrusos.

Los IDS pueden detectar ataques con una metodología basada en firmas (comparando los ataques con una base de datos de firmas o rules) o con una metodología basada en anomalías (empleando un algoritmo de aprendizaje), los primeros se han implementado ampliamente en IDS comerciales, sin embargo presentan la limitante de no detectar ataques nuevos; los segundos detectan ataques nuevos con cierto porcentaje de exactitud. El compromiso de los investigadores ha sido elevar los niveles de exactitud en la detección. Para ello, inicialmente en contextos de simulación, los investigadores toman colecciones de datos o datasets y los someten a un proceso que implica varias fases: preprocesamiento, normalización, entrenamiento y clasificación, con el objeto de evaluar la eficiencia de los algoritmos de entrenamiento usados para desarrollar los IDS. Si los algoritmos generan altas métricas de eficiencia en la detección, podrían luego ser implementados en IDS reales. Por otra parte, cada ataque en la red es identificado como un registro de conexión del dataset, y cada registro posee un total de 41 características que posibilitarán la identificación del ataque.

Producto de las evaluaciones efectuadas, los investigadores han detectado que una variable que incide directamente en la eficiencia del algoritmo de aprendizaje seleccionado es la escogencia de las características que se van a evaluar durante la fase de preprocesamiento, debido a que la escogencia de la totalidad de características o algunas de ellas que no sean las apropiadas, generará largos tiempos de respuesta computacional, incidiendo negativamente en la evaluación final del algoritmo de aprendizaje. Se utilizan entonces algoritmos de reducción de características, durante la fase de preprocesamiento, que posibiliten la adecuada esco-

gencia de las características más relevantes que permitan diferenciar el tráfico normal del que se constituye como ataque. El problema radica en analizar qué métodos de reducción o extracción de características generan mejores resultados.

Para poder identificar con precisión la magnitud del problema y las posibles alternativas de solución, se deben abordar con detalle: la fundamentación referida a los Sistemas de Detección de Intrusos, las características inherentes a los dataset DARPA, las técnicas o algoritmos existentes en relación con la extracción de características, la funcionalidad de los SOM (mapas autorganizativos) y la integración de estos fundamentos en un proceso de reproducción de experiencias de simulación.

Como estrategia de investigación se implementaron varios escenarios de simulación, los cuales comprendieron tres fases (entrenamiento, clasificación y métricas); se realizó variación de la cantidad de características por implementar en cada simulación. En la fase de entrenamiento se cargó un dataset de DARPA (con conexiones normales y ataques), luego se efectuó la reducción de características aplicando la tasa discriminante de Fisher (FDR), seleccionándolas por orden de relevancia y finalmente se realiza el entrenamiento del SOM. En la fase de clasificación se carga otro dataset de DARPA, diferente del conjunto de datos de entrenamiento, se reducen las características de la nueva colección de datos usando FDR, teniendo en cuenta la misma cantidad de características de la fase de entrenamiento y, por último, se clasifican los datos, basándose en el mapa generado en el proceso de entrenamiento y la data correspondiente al nuevo dataset cargado. En la fase final se calculan las métricas de desempeño de sensibilidad y especificidad que van a indicar la eficiencia del modelo planteado.

El tema de estudio genera un positivo impacto científico, mediante la implementación del modelo propuesto de detección de intrusiones en sistemas de red, en IDS comerciales, lo que posibilitará y favorecerá los procesos de detección y clasificación de tráfico normal y anómalo, de forma no supervisada, suprimiendo la necesidad de una actualización manual de la base de datos de ataques, por parte de un especialista humano. El modelo entrena una red neuronal que posteriormente, de forma automática, efectúa el proceso de clasificación de flujos de datos. Tal red neuronal es capaz de identificar el tipo de tráfico, independientemente de si se generan nuevos tipos de ataques.

Dado que a la fecha no se han implementado IDS a nivel comercial que hagan uso de técnicas de detección de intrusos sin supervisión (o basadas en anomalías), esta investigación ha tomado como referentes algunos antecedentes en materia de implementación comercial de IDS que utilizan técnicas de detección de intrusos basadas en firmas (o basadas en abuso) tales como: Snort [1], NFR (Network Flight Recorder) [2], NSM (Network Security Monitor) [3], Cisco Intrusion Detection (NetRanger) [4] y RealSecure [5]. Por otra parte, algunos referentes en materia de investigación en cuanto al uso de técnicas de inteligencia artificial en la detección de intrusiones pueden ser consultados en [6] y [7].

El propósito inicial de esta investigación es

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

evaluar un modelo de detección de intrusiones en sistemas de red propuesto, el cual luego de su afinamiento futuro —se vislumbra— podrá ser implementado en IDS comerciales capaces de clasificar flujos de datos de conexiones normales y ataques, de forma no supervisada.

SISTEMAS DE DETECCIÓN DE INTRUSOS

Los IDS son eficaces herramientas de protección de datos que complementan en gran medida el uso de otras técnicas de seguridad. Los sistemas informáticos pueden estar expuestos a diversidad de ataques que viajan por la red informática, en forma de conexiones.

Las dos finalidades primordiales de los IDS son la prevención y la reacción. La primera implica la "escucha" del tráfico en la red o de una computadora en particular, con la intención de identificar el ataque, aplicando técnicas de reconocimiento de patrones o técnicas inteligentes basadas en modelos estadísticos y soportados en redes neuronales, informando en tiempo real al sistema sobre los intentos de ataques o de actividades sospechosas. Como complemento a ello la segunda finalidad hace posible la elaboración de respuestas defensivas implementadas a través de scripts, ante la materialización de los ataques. Esto es posible mediante el análisis de archivos logs en los sistemas.

Se ha tratado de estandarizar la arquitectura de los IDS inicialmente mediante las propuestas efectuadas por CIDF (Marco común de Detección de Intrusiones) definido en [8], [9] y [10], y autopost de AusCERT (Equipo australiano de Respuestas a Emergencias en Computación) que puede ser consultado en [11]; ambos proyectos culminaron infructuosamente, razón por la cual los esfuerzos de estandarización actuales se han encaminado por IDWG (Grupo de Trabajo en Detección de Intrusiones, presentado por el Grupo de Trabajo de Ingeniería en Internet - IETF) que se define en [12] y CVE (Exposiciones y Vulnerabilidades Comunes) que puede ser consultado en [13].

Los IDS se clasifican de acuerdo con los criterios de enfoque o tipo de análisis, origen de los datos o fuentes de información, por su estructura y según su respuesta o comportamiento. Apréciese el esquema de clasificación en la Fig. 1. Información detallada sobre estos sistemas puede ser consultada en [14], [15] y [16].

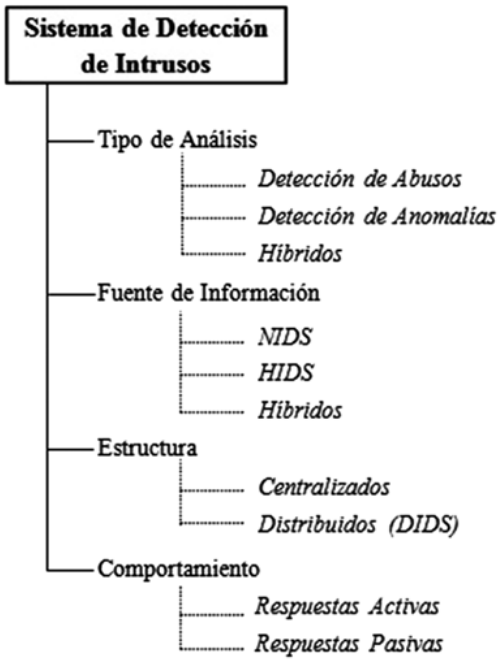

Fig. 1 Clasificación de los IDS

En relación con el enfoque o tipo de análisis, los IDSs se clasifican en detección de intrusos basada en abusos (Misuse-based

Intrusion Detection) y detección de intrusos basada en anomalías (Anomaly-based Intrusion Detection); en la Fig. 2 se aprecia su arquitectura funcional. Mayor información en relación con los algoritmos utilizados como estrategia de análisis, por cada uno de estos enfoques, puede ser consultada en [17] y [18].

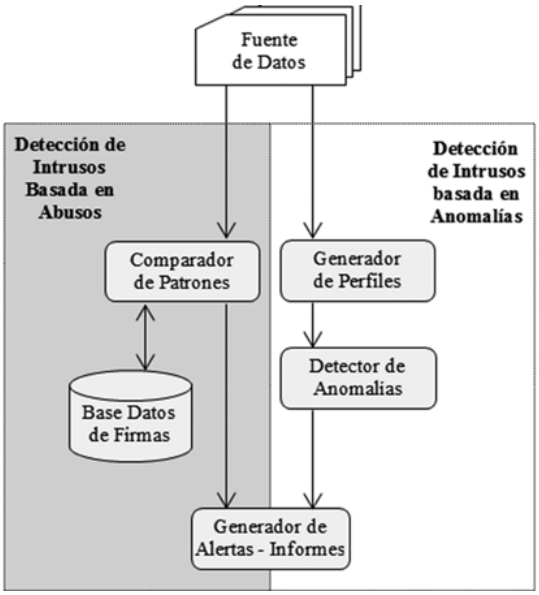

Fig. 2 Arquitectura de los IDS de acuerdo con el enfoque

La detección de intrusos basada en abusos monitoriza las actividades que ocurren en un sistema y las compara con una base de datos de firmas de ataques predefinida, generando una alerta si la actividad es identificada como ataque. Según [6] esta técnica es muy utilizada en productos comerciales, debido a su previsibilidad y alta precisión; sin embargo, para que el método sea efectivo, es necesario mantener actualizada la base de datos de firmas. La deficiencia de este método es que no identifica nuevos ataques inexistentes en dicha base de datos. Ejemplos de IDS que emplean las técnicas de detección de intrusos basadas en abusos, ya mencionados anteriormente, son: Snort [1], NFR (Network Flight Recorder) [2], NSM (Network Security Monitor) [3], Cisco Intrusion Detection (NetRanger) [4] y RealSecure [5].

