Schemes of combination of decision trees as a strategy for anomaly detection

I.J.C. Quintana-Zaez, Dr.[1], Héctor R. Velarde-Bedregal, Dr.[2], J. Anton-Vargas, MsC.[1], G. Joaquim-Luis, Ing.[1]
[1]Facultad de Informática y Ciencias Exactas, Universidad de Ciego de Ávila, Cuba,
cjquintanazaez@gmail.com, jarvinalberto@gmail.com, geronimo@unica.cu
[2]Universidad Catolica de Santa Maria, Perú. Facultad de Ciencias e Ingenierías Físicas y Formales,
hvelardeb@ucsm.edu.pe

Abstract – Security of the data consumed, generated and stored is crucial to the quality of life in today's society. For this reason, this paper proposes a comparative study of different combination schemes of multiple classifiers based on decision trees, due to its scalability and easy implementation. As a result, precision and recall values of about 97% and 100% were obtained, showing their high reliability, reducing false alarms and high generalization capacity. A comparison with a deep learning based algorithm showed that tree combination strategies are competitive and with statistically similar and superior results to the state-of-the-art. In the end, the results suggest that adaptive strategies such as XGBoost or highly randomized strategies such as Random Forest or Extra-Tree can be alternatives for the protection of precious data on the network.

Keywords: IDS, decision trees, Random Forest, Extra-Tree, Deep learning.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2020.1.1.306
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Esquemas de combinación de árboles de decisión como estrategia para la detección de anomalías.

I.J.C. Quintana-Zaez, Dr.[1], Héctor R. Velarde-Bedregal, Dr.[2], J. Anton-Vargas, MsC.[1], G. Joaquim-Luis, Ing.[1]
[1]Facultad de Informática y Ciencias Exactas, Universidad de Ciego de Ávila, Cuba,
cjquintanazaez@gmail.com, jarvinalberto@gmail.com, geronimo@unica.cu
[2]Universidad Catolica de Santa Maria, Perú. Facultad de Ciencias e Ingenierías Físicas y Formales,
hvelardeb@ucsm.edu.pe

Abstract – Security of the data consumed, generated and stored is crucial to the quality of life in today's society. For this reason, this paper proposes a comparative study of different combination schemes of multiple classifiers based on decision trees, due to its scalability and easy implementation. As a result, precision and recall values of about 97% and 100% were obtained, showing their high reliability, reducing false alarms and high generalization capacity. A comparison with a deep learning based algorithm showed that tree combination strategies are competitive and with statistically similar and superior results to the state-of-the-art. In the end, the results suggest that adaptive strategies such as XGBoost or highly randomized strategies such as Random Forest or Extra-Tree can be alternatives for the protection of precious data on the network.

Keywords: IDS, decision trees, Random Forest, Extra-Tree, Deep learning.

I. Introducción

En la actualidad no solo las personas están conectadas a las redes. Cada vez más, dispositivos de diversa índole están siendo conectados a estas, desde teléfonos inteligentes, electrodomésticos, automóviles, sensores, entre otros [1]. La diversidad de datos generados y almacenados en la red alcanzan niveles considerables [2]. Prestarle la atención requerida a integridad de los datos cobra cada vez más fuerza, siendo la seguridad de la redes de computadoras un tópico rigurosamente tratado en nuestros días [3]. Hondar en las diferentes formas de proteger tales datos es un proceso engorroso para los especialistas debido a la cantidad y variabilidad de estos. Pero no solo las características de los datos hacen difícil mantener su integridad, con el surgimiento y evolución de tecnologías avanzadas, los atacantes incrementan los ataques a las redes, los cuales son cada vez más potentes y variables [4].

Con el objetivo Con el objetivo de proteger la integridad de los datos que son transmitidos en red y de los dispositivos empleados para esta transmisión pueden ser empleados los sistemas de detección de intrusos en redes (*Network Intrusion Detection System*, NIDS) [5], [4]. Estos métodos pueden detectar anomalías (intrusiones) dada la ocurrencia de eventos que generen cambios relativamente rápidos o con baja probabilidad de ocurrencia. Los sistemas de detección de intrusos pueden ser divididos en varias ramas según su aplicación; (a) métodos de detección basados en firmas (*Signature-based*), basados en aprendizaje automatizado e híbridos. La principal diferencia entre los dos primeros radica en que los basados en firmas necesitan la supervisión de expertos, que son encargados de definir los patrones anómalos que serán empleados por el método de detección, por lo que un factor importante que es la capacidad de generalización depende de los expertos. En cambio los métodos de detección basados en aprendizaje automatizado emplean algoritmos computacionales capaces de aprender por si solos y adaptarse a patrones nunca antes vistos, por lo que su capacidad de generalización es superior [6].