La detección de intrusos basada en anomalías funciona asumiendo que los ataques son diferentes a la actividad normal; se puede llegar a esta inferencia luego de un proceso de entrenamiento, en el cual se identificará "¿qué se considera como actividad normal?", analizando comportamientos inusuales tanto en los host como en el tráfico de la red. Para ello se construyen perfiles generados a partir del análisis de asociación de patrones; estos perfiles representan el comportamiento normal de los usuarios, hosts o conexiones de red. Las medidas y técnicas comúnmente utilizadas en los IDS para la detección de anomalías son: la detección de umbral y el uso de medidas estadísticas. Las técnicas de detección de anomalías como la aplicación de IDES (Sistema Experto de Detección de Intrusos), redes neuronales, el uso de algoritmos genéticos, la modelación de un sistema inmune y NIDES (Next-generation Intrusion Detection Expert), no son empleadas en la actualidad en los IDS con fines comerciales debido a que aún siguen siendo objeto de investigación.

Ataques

Son métodos mediante los cuales se intenta tomar el control de un sistema informático con el objeto de dañarlo o desestabilizarlo, haciendo uso de aplicaciones especialmente diseñadas para ello. Se agrupan en cuatro categorías principales: DoS (Denial Of Service - Ataques de Denegación de Servicios),

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

R2L (Remote to Local - Ataques de acceso Remoto a Local), U2R (User to Root - Ataques de Usuario a Súper usuario) y Probing (Sondeo de redes).

Los ataques de DoS propician la pérdida de conectividad de un sistema de cómputo o red informática, debido a la sobrecarga de los recursos computacionales (por ejemplo, ancho de banda) de la red víctima. Durante este tipo de ataques se saturan los puertos de comunicación con excesivo flujo de datos, de tal forma que la sobrecarga del sistema haga imposible la correcta prestación del servicio, denegando las diferentes peticiones efectuadas por los clientes que las solicitan. Los ataques del tipo DoS son: apache2, back, land, mailbomb, neptune, pod, processtable, smurf, teardrop y udpstorm.

Los ataques R2L se producen cuando un atacante que no posee cuenta de usuario en una máquina remota, logra autenticarse como súper usuario (root) o como usuario con restricciones, en tal máquina, accediendo a ella a través de la web. Los ataques de la categoría R2L son: ftp_write, guess_passwd, httptunnel, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, worm, xlock y xsnoop.

Los ataques de U2R se producen cuando un atacante que ya dispone de una cuenta en un sistema informático, obtiene mayores privilegios de los inicialmente establecidos para él. Esto sucede debido a las vulneraciones existentes en los sistemas operativos o a la previa instalación de programas espías que posibiliten el acceso intrusivo. Los ataques del tipo U2R son: buffer_overflow, loadmodule, perl, ps, rootkit, sqlattack y xterm.

Los ataques Probing escanean redes de datos con el objeto de identificar direcciones IP válidas y recopilar información acerca de ellas. Este tipo de ataque busca detectar qué servicios se ofrecen y cuáles son los sistemas operativos que utilizan. A partir de esto el atacante identifica una lista de vulnerabilidades potenciales para lanzar ataques tanto a los servicios como a las respectivas máquinas sobre las que éstos se ejecutan. Los ataques categorizados como Probing son: ipsweep, mscan, nmap, portsweep, saint y satan.

Cualquiera de los ataques mencionados en las categorías anteriores implican el establecimiento de conexiones y cada una de éstas genera un registro que ocupa 100 bytes y está compuesto por 41 atributos. Tales atributos permiten identificar si la conexión es normal o si es algún tipo de ataque.

Atributos o características

Existe una clasificación de los atributos en relación con el tipo de evaluación que posibilitan las conexiones: Los “atributos de contenido” permiten evaluar el número de intentos de acceso fallidos; los atributos de “mismo host” tienen en cuenta solo las conexiones en los dos últimos segundos que tengan el mismo destino que la conexión actual y estadísticas relacionadas con el protocolo y los servicios; los atributos de “mismo servicio” examinan solo las conexiones en los dos últimos segundos que tienen el mismo servicio que la conexión actual. Tanto los atributos de “mismo host” como los de “mismo servicio” permiten la evaluación del tráfico de las conexiones en el tiempo.

Dado que los ataques de Probing escanean los puertos con un intervalo de tiempo mucho mayor de dos segundos (posiblemente una vez por minuto). Se debe hacer una clasificación de los registros por host de destino, produciendo una serie de atributos denominados "tráfico basado en host".

Los ataques de las categorías R2L y U2R generan registros de conexión en los cuales los atributos no generan patrones secuenciales frecuentes, debido a que tales tipos de ataques solo requieren de una única conexión; contrario a esto, los ataques de las categorías DoS y Probing requieren muchas conexiones con un mismo host en un período muy corto de tiempo, lo que permite evaluar patrones secuenciales frecuentes.

En las Tablas I, II y III se aprecia una lista de atributos clasificados por atributos básicos, especiales y aquellos que hacen posible la evaluación con una ventana de tiempo de dos segundos.

Tabla I
Atributos básicos de las conexiones

Atributo	Descripción	Tipo
Duration	Longitud (número de segundos) de la conexión.	Continuo
protocol_type	Tipo de protocolo (tcp...).	Discreto
Service	Tipo de servicio de destino (HTTP, Telnet, SMTP...).	Discreto
src_bytes	Número de bytes de datos de fuente a destino.	Continuo
dst_bytes	Número de bytes de datos de destino a la fuente.	Continuo
Flag	Estado de la conexión (SF, SI, REJ...).	Discreto
Land	1 si la conexión corresponde mismo host/puerto; 0 de otro modo.	Discreto
wrong_fragment	Número de fragmentos erróneos.	Continuo
Urgent	Número de paquetes urgentes.	Continuo

Tabla II
Atributos especiales de las conexiones

Atributo	Descripción	Tipo
Hot	Número de indicadores "hot".	Continuo
num_failed_logins	Número de intentos de acceso fallidos.	Continuo
logged_in	1 si acceso exitoso; 0 de otro modo.	Discreto
num_compromised	Número de condiciones "sospechosas".	Continuo
root_shell	1 si se obtiene superusuario para acceso a root; 0 de otro modo.	Discreto
su_attempted	1 si se intenta el comando "su root"; 0 de otro modo.	Discreto
num_root	Número de accesos a root.	Continuo
num_file_creations	Número de operaciones de creación de ficheros.	Continuo
num_shells	Número de shell prompts.	Continuo
num_access_files	Número de operaciones de control de acceso a ficheros.	Continuo
num_outbound_cmds	Número de comandos de salida en una sesión ftp.	Continuo
is_hot_login	1 si el login pertenece a la lista "hot"; 0 de otro modo.	Discreto
is_guest_login	1 si el acceso es un "guest" login; 0 de otro modo.	Discreto

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

Tabla III
Atributos con ventana de dos segundos

Atributo	Descripción	Tipo
Count	Número de conexiones a la misma máquina que la conexión actual en los dos últimos segundos.	Continuo
Los siguientes atributos se refieren a las conexiones de mismo host		
serror_rate	Porcentajes de conexiones que tienen errores "SYN".	Continuo
rerror_rate	Porcentaje de conexiones que tienen errores "REJ".	Continuo
same_srv_rate	Porcentaje de conexiones con el mismo servicio.	Continuo
diff_srv_rate	Porcentaje de conexiones con diferentes servicios.	Continuo
srv_count	Número de conexiones al mismo servicio que la conexión actual en los dos últimos segundos.	Continuo
Los siguientes atributos se refieren a las conexiones de mismo servicio		
srv_serror_rate	Porcentaje de conexiones que tienen errores "SYN".	Continuo
srv_rerror_rate	Porcentaje de conexiones que tienen errores "REJ".	Continuo
srv_diff_host_rate	Porcentaje de conexiones a diferentes hosts.	Continuo

Los atributos anteriormente descritos pueden ser de dos tipos: continuos o numéricos (toman valores reales o enteros) y discretos o simbólicos (toman valores a partir de una lista especificada). Por ejemplo: el atributo "protocol_type" toma los valores tcp, udp e icmp; el atributo "service" toma los valores http, mtp, smtp, finger, domain, etc; el atributo "flag" toma los valores SF, S1, REJ, S2, etc. y el atributo "class" toma los valores back, buffer_overflow, ftp_write, guess_passwd, etc.

Para efectuar la detección de intrusiones se requiere evaluar las diferentes características o atributos que constituyen a las respectivas conexiones que hacen referencia a los ataques. Un IDS 100% efectivo no existe en la actualidad; para determinar su eficiencia se deben valorar diferentes métricas.

Métricas de desempeño

La *sensibilidad* (sensitivity) y *especificidad* (specificity) son medidas estadísticas del desempeño de una prueba de clasificación binaria, también conocida en las estadísticas como la función de clasificación. La **sensibilidad** (también llamada tasa de recuperación - recall rate) mide la proporción de "verdaderos positivos" que son correctamente identificados como tales. La **especificidad** mide la proporción de "verdaderos negativos" que se han identificado correctamente.

Un predictor perfecto sería descrito con un 100% de sensibilidad y un 100% de especificidad, sin embargo teóricamente cualquier predictor poseerá un error de límite (bound) mínimo, conocido como la tasa de error de Bayes.

Para el proceso de clasificación de tráfico "normal" y "ataques", es necesario evaluar las siguientes métricas:

- Verdaderos Positivos (VP): ataque correctamente identificado como ataque.
- Falsos Positivos (FP): tráfico normal

identificado incorrectamente como ataque.

- Verdaderos Negativos (VN): tráfico normal correctamente identificado como tráfico normal.
- Falso Negativo (FN): ataque identificado incorrectamente como tráfico normal.

A partir de lo anterior, entendiendo que la "sensibilidad" es la capacidad de una prueba para identificar resultados verdaderos positivos y que la "especificidad" se refiere a la capacidad de la prueba para identificar los resultados negativos, se tiene que:

$$Sensibilidad = \frac{VP}{VP + FN} \quad (1)$$

$$Especificidad = \frac{VN}{VN + FP} \quad (2)$$

Si una prueba tiene una "sensibilidad" del 100% significa que la prueba reconoce todos los verdaderos positivos, es decir, todos los ataques se detectan efectivamente como ataques. En contraste, una prueba que tiene un 100% de "especificidad", detecta todos los verdaderos negativos, es decir, todo el tráfico normal es correctamente identificado como tal.

Por otra parte, en los sistemas de medición la "exactitud" (accuracy) [19] es el grado de cercanía de las mediciones de una cantidad al valor de la magnitud real y la "precisión, reproducibilidad o repetibilidad" (precision) [19] es el grado en que las mediciones repetidas en condiciones iguales muestran los mismos resultados. Un sistema de medición se denomina válido si es a la vez exacto y preciso, sin embargo un sistema de medición puede ser exacto pero no preciso, preciso pero no exacto, ninguno de los dos, o ambos. La Fig. 3 describe estas dos métricas.

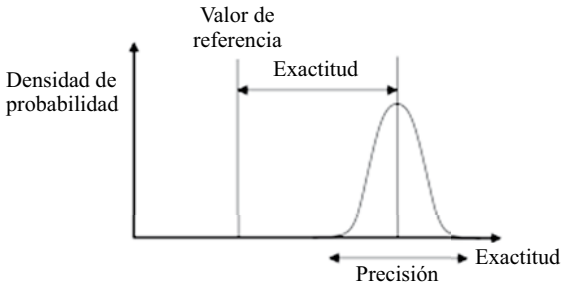

Fig. 3. Exactitud y precisión

La exactitud es la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos) en la población. Una exactitud del 100% significa que los valores medidos son exactamente los mismos que los valores dados. La exactitud es definida a partir de (3).

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN} \quad (3)$$

Por otra parte, el valor de la precisión o valor predictivo positivo se define como la proporción de verdaderos positivos contra todos los resultados positivos, definida en (4).

$$Precisión = \frac{VP}{VP + FP} \quad (4)$$

En la Tabla IV se muestra la relación existente entre las métricas de desempeño de los sistemas de clasificación, de acuerdo con los resultados y condiciones.

Se siguen valorando diferentes metodologías y técnicas con el objeto de desarrollar

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

Tabla IV
Relación de métricas de desempeño de un clasificador binario

		Condición (según lo determinado por el "Gold Standard")		
		Condición **Positiva**	Condición **Negativa**	
Resultado de la prueba	Resultados **Positivos** de la prueba	**Verdadero Positivo**	Falso Positivo (error tipo I)	**Valor predictivo positivo (precisión)=** $\frac{\sum VerdaderosPositivos}{\sum ResultadosPositivos}$
	Resultados **Negativos** de la prueba	Falso Negativo (error tipo II)	**Verdadero Negativo**	**Valor predictivo negativo=** $\frac{\sum VerdaderosNegativos}{\sum ResultadosNegativos}$
		Sensibilidad= $\frac{\sum VerdaderosPositivos}{\sum CondiciónPositiva}$	**Especificidad=** $\frac{\sum VerdaderosNegativos}{\sum CondiciónNegativa}$	**Exactitud=** $\frac{\sum Verdaderos}{\sum ResultadosVerdaderosyNegativos}$

una solución IDS cada vez más eficiente; para ello se requiere de un ambiente que permita simular el tráfico de red de la forma más real posible. Razón por la cual el MIT (Instituto Tecnológico de Massachusetts) y DARPA (la Agencia de Proyectos de Investigación Avanzada de Defensa) han simulado tal escenario, alimentando colecciones de datos, con el propósito de dotar a los investigadores de una base de datos de tráfico de red, que sirva de insumo para el desarrollo de investigaciones en el ámbito de la detección y prevención de intrusos.

EL DATASET DARPA

El Grupo de Tecnología de Sistemas de Información (IST), del Laboratorio Lincoln del Instituto Tecnológico de Massachusetts LL-MIT, con la cooperación de la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA ITO) y el Laboratorio de Investigación de las Fuerzas Aéreas (AFRL/SNHS), recopiló el primer dataset que contiene tráfico de red con una variada colección de conexiones. El dataset se utiliza para la evaluación de la eficiencia de los sistemas de detección de intrusos en redes informáticas. Los criterios medibles son la probabilidad de detección y la probabilidad de falsas alarmas del respectivo sistema testeado.

Los dataset publicados por LL-MIT en su web oficial, son los resultados de las evaluaciones en detección de intrusiones efectuada por DARPA en 1998 y 1999. También se encuentran experimentos dirigidos a escenarios específicos realizados en 2000. El LL-MIT distribuye libremente los dataset, la documentación, publicaciones, evaluaciones de resultados y herramientas de software relacionadas, disponibles en [20].

El dataset DARPA 1998 contiene un conjunto de ataques realistas, integrados a un conjunto de conexiones normales, lo cual suministra el insumo de datos que permite evaluar las falsas alarmas y las tasas de detección de IDS; para construir este dataset se efectuaron dos evaluaciones: una off-line

y otra en tiempo real. La primera consta de tráfico de red y logs de auditoría recogidos en una red de simulación, para la segunda se insertaron sistemas de detección de intrusión en el banco de pruebas de la red AFRL con la intensión de identificar sesiones de ataque en medio de actividades normales, en tiempo real.

La red física usada para la simulación incluye una subred interna y una externa separadas por un enrutador. La externa incluye dos estaciones de trabajo que simulan gateways en un Internet exterior virtual. Una estación de trabajo simula varias estaciones usando modificaciones del software cliente del kernel de Linux, proporcionados por el grupo Air Force ESC. Un gateway controla a cien estaciones y otro a miles de sitios web cuyo contenido se actualiza diariamente. La subred interna incluye máquinas críticas de muchos tipos (Linux, Solaris, Sun OS) y un gateway para muchas otras estaciones de trabajo internas. Los datos fueron recogidos desde un host interno que ejecuta Solaris y desde un sniffer externo.

En el escenario anteriormente descrito se enviaron emails, broadcasts, correo simple, y listas de servidores de dominio, así como tráfico ftp a través de la descarga de usuarios de variedad de código original y archivos de documentación de sitios ftp anónimos, tanto internos como externos. Se registró además la actividad de seis usuarios con identidades específicas profesionales (programador, secretario, administrador de sistema y gerente), los cuales diariamente realizaron sesiones telnet ejecutando tareas, accediendo con su identidad; algunas de esas acciones pueden considerarse actividades anómalas. Tales actividades comprenden: denegación de servicios, acceso desautorizado, transición desautorizada, obtención de privilegios de root por un usuario sin privilegios (para esto fue necesario efectuar vigilancia y testeo) y en general diferentes comportamientos anómalos de usuarios.

A partir de todos los datos recolectados se organizaron distintos subconjuntos de datos que componen el dataset DARPA 1998, tales como: datos de ejemplo, cuatro horas de subconjuntos de datos de entrenamiento, datos de entrenamiento (contienen siete semanas de ataques basados en red en medio de datos en segundo plano, normales) y datos de test (contiene dos semanas de ataques basados en red en medio de actividad normal en segundo plano).

El dataset DARPA 1999, al igual que su predecesor, está constituido por una evaluación off-line y una evaluación en tiempo real, basándose en los mismos principios que en el conjunto de datos del año anterior e incluyendo adicionalmente las siguientes características: ataques y tráfico desde ordenadores que ejecutan Windows NT, ataques en la red interna, archivos de sistema dump que proporcionan importantes componentes desde sistema de ficheros de cinco víctimas cada noche, incluyendo logs de auditoría de Windows NT y archivos de sniffing que proporcionan datos de sniffing de la red interna.

DARPA 1999 centra la evaluación de actividades de estaciones de trabajo UNIX, Windows NT y a partir de los siguientes eventos: Denegación de Servicios (DoS), Remoto a Local (R2L), Usuario a Root (U2R) y acceso desautorizado o modificación de datos en un host local o remoto.

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

Estos ataques ocurren en el contexto de uso normal de computadores y redes en una base militar. La organización de los datos utiliza un esquema similar al seguido por el dataset DARPA de 1998 con algunas modificaciones (no hay datos de ejemplo ni subconjuntos de datos de entrenamiento). Quedando el dataset DARPA 1999 constituido por: datos de entrenamiento (tres semanas de ataques, teniendo en cuenta que la primera y la tercera semana no contienen ataques, la segunda semana contiene un subconjunto selecto de ataques que van desde los ataques de 1998 a otros ataques nuevos), datos de test (dos semanas de ataques basados en red en medio de actividad normal en segundo plano).

En DARPA 2000 los datos se obtuvieron a partir de varios escenarios:

- *Escenario 1: LLDOS 1.0.* Este escenario está compuesto de múltiples sesiones de red y auditoría. Estas sesiones están agrupadas en cinco fases de ataques, en las cuales el atacante testea la red, interrumpe la vulnerabilidad de un host ejecutando Solaris, instala el software del troyano mstream DDoS, y lanza un ataque de DDoS en un servidor del sitio desde el host comprometido.
- *Escenario 2: LLDOS 2.0.2.* (igual que el anterior).
- *Conjunto de Datos de Ataques NT.* En enero del 2000 se realizó un experimento con un elevado nivel de auditoría de NT. En este dataset se presentan las trazas recogidas del tráfico de un día y el ataque que afecta a la máquina de NT. Este escenario está compuesto principalmente por: datos de auditoría de log de eventos NT, datos Tcpdump de la red externa, datos Tcpdump de la red interna y archivo con altos niveles de ataques reales.

El dataset NSL-KDD es una colección de datos construido con el objeto de solventar los problemas que presenta el conjunto KDD'99 [21], pese a no ser una representación perfecta de los datos reales, debido a que no contiene conjuntos de datos públicos de los IDS; sin embargo, demuestra mucha utilidad al ser aplicado como un conjunto de datos de referencia eficaz para ayudar a los investigadores en el proceso de comparación de diferentes métodos de detección de intrusos.

El número de registros que contiene el data set NSL-KDD es razonable, lo cual se constituye en una ventaja a la hora de realizar los experimentos con la colección de datos completa, para efectos de tiempo de procesamiento de la información, sin necesidad de elegir al azar a una pequeña porción de los datos, lo que consecuentemente conlleva a que los resultados de la evaluación de los trabajos de investigación lleguen a ser consistentes y comparables.

Las mejoras que presenta el NSL-KDD respecto a sus predecesores, son las siguientes:

- La colección de datos no incluye registros redundantes; por lo tanto los clasificadores no realizarán correcciones con tanta frecuencia.
- No existen registros duplicados en los conjuntos de pruebas propuestos; por lo tanto, el rendimiento del aprendizaje no

está sesgado por los métodos que tienen mejores tasas de detección en los registros frecuentes.

- El número de registros seleccionados de cada grupo de nivel de dificultad es inversamente proporcional al porcentaje de registros en el conjunto original de datos KDD. Como resultado, las tasas de clasificación de los distintos métodos de aprendizaje de máquinas varían en un rango más amplio, lo que hace que sea más eficiente para tener una evaluación precisa de las diferentes técnicas de aprendizaje.
- El número de registros tanto en la colección completa de datos como en el porcentaje del dataset es razonable, lo cual hace posible realizar los experimentos con el conjunto de datos completo sin necesidad de seleccionar al azar una pequeña porción de éste.