Los métodos de detección de anomalías basadas en aprendizaje automatizado han devenido en una herramienta útil debido a la capacidad de tratar con grandes cantidades de datos y a la capacidad de detectar ataques nunca antes vistos. De manera general en este campo de investigación se emplean métodos basados en naives Bayes [7], [8], [9], máquinas de vector soporte [10], [11], [12], basados en vecindad [13], [14], árboles de decisión [15], [16], esquemas de múltiples clasificadores [17], [18] y redes neuronales y aprendizaje profundo [19], [20], [21], [22], [23], entre otros [24], [25].

A pesar de los avances logrados en el estado-del-arte [4], [26], [27], no todos los métodos logran la capacidad de generalización deseada, obteniendo niveles de precisión que varía entre 60% y 98% aproximadamente, para diferentes conjuntos de datos de prueba, además no se explotan en su totalidad esquemas de múltiples clasificadores basados en árboles de decisión. No obstante, se continúan dedicando esfuerzos en pos de mejoras en el proceso de detección de anomalías.

En el presente artículo se propone realizar un estudio sobre diferentes esquemas de combinación de clasificadores basados en árboles de decisión, que permita establecer si estos pueden ser una alternativa para la detección de anomalías frente a la total expansión de las técnicas de aprendizaje profundo [28]. El aporte del artículo está en mostrar como diferentes métodos altamente aleatorios pueden constituir una contraparte a métodos conexionistas como las redes neuronales. Incluso, en estudios posteriores se pretende crear combinaciones de estos métodos para analizar su competitividad con el estado-del-arte.

El artículo está estructurado como sigue: la sección de materiales y métodos donde se muestran tendencias del estado-del-arte y el diseño de la experimentación. La sección de análisis y discusión de los resultados, donde se evalúa el desempeño de los métodos seleccionados para la comparación. Finalmente, las conclusiones del trabajo resumen los resultados encontrados y su discusión.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2020.1.1.306
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

II. Materiales y Métodos

La propuesta de investigación desarrollada en este artículo se centra en el análisis del desempeño de los esquemas de combinación de múltiples clasificadores que emplean árboles de decisión como clasificadores base. Dicho empleo de los árboles de decisión permite a los investigadores, obtener clasificadores rápidos en su construcción, y muy importante abstraerse de la selección de características ya que los árboles de manera intrínseca permiten seleccionar los mejores atributos.

La selección de atributos para mejorar el rendimiento de los métodos empleados para la detección de anomalías es una tarea ardua que ha sido llevada a cabo de diferentes maneras en el estado-del-arte. En [15] se propone un método para seleccionar rasgos basado en dos capas. La primera capa emplea Ganancia de Información para ordenar los atributos y generar un nuevo conjunto de atributos basándose en una medida global de precisión. Posteriormente, la segunda capa realiza un proceso similar a la primera, pero en busca de un máximo local de la precisión de los atributos. Los autores de este estudio citado sugieren que reducir los rasgos puede aumentar la precisión de los modelos.

Pero no solo se ha aplicado la selección de atributos en su forma tradicional, con la expansión de las redes neuronales y el aprendizaje profundo a diversos campos de investigación, los investigadores han aprovechado las bondades de estas técnicas. En [29] se emplea un esquema de detección de anomalías que aplica un modelo *Deep Belief Network* basado en *Restricted Boltzmann Machine* para reducir atributos. Posteriormente, los autores realizan un análisis comparativo entre variantes, que emplean la reducción de rasgos propuesta y sin aplicar tal reducción. Y sugieren que dicha variante de reducción de rasgos aporta mejores resultados debido a la capacidad de abstraerse de las arquitecturas basadas en redes neuronales.

Por otro lado en [30], se desarrolla un método basado en aprendizaje profundo para la detección de anomalías. El modelo se divide en dos procesos, en el primero se aplica aprendizaje de rasgos no supervisado (*Unsupervised Feature Learning*) sobre datos no etiquetados. Seguido, dicha representación de rasgos es aplicada a datos etiquetados (con un atributo clase), para un posterior proceso de clasificación. Los autores de dicho modelo emplean aprendizaje autodidáctico (*Self-taught Learning*) como técnica de reducción de características no supervisada.