En [22] se encuentran los archivos de datos del NSL_KDD tanto en formato .txt como en formato .arff, cuya descripción se aprecia en la Tabla V. Este último formato "Attribute Relation File Format " es usado por compatibilidad con el software WEKA (Waikato Environment for Knowledge Analysis) [23], con el objeto de poder efectuar el análisis de datos de los dataset KDDTrain+ y KDDTest+. WEKA es un entorno de trabajo desarrollado por la Universidad de Waikato (Nueva Zelanda), construido en JAVA y con licenciamiento GPL, que se utiliza para procesos de experimentación de análisis de datos que hagan posible la aplicación, análisis y evaluación, sobre un dataset empleando técnicas relativas al aprendizaje automático.

Tabla V
Archivos del dataset DARPA NSL-KDD

Archivo	Descripción
KDDTrain+.arff	El conjunto de datos completo para el entrenamiento (train NSL-KDD), con etiquetas binarias y en formato ARFF.
KDDTrain+.txt	El conjunto de datos completo para el entrenamiento (train NSL-KDD), incluyendo etiquetas de tipos de ataques y el nivel de dificultad, en formato CSV.
KDDTrain+_20Percent.arff	Un subconjunto del 20% del archivo KDDTrain+.arff
KDDTrain+_20Percent.txt	Un subconjunto del 20% del archivo KDDTrain+.txt
KDDTest+.arff	El conjunto de datos completo para el test con etiquetas binarias y en formato ARFF.
KDDTest+.txt	El conjunto de datos completo para el test, incluyendo etiquetas de tipos de ataques y el nivel de dificultad, en formato CSV.
KDDTest-21.arff	Un subconjunto del archivo KDDTest+.arff el cual no contiene registros con el nivel de dificultad 21 de un total de 21.
KDDTest-21.txt	Un subconjunto del archivo KDDTest+.txt el cual no contiene registros con el nivel de dificultad 21 de un total de 21.

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

FASES DEL PROCESO DE SIMULACIÓN DE DETECCIÓN DE INTRUSIONES

La eficacia del proceso de detección del tráfico malicioso en una red informática, mediante la aplicación de un IDS que utilice técnicas de reducción de características, algoritmos de aprendizaje de máquina y detección de tráfico anómalo, es susceptible de ser evaluada mediante simulación de laboratorio. Ello requiere de la ejecución de varias fases: escogencia de la colección de datos (dataset), preprocesamiento, normalización, entrenamiento (training) y clasificación. La Fig. 4 ilustra dichas fases.

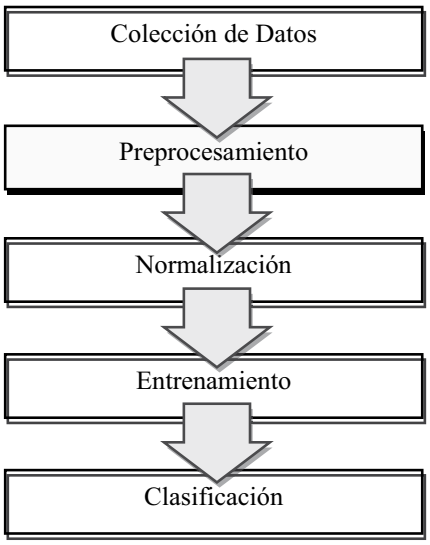

Fig. 4. Fases del proceso de simulación de detección de intrusiones

Fase de elección de la colección de datos

En esta fase inicial se debe seleccionar la colección de datos que se va a usar para las fases subsiguientes. Aunque existe una amplia variedad de datasets los investigadores comúnmente se han decantado por el uso de DARPA NSL-KDD, por las ventajas que ofrece en relación con la variedad y depuración de sus datos con respecto a otros dataset de su misma familia y de otras organizaciones. La Tabla VI muestra un listado de los dataset más destacados en procesos de simulación de sistemas de detección de intrusiones.

Fase de preprocesamiento

Los datos procedentes del dataset deben estar en el rango de [0 a 1] o de [-1 a 1]. Sin embargo no lo están, debido a que todas las conexiones en sus 41 características poseen valores continuos, discretos o simbólicos y en diferentes rangos de significancia. Con el propósito de estandarizar dichos valores para que puedan ser eficazmente procesados por los algoritmos de aprendizaje de máquina, se debe hacer un preprocesamiento y normalización de los datos contenidos en las conexiones.

Para la conversión de los símbolos en formato numérico, a cada símbolo se asigna un código entero. Por ejemplo, en el caso de la característica protocol_type, se asigna “0” a tcp, “1” a udp y “2” a icmp. De forma similar los nombres de ataque son mapeados asignando valores enteros a las cinco categorías así: “0” para tráfico normal, “1” para el ataque de sondeo (probe), “2” para la Denegación de Servicios (DoS), “3” para U2R y “4” para R2L.

Por otra parte, debido a que existen características cuyos valores se extienden por un rango de números enteros muy grande, es decir, src_bytes toma valores entre [0 y 1.3 billones] igual que dst_bytes. Se aplica entonces una escala logarítmica (de base 10) a

TABLA VI
DATASETS UTILIZADOS EN PROCESOS DE SIMULACIÓN DE SISTEMAS DE DETECCIÓN DE INTRUSIONES

Dataset	Patrocinadores - Miembros
Dataset DARPA	• IST-LLMIT (Grupo de Tecnologías de Sistemas de Información - Laboratorio del Instituto de Tecnología de Massachusetts). • DARPA ITO (Agencia de Proyectos de Investigación Avanzada de Defensa - Oficina de Tecnología de la Información). • AFRL/SNHS (Laboratorio de Investigación de las Fuerzas Aéreas).
Datasets USC/ISI ANT Programa PREDICT (Repositorio de Protección para la Defensa de la infraestructura frente a las amenazas informáticas). Proyecto LANDER.	• ANT (Grupo de Investigación de Análisis de Tráfico de Red). • ISI (Instituto de Ciencias de la Información). • USC (Universidad del Sur de California). • Departamento de Ciencias Computacionales de la Universidad Estatal de Colorado. • Departamento de Ingeniería Eléctrica de USC. • Servicios de Tecnologías de la Información de la USC.
Datasets CAIDA Asociación Cooperativa para el Análisis de Datos en Internet	Patrocinadores: ARIN (American Registry for Internet Numbers), CISCO, Endance Measurement Systems, U.S. Departament of Homeland Security, NSF (National Science Fundation). Miembros: Digital Envoy, Intel, NTT (Nippon Telegraph and Telephone Corporation), Ripe NCC, University of California San Diego.
Datasets CRAWDAD Comunidad de recurso para archivar datos inalámbricos en Dartmouth	• ACM SIGMOBILE. • Intel Corporation. • Fundación Nacional de Ciencias.
Dataset DRDC Defense Research and Development Canada	• Sección de Operaciones de Información de Red (NIO) de la DRDC Ottawa, Canadá. • Red de Establecimiento para la Investigación y Defensa (DREnet).
NIST SAMATE Reference Dataset Project NIST : National Institute for Standard and Technology SAMATE : Software Assurance Metrics and Tools Evaluation	Departamento de Estado de EE.UU.
Virtual Dataset Repository	MERIT NETWORK INC. Programa PREDICT (Protected Repository for the Defense of Infrastructure against Cyber Threats).

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

estas características para reducir el rango de [0.0 a 9.14]. Todas las demás características son booleanas, en el rango de [0.0 a 1.0]. Por lo tanto, el escalado no es necesario para estos atributos.

En la fase de preprocesamiento también se debe identificar la técnica de reducción de características que se van a utilizar, debido a que no es conveniente efectuar el entrenamiento de la red con la totalidad de características, dado que ello podría ralentizar considerablemente el procesamiento, sin añadir una significativa exactitud en la clasificación del tráfico. Posteriormente se abordarán las técnicas de reducción de características más utilizadas en la actualidad.

Fase de normalización

Para normalizar los valores de las características, se requiere efectuar un análisis estadístico sobre los valores de cada una de ellas en función de los datos existentes en el dataset, teniendo en cuenta el valor máximo aceptable que se ha determinado para cada característica.

De acuerdo con los valores máximos y la fórmula enunciada a continuación, se calcula la normalización de los valores de las características en el intervalo de [0 a 1], así:

Si (f > MaxF) Nf = 1, Si no Nf = (f / MaxF)

donde:

F: Característica - f: Valor de la característica

MaxF: Máximo valor aceptable para la característica

Nf: Valor normalizado o escala de F.

Fase de entrenamiento

En esta fase se entrena la red neuronal a partir del algoritmo de aprendizaje seleccionado y tomando como insumo el archivo procedente del dataset para tal fin. Normalmente se usa un archivo que contiene una cantidad de registros equivalente al 20% del total de los datos contenidos en el dataset (KDDTrain+_20Percent).

Fase de clasificación

Una vez la red neuronal ha sido entrenada, se procede con la fase de clasificación en la cual, de forma autónoma, el algoritmo clasificador determina qué tráfico es normal y cuál es un ataque, efectuando la subsiguiente clasificación de cada una de las conexiones del dataset. Gracias a esto se podrá presentar la información de resumen del proceso, de forma estadística, mediante gráficos por estado (tráfico normal o ataques), agrupados por tipo de ataque y listando las métricas de desempeño para valorar la eficiencia del sistema.

Una vez la red neuronal está entrenada, se procede con la prueba, la cual se realiza con el 100% de los datos contenidos en el dataset; para ello usualmente se utiliza el dataset KDDTest+.

Dado que el énfasis de este artículo es la fase de preprocesamiento y en particular las técnicas de extracción de características, a continuación se abordarán éstas con mayor detalle, puntualizando en la razón discriminante de Fisher (FDR).

Técnicas de extracción de características

El proceso de extracción de características documentado en [24], implica el mapeado de un espacio multidimensional a un espacio de menos dimensiones. Esto significa que el espacio de características original es transformado mediante la aplicación de una técnica de reducción de características, por ejemplo utilizando la transformación lineal del Análisis de Componentes Principales PCA.

El proceso de extracción de características simplifica la cantidad de recursos necesarios para describir con precisión un amplio conjunto de datos. Lo que es necesario cuando se realiza un análisis de datos complejos, debido a que uno de los principales problemas del proceso de clasificación deriva del número de variables involucradas.

Cuando se evalúa un considerable número de variables consecuentemente se requiere una gran cantidad de memoria y potencia de cálculo, por ello es importante controlar la cantidad de características que participan en el proceso de clasificación.

La extracción de características, efectuando reducción de dimensionalidad, en relación con el número de características por evaluar, hace posible la construcción de combinaciones de variables que minimicen estos problemas y al mismo tiempo describan los datos con suficiente precisión. A continuación se presentan tres técnicas de reducción de la dimensionalidad: Razón Discriminante de Fisher (FDR - Fisher Discriminant Ratio), Análisis de Componentes Independientes (ICA - Independent Component Analysis) y el Análisis de Componentes Principales (PCA - Principal Component Analysis).

La razón discriminante de Fisher

Definida en [25], encuentra la matriz de transformación óptima preservando la mayor parte de la información que se puede utilizar, para discriminar entre las diferentes categorías. Por lo tanto, el análisis requiere que los datos tengan etiquetas de categoría, a fin de formular matemáticamente el procedimiento de optimización o reducción.

FDR, también conocido como Análisis de Discriminante Lineal (LDA), fue originalmente desarrollado en 1936 por R.A. Fisher, como un método clásico que se utiliza para la clasificación (es decir, con una variable categórica objetivo). Mayor información puede ser consultada en [26]-[29].

Es un método utilizado en la estadística para el reconocimiento de patrones y de aprendizaje de máquina, para encontrar una combinación lineal de las características que determinan o separan dos o más clases de objetos o acontecimientos. La combinación resultante puede ser utilizada como un clasificador lineal, o más frecuentemente, para la reducción de la dimensionalidad antes de la clasificación.

Este método requiere que se calcule el vector de medias y la matriz de covarianza para cada categoría y para el conjunto completo de datos (con todas las clases agrupadas). A partir de esto, se puede formular el criterio de optimización.

El numerador representa la covarianza de los datos de entrenamiento agrupados en el espacio de características transformado. El

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

denominador representa la covarianza promedio dentro de cada clase en el espacio de características transformado. Por lo tanto, el criterio de verdad trata de maximizar la "distancia" entre las clases, mientras se minimiza el "tamaño" de cada una de estas, al mismo tiempo. Este criterio garantiza conservar la mayor parte de la información discriminante en el espacio de características transformado. En (6) se muestra la fórmula que determinar FDR, a partir de los puntos de datos "x" en el espacio m-dimensional y suponiendo que éstos se originan a partir de dos clases. El objetivo es generar una función "y" como una combinación lineal de los componentes de las "x", de esta manera, se espera "exprimir" la información relacionada con la clasificación que reside en "x" en un número menor de características (en este caso una).

Consecuente con lo anterior, dada una $x \epsilon R_m$, el escalar:

$$y = \frac{w^T x}{\|w\|} \qquad (5)$$

es la proyección de "x" a lo largo de "w". Escalar todos los vectores de características por el mismo factor no añade ninguna información relacionada con la clasificación, ignorando el factor de escalado ||w||; se adopta la relación de Fisher discriminante (FDR).

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \qquad (6)$$

donde μ_1 *y* μ_2 son los valores medios y σ_1^2, σ_2^2 las varianzas de "y" en las dos clases w_1, w_2 respectivamente, después de la proyección a lo largo de w.

En (7) se muestra la fórmula que determina LDA multiclase.

$$LDA = \sum_i^m \sum_{j \neq i}^m \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \qquad (7)$$

donde los subíndices i, j se refieren a la media y la varianza correspondiente a la función bajo investigación para las clases w_i, w_j, respectivamente.

Análisis de componentes independientes

ICA, definido en [30], es un caso especial de BSS (Blind Signal Separation). En ICA una característica relevante se define como una característica cuya eliminación deteriora el rendimiento o exactitud del clasificador, mientras que una característica redundante es irrelevante. Características irrelevantes podrían deteriorar el rendimiento del clasificador, por lo tanto la motivación del selector de características es:

- Simplificar el clasificador mediante la selección de características.
- No reducir significativamente la exactitud del clasificador.
- Reducir la dimensionalidad de los datos para que un clasificador pueda manejar grandes volúmenes de estos.

Las técnicas ICA proporcionan las herramientas para el procesamiento de señales estadísticas para óptimas transformaciones lineales de datos multivariantes. Estos métodos son adecuados para la extracción de características, la reducción del ruido, la estimación de la densidad y la regresión.

Es un método computacional para la separación de una señal multivariante en sub-

componentes aditivos suponiendo la mutua independencia estadística de tales señales de origen no gaussiana.

En ICA se encuentran los componentes independientes mediante la maximización de la independencia estadística de los componentes estimados. Se puede elegir una de las muchas maneras de definir la independencia, y esta elección gobierna la forma de los algoritmos de ICA. Las dos definiciones más amplias de la independencia de ICA son: la reducción al mínimo de información redundante y la maximización de la no-gaussianidad. En el estudio de los IDS basados en detección de anomalías se usa la primera definición con el propósito de emplearla como técnica de reducción de características.

La metodología de ICA puede ser descrita de la siguiente forma: cada una de las "h" señales mezcladas $x_1(k)$, $x_2(k)$, ..., $x_h(k)$ es una combinación lineal de "q" componentes independientes $s_1(k)$, $s_2(k)$, ..., $s_q(k)$, es decir, X=AS, donde "A" es una matriz de mezcla. Dada "X", el problema es calcular A y S. Con base en las siguientes dos hipótesis estadísticas, ICA obtiene con éxito los resultados:

- Los componentes son independientes entre sí.
- Cada componente sigue una distribución no gaussiana. Por X=AS, tenemos $S=A^{-1}$, es decir, S=A inversa de X; de otra forma X=WX (donde W es la inversa de A).

La tarea consiste en seleccionar una "W" adecuada que se aplica en la "X" para maximizar el comportamiento no gaussiano de los componentes. Esto se puede hacer a través de un proceso iterativo.

Dado un conjunto de vectores n-dimensionales, las componentes independientes son las direcciones (vectores) a lo largo de los cuales las estadísticas de las proyecciones de los vectores de datos son independientes uno del otro. Formalmente "A" es una transformación del marco de referencia dado hacia el marco de referencia de componentes independientes. Donde X=AS indica que es la distribución marginal:

$$P(S) = \prod P_a(S_i) \qquad (8)$$

"P(s)" es la distribución conjunta en el vector n-dimensional "s". Por lo general, la técnica para la realización del Análisis de Componentes Independientes - ICA se expresa como la técnica para derivar un determinado "W", y=Wx, tal que los componentes de "y" son independientes el uno del otro. Si las distribuciones marginales individuales son no gaussianas entonces la derivada de las densidades marginales llega a una permutación a escala de las funciones de densidad original si, por ejemplo, un "W" se puede obtener. Una de las técnicas generales de aprendizaje para encontrar una adecuada "W" es:

$$W = n(I - \Phi(y)yr)W \qquad (9)$$

Donde $\Phi(y)$ es una función no lineal de la salida del vector ***"y"***.

Análisis de componentes principales

PCA, definido en [31], es una de las técnicas de reducción de dimensionalidad más utili-

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

zada para el análisis y compresión de datos. Esta técnica identifica patrones en los datos, y los expresa en términos de sus semejanzas y diferencias. Una vez que los patrones son localizados a partir de la colección de datos, dichos datos pueden ser comprimidos reduciendo el número de dimensiones, sin una pérdida significativa de información.

El Análisis de Componentes Principales (PCA) es un procedimiento matemático que utiliza una transformación ortogonal para convertir un conjunto de observaciones de variables correlacionadas, en un conjunto de valores de variables no correlacionadas llamadas componentes principales. El número de componentes principales es inferior o igual al número de variables originales. Esta transformación se define de tal manera que el primer componente principal tiene la más alta variación posible, y cada componente de éxito, a su vez, tiene la mayor variación posible con la restricción de que sea ortogonal o correlacionado con los componentes anteriores.

Los componentes principales son garantizados para ser independientes solo si el conjunto de datos es una distribución normal. PCA es sensible a la escala relativa de las variables originales. Dependiendo del ámbito de aplicación, también es llamado la transformada Karhunen-Loéve discreta (KLT), la transformada de Hotelling o la descomposición ortogonal adecuada (POD).

PCA fue inventado en 1901 por Karl Pearson [32] y en la actualidad se utiliza más que todo como una herramienta en el análisis exploratorio de datos y para hacer modelos predictivos. PCA se puede generar por la descomposición de valores propios de una matriz de covarianza de datos o de la descomposición de valor singular de una matriz de datos, por lo general después de la media de centrar los datos de cada atributo. Los resultados de la PCA se suelen tratar en términos de puntuaciones de los componentes (los valores de las variables transformadas correspondientes a un caso particular de los datos) y cargas (el peso por el cual debe ser cada variable original estándar, multiplicado para obtener la puntuación del componente).

Si cada dato tiene ***"N"*** características representadas por ejemplo por x_{11} x_{12} ... x_{1N}, x_{22} x_{21} ... x_{2N}., el conjunto de datos puede ser representado mediante una matriz $X_{n\times m}$. Para la aplicación del método se utiliza el siguiente fundamento matemático. La observación promedio se define como:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (10)$$

La desviación de la media se define como:

$$\boldsymbol{\Phi_i} = x_i - \mu \qquad (11)$$

La matriz de covarianza de la muestra del conjunto de datos se define como:

$$\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

$$= \frac{1}{n}\sum_{i=1}^{n}\Phi_i\Phi_i^T = \frac{1}{n}AA^T \qquad (12)$$

Valores y vectores propios de la muestra de covarianza matriz ***"C"*** son generalmente calculados por la descomposición en valores singulares. Suponiendo que (λ_1, μ_1),

(λ_2, μ_2) ... (λ_m, μ_m) son ***"m"*** pares de vectores propios de la muestra covarianza de la matriz ***"C"***. Los ***"k"*** vectores propios tienen que ser los más grandes valores propios seleccionados. La dimensionalidad del subespacio ***"k"*** se puede determinar por:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \geq \alpha \qquad (13)$$

Dónde ***"α"*** es la razón de la variación en el subespacio a la variación total en el espacio original. Se forma una matriz ***"U"*** con dimensión m×k y cuyas columnas constan de los vectores propios ***"k"***.

Luego del preprocesamiento y normalización del dataset se debe seleccionar el método apropiado para realizar la fase de entrenamiento. En este artículo se aborda el método estadístico basado en Redes Neuronales, denominado SOM (Self-Organizing Map), para la ejecución de esta fase.

SOM

Los Mapas Autorganizativos SOM [33], [34] fueron concebidos por Teuvo Kohonen, investigador del Centro de Investigación de Tecnologías en Redes Neuronales de la Universidad de Helsinki, en Finlandia. Estos mapas hacen posible la representación de datos multidimensionales en espacios de dimensiones mucho menores, por lo general de dimensión 1, 2 o 3. La reducción de la dimensionalidad de los vectores es una técnica de compresión de datos conocida como cuantificación vectorial. Los SOM crean una red que almacena la información de tal manera que todas las relaciones topológicas en el conjunto de entrenamiento se mantienen.

La característica más preponderante de los SOM es que aprende a clasificar los datos mediante un algoritmo de aprendizaje NO supervisado (un SOM aprende a clasificar los datos de entrenamiento sin ningún tipo de control externo). En el enfoque SOM, un vector de entrada se presenta a la red (normalmente una red multicapa feedforward) y este vector es comparado iterativamente con cada uno de los vectores de pesos asociados a los nodos de la estructura del mapa, de tal forma que los pesos de cada vector se recalculen en relación con el vector de entrada. Esto se repite muchas veces y con varios conjuntos de pares de vectores hasta que la red converja en el resultado deseado.

Una red SOM es usualmente creada a partir de un entramado de nodos en dos dimensiones (2D), donde cada uno de ellos está completamente conectado a la capa de entrada. La Fig. 5 ilustra una pequeña red SOM de 4x4 nodos conectados a la capa de entrada que representa un vector de tres entradas.

Cada nodo posee una posición topológica específica con coordenadas (x,y) en el entramado y contiene un vector de pesos de la misma dimensión que los vectores de entrada. Los datos empleados para el entrenamiento usan vectores "V" con una dimensión "n", de la forma: $V_1, V_2, V_3,...,V_n$. Además, cada nodo del entramado contendrá el correspondiente vector de pesos W, de tamaño igual a la dimensión del vector de la capa de entrada, "n", con la forma siguiente: $W_1, W_2, W_3...W_n$. Las líneas que conectan los nodos en la Fig. 5 solo repre-

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

sentan adyacencia, no significa que exista una conexión como se indica normalmente cuando se habla de una red neuronal.

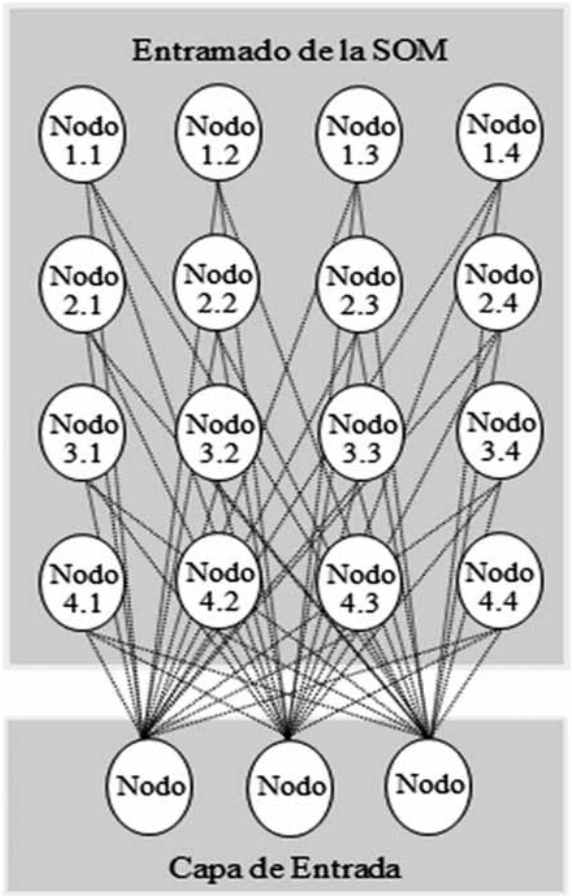

Fig. 5 Arquitectura SOM 4x4

El algoritmo de aprendizaje

El proceso comienza con una distribución inicial de pesos al azar en cada nodo del entramado, y luego de muchas iteraciones, el SOM eventualmente se asienta en un mapa de zonas estables. Cada zona es un clasificador de características y una representación gráfica de este es pensar en la salida como un mapa de características del espacio de entrada.

El entrenamiento ocurre en varios pasos y en muchas iteraciones; éstos se enuncian a continuación:

1. Cada nodo es inicializado con su respectivo vector de pesos, éstos se establecerán en pequeños valores aleatorios estandarizados.
2. Un vector es elegido al azar entre el conjunto de datos de entrenamiento y se presenta al entramado (vector de entrada).
3. Cada nodo se examina para calcular cuál es el peso que más se "aproxima" al vector de entrada. El nodo ganador se conoce comúnmente como la mejor unidad de coincidencia (BMU - Best Matching Unit).
4. Se calcula el radio de la vecindad del BMU, que es un valor inicialmente grande, asociado al tamaño del entramado, y va disminuyendo con el pasar del tiempo. Los nodos que se localizan dentro de este radio se consideran vecinos de la BMU.
5. Cada nodo vecino a la BMU ajusta sus pesos para hacerlos coincidir más con el vector de entrada. Cuanto más cerca esté el nodo de la BMU, más será alterado su peso.
6. Se repite el paso 2 para "n" iteraciones.

Para determinar la mejor unidad de coincidencia BMU, se efectúa un proceso de iteración a través de cada uno de los nodos del entramado y se calcula la distancia euclidiana entre el vector de peso de cada nodo y el vector de entrada actual. El nodo con un vector de pesos más cercano al vector de

entrada se etiqueta como BMU. La distancia euclidiana se calcula en (14).

$$Dist.Euclidiana = \sqrt{\sum_{i=0}^{n}(V_i - W_i)^2} \quad (14)$$

Donde "V" es el vector de entrada actual y "W" es el vector de pesos de cada nodo del entramado.

Luego de determinar cuál es la BMU, el paso siguiente es calcular cuál de los otros nodos del entramado, que no son la BMU, están en el vecindario. Una vez identificados dichos nodos, se procederá a alterar sus vectores de peso. Por lo tanto, se calcula el radio de la vecindad, aplicando Pitágoras para determinar si cada nodo está dentro de la distancia radial o no. En la Fig. 6, suponiendo que la BMU posee la coordenada 3.3 y la distancia radial es 2, los nodos del vecindario son los contenidos dentro de la circunferencia punteada, dichos nodos están sombreados con gris claro.

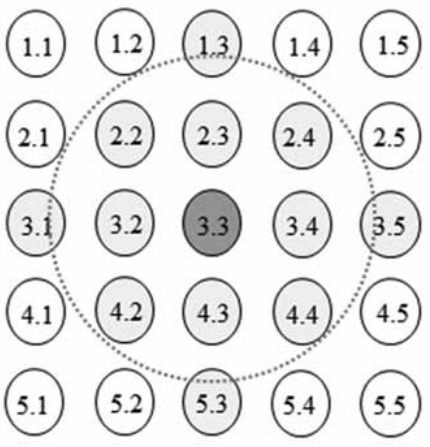

Fig. 6. Vecindario de una SOM 5x5

El área del vecindario se contrae con el tiempo, dado que es directamente proporcional al radio del vecindario, el cual se calcula a partir de la función exponencial decreciente, definida en (15).

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right), t = 1,2,3... \quad (15)$$

Donde sigma sub cero (σ_0) denota el ancho del entramado en el tiempo "t_0" y lambda (λ) denota un tiempo constante "t" que es el actual paso de tiempo.

Luego de varias iteraciones la vecindad se ajustará al tamaño de un solo nodo, la BMU, que determinará el valor del radio, lo cual se requiere para poder identificar si un nodo está o no dentro del vecindario. Si un nodo se encuentra dentro de la vecindad, entonces su vector de pesos se debe ajustar.

Tanto el nodo BMU como cada uno de los nodos ubicados en la vecindad, tienen un vector de pesos ajustado de acuerdo con (16).

$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)) \quad (16)$$

Donde "*t"* es el paso del tiempo y "*L"* es una variable pequeña llamada la tasa de aprendizaje, la cual disminuye en el tiempo; (16) indica que el peso en el instante "*t+1"* se ajusta para los nodos del vecindario a partir del peso en el instante actual "*W(t)*", más una fracción *"L(t)"*, de la diferencia entre el peso actual del nodo *"W(t)"* y el peso del vector de entrada en el instante actual *"V(t)"*.

La tasa de aprendizaje, al igual que el radio del vecindario, usan una función exponencial decreciente para determinar su valor en la variación del tiempo. Tal como se muestra en (17).

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right), t = 1,2,3... \qquad (17)$$

En (16) $\Theta(t)$, representa la cantidad de influencia que la distancia de un nodo a la BMU tiene en su aprendizaje, en el instante de tiempo actual. $\Theta(t)$ se calcula mediante (18).

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right), t = 1,2,3... \qquad (18)$$

Donde "*dist"* es la distancia de un nodo a la BMU y ***"σ"*** es el radio de la vecindad, anteriormente enunciado. La función $\Theta(t)$ también decrece en el tiempo.

Aplicaciones de SOM

Los Mapas Autorganizativos de Kohonen facilitan la comprensión de las relaciones existentes en grandes colecciones de datos, poseen aplicaciones en una variada gama de áreas, como: la clasificación de información bibliográfica, exploración y análisis de imágenes, diagnóstico médico, interpretación de la actividad sísmica, reconocimiento de la voz, compresión de datos, separación de fuentes de sonido, modelización ambiental y otras más.

Algunos ejemplos puntuales de aplicación de los SOM se refieren a: el agrupamiento gráfico basado en SOM para la extracción de ideas principales en documentos [35]; un sistema de correlación basado en la ingeniería de kansei y en la evaluación de características reales de automóviles, como soporte a la compra de coches, basado en SOM [36]; sistema híbrido basado en mapas auto-organizativos de Kohonen, para la predicción de variación de precios de las acciones de la bolsa de valores [37]; sistema sistólico de alto rendimiento SOM sobre núcleo IP en FPGAs, para el procesamiento de miles de elementos en tiempo real, clasificando datos relacionados con la genómica o proteómica [38]; reconocimiento de patrones de falla en sistemas termodinámicos basados en SOM, aplicados a centrales de energía térmica [39]; extracción automática de caminos mediante el censado de imágenes de alta resolución, basado en SOM [40]; aplicación de mapas auto-organizativos (SOM) para determinar la descomposición de productos químicos [41]; sistemas de detección de intrusiones anormales en redes informáticas, basados en SOM [42]; técnicas para visualización de ADN, basadas en SOM, usando Microarrays para el análisis de datos [43] y modelamiento del contorno de imágenes basado en SOM [44]. Por otra parte, existe una gran variedad de SOM que pueden ser consultadas en Redes SOM constructivas [45], Red de crecimiento en malla incremental [46], Mapas autorganizativos de crecimiento [47], redes GAS de crecimiento neural [48], red GAS neural [49], red GAS de crecimiento neural basada en densidad [50], [51], Aprendizaje Hebbiano Competitivo [52] y Mapas Auto-organizativos de Crecimiento Jerárquico [53].

EXPERIENCIA DE SIMULACIÓN

Los experimentos de simulación se efectuaron sobre un tipo de datos. Se analizaron registros de conexiones de red, procedentes del dataset NSL-KDD DARPA.

Modelo propuesto

El modelo comprende tres fases: entrenamiento, clasificación y cálculo de métricas de desempeño. Para su aplicación se implementaron varios escenarios de simulación variando la cantidad de características por evaluar en las dos primeras fases; para ello se priorizó la escogencia de las características mediante su razón discriminante de Fisher (FDR). En la fase de entrenamiento se carga el dataset KDDTrain+_20Percent de DARPA, ya balanceado por tipo de conexión (normal o ataque), este dataset representa el flujo de datos de entrada; a continuación se aplica el algoritmo de reducción de características FDR y se seleccionan las características por orden de relevancia y, por último, se realiza el entrenamiento del SOM, lo cual implica una normalización, creación de la estructura de datos, inicialización del mapa, entrenamiento del mismo y un etiquetado de los datos.

En la fase de clasificación se carga el dataset KDDTest+ de DARPA, el cual representa el flujo de datos que se va a clasificar, diferente del conjunto de datos de entrenamiento; se reducen las características usando la tasa discriminante de Fisher generada a partir del nuevo dataset, teniendo en cuenta la misma cantidad de características seleccionadas en la fase de entrenamiento y se procede por último a clasificar los datos, generando una estructura de datos que contiene tanto el etiquetado de la nueva data como el etiquetado predictivo a partir del cálculo de las BMU basado en el mapa creado en la fase de entrenamiento.

En la fase final se calculan las métricas de desempeño; para ello se recorre la estructura de datos generada en la fase anterior, calculando falsos positivos, verdaderos positivos, falsos negativos y verdaderos negativos, los cuales permiten determinar las métricas de sensibilidad y especificidad que van a indicar la eficiencia del modelo planteado. Una descripción gráfica de lo anteriormente expuesto se aprecia en la Fig. 7.

Medidas de desempeño utilizadas

Para evaluar el modelo propuesto se emplearon dos medidas de desempeño: la *sensibilidad* (sensitivity) y la *especificidad* (specificity), utilizadas en pruebas de clasificación binaria, como es nuestro caso (conexiones normales y ataques). La sensibilidad mide la proporción de "verdaderos positivos" que son correctamente identificados como tales, entendiendo que un "verdadero positivo" es un ataque que ha sido correctamente identificado como tal. La especificidad en cambio, mide la proporción de "verdaderos negativos" que se han identificado correctamente, entendiendo por "verdadero negativo" una conexión de tráfico normal correctamente identificada como tal. Las fórmulas que definen tales métricas han sido enunciadas anteriormente en (1) y (2).

Dataset empleado en la simulación

Los registros de conexión de red utilizados en las fases de entrenamiento y clasificación de las simulaciones proceden del dataset NSL-KDD DARPA. Para el entrenamiento se utilizó el archivo KDDTrain+_20Percent.txt del cual se extrajeron 25.192 registros

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

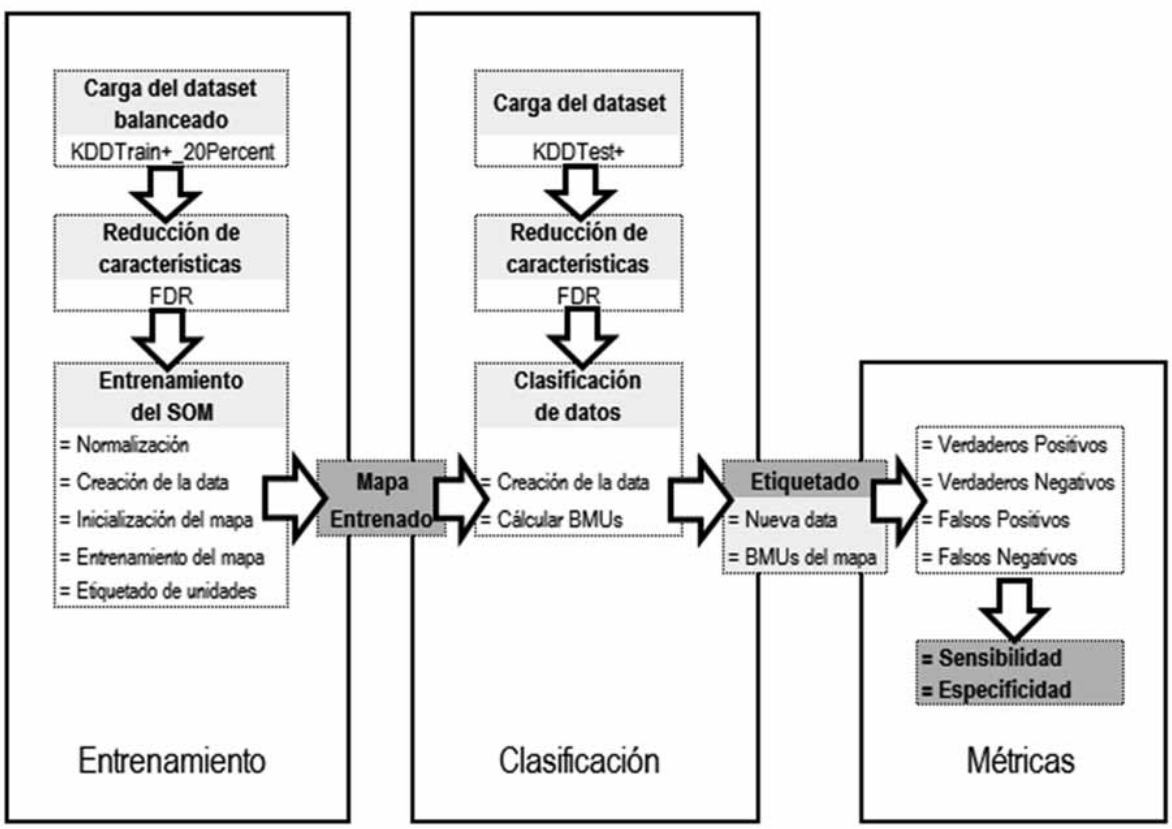

Fig. 7 Modelo de detección de intrusiones propuesto

de conexiones de red, cada uno de los cuales contiene 41 características. Con el propósito de que el algoritmo de entrenamiento aprendiese a identificar equitativamente tanto conexiones normales como ataques, se realizó un balanceo del dataset antes mencionado, seleccionando aleatoriamente 23.486 registros de conexiones de éste, distribuidos en un 50% a conexiones normales y otro 50% a conexiones de ataques. Una vez entrenada la red neuronal, para la fase de clasificación se utilizó el archivo KDDTest+.txt del cual se extrajeron 22.544 registros de conexión de red, cada uno de los cuales contiene 41 características y se realizó el proceso de clasificación con tales registros.

Montaje y resultados experimentales

Se generaron ocho (8) escenarios de simulación aplicando el modelo propuesto, para: 5, 10, 15, 20, 25, 30, 35 y 40 características, identificadas por el orden de prioridad establecido por FDR, teniendo en cuenta que durante el entrenamiento se normalizó utilizando la operación lineal de varianza, se inicializó el mapa con un tamaño de 4x4, el radio inicial del entrenamiento fue de 4, el radio final del entrenamiento fue de 0,00001, se utilizó una longitud de entrenamiento de 2000 y un nivel de seguimiento de 1. Se utilizó el algoritmo de procesamiento por lotes (batch) descrito en la sección "El Algoritmo de Aprendizaje", de este documento, para el entrenamiento del mapa auto-organizativo.

A partir de todo lo anteriormente indicado se obtuvieron los resultados que se aprecian en la Tabla VII.

Tabla VII
Simulación de reducción de características usando FDR y comparando las métricas Sensibilidad y Especificidad

Cantidad de características	Métricas	
	Sensibilidad	Especificidad
5	99,36	53,36
10	97,06	50,48
15	98,47	56,15
20	99,69	51,21
25	44,09	15,23
30	39,34	12,88
35	34,49	36,06
40	81,99	44,10

A partir de lo anterior se ha determinado que al utilizar las 20 características de mayor prioridad, identificadas con FDR, el clasificador presenta la mayor tasa de detección de ataques, es decir, una sensibilidad del 99,69%, y que al utilizar las 15 características de mayor prioridad, identificadas con FDR, el clasificador presenta la mayor tasa de detección de tráfico normal, es decir, una especificidad del 56,15%. Es importante resaltar que en un IDS el objetivo primordial es la identificación de ataques o tráfico malicioso, por ello es preponderante lograr altos porcentajes de acierto en la sensibilidad. Se aprecia además que con la escogencia de 5 características el porcentaje de sensibilidad es del 99,36%, que es muy cercano al porcentaje que se lograr usando 20 características. Se debe tener en cuenta que la carga computacional generada en tiempo real haciendo uso de 5 características es comparativamente inferior que hacer uso de 20 características.

CONCLUSIONES

El método de extracción de características FDR es más susceptible a la detección de ataques (Verdaderos Positivos) que a la detección de tráfico normal (Verdaderos Negativos). Pese a esto es considerablemente útil cuando se aplica en IDS dado que éstos tienen como finalidad primordial la detección de ataques.

El modelo propuesto debe ser mejorado con miras en una futura implementación a nivel comercial en un IDS, ya que la tasa de especificidad es muy baja, lo que indica que se debe fortalecer la detección de verdaderos negativos.

Los Sistemas de Detección y Prevención de Intrusos no deben ser vistos como una solución totalitaria en la identificación de ataques o tráfico malicioso, más bien hacen parte de la solución, coexistiendo con otros mecanismos de prevención y actuación como VPN (Redes Privadas Virtuales), Cortafuegos y Listas de Control de Acceso, entre otros.

TRABAJOS FUTUROS

Generar un escenario comparativo de evaluación de las técnicas de reducción de características FDR, ICA y PCA aplicados a colecciones de datos de IDS entrenados con SOM y GHSOM (Mapas Auto-organizativos de Jerarquía Creciente).

Plantear un sistema híbrido que identifique ataques con base en abusos y anomalías, capaz de comparar conexiones en tiempo real con una base de datos de firmas, y de detectar nuevos ataques mediante el uso de

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

algoritmos de aprendizaje no supervisado, posibilitando la evaluación de tráfico procedente de host y de red, de forma distribuida.

AGRADECIMIENTOS

Este artículo pudo desarrollarse gracias al apoyo del Departamento de Arquitectura y Tecnología de Computadores y a la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada, España y a diferentes estamentos de la Corporación Universidad de la Costa de Barranquilla, Colombia, como son la dirección del programa de Ingeniería de Sistemas, la Facultad de Ingeniería, el grupo de Investigación de Ingeniería del Software - Redes y el Centro de Investigaciones de la Facultad de Ingenierías.

REFERENCIAS

[1] SourceFire - Snort. Disponible en: http://www.snort.org/

[2] CheckPoint® Software Technologies Ltd. NFR (Network Flight Recorder). Disponible en: http://www.checkpoint.com/corporate/nfr/index.html

[3] L. T. Heberlein. *Network Security Monitor, Final Report.* Lawrence Livermore National Laboratory (LLNL) and the University of California, Davis (UCD). February 1995. Disponible en: http://seclab.cs.ucdavis.edu/papers/NSM-final.pdf

[4] CISCO System. *Cisco Intrusion Detection (NetRanger)*. Disponible en: http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/index.shtml

[5] IBM. *RealSecure Network Sensor*. Disponible en: http://www-947.ibm.com/support/entry/portal/Overview/Software/Tivoli/ RealSecure_Network_Sensor

[6] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. "A Detailed Analysis of the KDD CUP 99 Data Set", *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009. CISDA 2009, pp. 1-6, july 2009.

[7] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang. "A novel anomaly detection scheme based on principal component classifier," *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03)*, pp. 172-179, 2003.

[8] USC Information Sciences Intitute. "Common Intrusion Detection Framework", Disponible en: http://gost.isi.edu/cidf/

[9] CIDF Working Group (Clifford Kahn, Don Bolinger and Dan Schnackenberg). *DRAFT Specification. Communication in the Common Intrusion Detection Framework* v 0.7. 8 June 1998. Disponible en: http://gost.isi.edu/cidf/drafts/communication.txt

[10] Rich Feiertag, Cliff Kahn, Phil Porras, Dan Schnackenberg *et al. A Common Intrusion Specification Language (CISL)*. 11 June 1999. Disponible en: http://gost.isi.edu/cidf/drafts/language.txt

[11] Australian Computer Emergency Response Team. Disponible en: http://www.auscert.org.au/

[12] Internet Engineering Task Force. Disponible en: http://datatracker.ietf.org/wg/idwg/

[13] Common Vulnerabilities and Exposures - CVE. Disponible en: http://cve.mitre.org/about/index.html

[14] Prelude Technologies. Disponible en: http://www.prelude-technologies.com/

[15] National Institute of Standards and Technology - National Computer Security Center. National Computer Security Conference. DIANE Publishing Company. October 1992. p. 272.

[16] SRI - International a real-time Intrusion-Detection Expert System (IDES). Disponible en: http://www.csl.sri.com/papers/9sri/9sri.pdf

[17] S. Noel, D. Wijesekera, and C. Youman. "Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt". In *Applications of Data Mining in Computer Security,* D. Barbarà and S. Jajodia (eds.), Kluwer Academic Publisher, 2002.

[18] A. Lazarevic, J. Srivastava, and V. A. Kumar, "Survey of Intrusion Detection techniques". book *Managing Cyber Threats: Issues, Approaches and Challenges*, to be published by Kluwer in spring 2004.

[19] Working Group 2 of the Joint Committee for Guides in Metrology (JCGM/WG 2). *International vocabulary of metrology - Basic and general concepts and associated terms (VIM)*. 3rd edition. 2008. Disponible en: http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2008.pdf

[20] Lincoln Laboratory of Massachusetts Institute Tecnology - MIT. Disponible en: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html

[21] KDD Cup 1999. Disponible en: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[22] The NSL-KDD Data Set. Disponible en: http://nsl.cs.unb.ca/NSL-KDD/

[23] The University of Waikato. Disponible en: http://www.cs.waikato.ac.nz/~ml/weka/index.html

[24] D. Graupe, *Principles of Artificial Neural Networks,* World Scientific Publishing Co. Pte. Ltd., Singapore. pp. 191-222, 1997.

[25] S. Balakrishnama and A. Ganapathiraju, *Linear Discriminant Analysis - A Brief Tutorial,* Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University. 1998.

[26] R. Fisher. "The Use of Multiple Measurements in Taxonomic Problems" In: *Annals of Eugenics*, 7, p. 179-188. 1936.

[27] McLachlan. "Discriminant Analysis and Statistical Pattern Recognition" In: *Wiley Interscience.* 2004.

[28] Martinez & Kak. "PCA versus LDA" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2): 228-233. 2004.

[29] V. Venkatachalam and S. Selvan. "Performance comparison of intrusion detection system classifiers using various feature reduction techniques". *International journal of simulation,* 2008 - Citeseer.

[30] A. Hyvärinen and E. Oja, "Independent Component Analysis: Algorithms and Applications", *Neural Networks*, Volume 13, Issue 4-5 pp. 411-430. 2000.

[31] I. T. Jolliffe, *Principal Component Analysis*, Springer Verlag, New York, NY, third edition. 2002.

[32] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space" (PDF). *Philosophical Magazine* 2 (6): 559-572. 1901.

[33] T. Kohonen. "Self-organizing Maps". *Springer Series in Information Sciences.* Volume 30, 1997. 2nd edition.

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA

[34] Kohonen's Self Organizing Feature Maps. Disponible en: http://www.ai-junkie.com/ann/som/som1.html

[35] Do Phuc, and Mai Xuan Hung, "Using SOM based Graph Clustering for Extracting Main Ideas from Documents". *Research, Innovation and Vision for the Future,* 2008. RIVF 2008. IEEE International Conference on. pp. 209-214. July 2008.

[36] I. Nakaoka, J.-I. Kushida and K. Kamei, "Proposal of Group Decision Support System Using "SOM" for Purchase of Automobiles". *Innovative Computing Information and Control*, 2008. ICICIC '08. 3rd International Conference on p. 482. June 2008.

[37] M. O. Afolabi and O. Olude, "Predicting Stock Prices Using a Hybrid Kohonen Self Organizing Map (SOM)". *System Sciences,* 2007. HICSS 2007. 40th Annual Hawaii International Conference on. p. 48. Jan. 2007.

[38] I. Manolakos and E. Logaras, "High throughput systolic SOM IP core for FPGAs". *Acoustics, Speech and Signal Processing*, 2007. ICASSP 2007. IEEE International Conference on. pp. II-61 - II-64. April 2007.

[39] Kuang Yin and Luo Gang, "Fault Pattern Recognition of Thermodynamic System Based on SOM". *Electrical and Control Engineering (ICECE)*, 2010. International Conference on. pp. 3742-3745. June 2010.

[40] Hao Ying, Wang Li-qiang and Zhao Xi'an. "Automatic Roads Extraction From High-resolution Remote Sensing Images Based on SOM". *Natural Computation (ICNC)*, 2010 Sixth International Conference on. pp. 1194-1198. Aug. 2010.

[41] H. Tokutaka, K. Yoshihara, K. Fujimura, K. Iwamoto, T. Watanabe and S. Kishida, "Applications of Self-organizing Maps (SOM) to the Composition Determination of Chemical Products". *Neural Networks Proceedings*, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on. pp. 301-305 vol. 1. May 1998.

[42] Li Min and Wang Dongliang, "Anormaly Intrusion Detection Based on SOM". *Information Engineering,* 2009. ICIE '09. WASE International Conference on. pp. 40-43. July 2009.

[43] J.C. Patra, J. Abraham, P.K. Meher, and G. Chakraborty, "An Improved SOM-based Visualization Technique for DNA Microarray Data Analysis". *Neural Networks (IJCNN),* The 2010 International Joint Conference on. pp. 1-7. July 2010.

[44] Y. V. Venkatesh, S.K. Raja, and N. Ramya, "A Novel SOM-based Approach for Active Contour Modeling". *Intelligent Sensors, Sensor Networks and Information Processing Conference,* 2004. Proceedings of the 2004. pp. 229-234. Dec. 2004.

[45] E. Cuadros-Vargas, *Recuperação de informação por similaridad e utilizando técnicas inteligentes.* PhD thesis, Department of Computer Science - University of Sao Paulo. 2004.

[46] J. Blackmore and R. Miikkulainen, "Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map". In P*roceedings of the International Conference on Neural Networks ICNN93,* 1993, volume I, pp. 450-455, Piscataway, NJ. IEEE Service Center.

[47] D. Alahakoon, S. K. Halgamuge and B. Srinivasan, "A structure adapting feature map for optimal cluster representation". In *International Conference on Neural Information Processing ICONIP98,* 1998. pp. 809-812.

[48] B. Fritzke, "A growing neural gas network learns topologies". In G. Tesauro, D. S. Touretzky and T. K. Leen, editors, *Advances in Neural Information Processing Systems* 7, 1995, pp. 625-632. MIT Press, Cambridge MA.

[49] T. Martinetz and K. Schulten, "Topology representing networks". *Neural Networks,* 1994. 7(3):507-522.

[50] A. Ocsa, C. Bedregal and E. Cuadros-Vargas, "DB-GNG: A constructive self-organizing map based on density". In *Proceedings of the International Joint Conference on Neural Networks (IJCNN07)*. IEEE, 2007.

[51] Y. Prudent and A. Ennaji, *A k nearest classifier design*. ELCVIA, 2005. 5(2): 58-71.

[52] R. H. White, "Competitive hebbian learning: algorithm and demonstrations". *Neural Networks*, 1992. 5(2): 261-275.

[53] *The Growing Hierarchical Self-Organizing Map*. Department of Software Technology. Vienna University of Technology. Septiembre 2011. Disponible en: http://www.ifs.tuwien.ac.at/~andi/ghsom/description.html#insertion

EDUCOSTA
EDITORIAL UNIVERSITARIA DE LA COSTA