Como se puede apreciar brevemente se experimenta de diferentes maneras para encontrar los mejores atributos, pero desde la perspectiva del empleo de múltiples clasificadores basados en árboles de decisión, se estarían aplicando múltiples selectores de atributos que combinarían sus decisiones para lograr un objetivo común.

A. Esquemas de combinación de clasificadores.

1. Bagging. Es el acrónimo de *Bootstrap AGGregatING* [31], que viene a significar agregado de remuestreos. El método construye N clasificadores base, cada uno de ellos utilizando el mismo algoritmo, pero distintos conjuntos de entrenamiento. Cada conjunto de entrenamiento se obtiene a partir de un remuestreo con reemplazamiento a partir del conjunto de entrenamiento original. Para elegir un clasificador base óptimo para Bagging es clave que sea sensible a las pequeñas variaciones en el conjunto de entrenamiento que pueda introducir el remuestreo (i.e., el clasificador base sea inestable) como por ejemplo los árboles de decisión.

Bagging ha sido empleado para la detección de anomalías en la red en diferentes artículos [32]. En este trabajo los autores seleccionan rasgos relevantes basándose en su viabilidad para cada tipo de ataque. Al final, el espacio de rasgos es reducido de 41 a 15 usando algoritmos genéticos.

2. Random Forest (RF) [33] es un modelo predictivo basado en las combinación de múltiples y diversos árboles de decisión. Entre sus ventajas está la reducción del error de generalización mientras mayor es el número de árboles empleados para construir el modelo. También que dicho error depende de la fortaleza de cada árbol y de la correlación existente entre ellos. Además, es capaz de manejar grandes cantidades de datos y hacer frente a valores raros (*outliers*). Por sus bondades ha sido empleado en diferentes ramas de investigación, tales como la predicción de estructuras de proteínas [34], [35], en el análisis de sentimiento en Twitter [36] o en la detección de objetos activos [37].

En [17] se emplea un modelo basado en RF para detectar anomalías en la red. El algoritmo atraviesa varias etapas, entre las cuales están, agrupamientos ya que el conjunto de datos original es dividido en cuatro partes y luego los atributos para cada parte son filtrados. Para finalmente entrenar un RF con cada subconjunto. Como resultado se obtiene que RF puede desempeñarse mejor que la mayoría de los métodos tradicionales en el proceso de detección de ataques, obteniendo hasta 99,67% de exactitud para el conjunto de datos NSL-KDD, además logra reducir el rango de falsas alarmas.

3. Extra Tree. Este esquema de combinación de árboles de decisión es llamado árboles extremadamente aleatorios (*Extremely Randomized Trees*). Este método es similar a RandomForest, debido a que construye múltiples árboles de decisión altamente aleatorios mediante la selección de subconjuntos de rasgos aleatorios para realizar las particiones en los nodos. Pero la diferencia radica en que, Extra-tree no realiza muestreo con remplazamiento y los nodos realizan la partición no seleccionando el mejor valor para la división sino empleando un criterio aleatorio, [38]. Extra-Tree se ha empleado para la detección de anomalías en construcciones inteligentes e Internet de las Cosas [39], [40].

4. AdaBoost. Se define *Boosting* como el problema de producir un clasificador muy preciso a partir de la combinación de otros más simples y moderadamente imprecisos. Ada Boost es uno de los algoritmos de referencia de la familia Boosting [41]. La construcción de este método sucede de forma iterativa, donde en cada etapa del entrenamiento los clasificadores base actuales dan más importancia a las instancias del conjunto de entrenamiento incorrectamente clasificadas por el clasificador base de la iteración anterior. Luego en la clasificación, la

predicción se realiza en base a un esquema de votación ponderado, dando mayor peso en la votación a aquellos clasificadores base con un mayor acierto sobre el conjunto de entrenamiento. Este método adaptativo se ha empleado en la detección de anomalías [42], donde su combinación con un algoritmo evolutivo muestra que puede ser una estrategia competitiva con respecto al estado-del arte [43].

5. XGBoost. XGBoost (*Extreme Gradient Boosting*) es un modelo de aprendizaje automatizado el cual entre sus bondades muestra una reducción de los tiempos de entrenamiento, altos niveles de exactitud y puede manejar valores perdidos [44], [45]. Este modelo basado en árboles de decisión ha sido aplicado en diversos campos mostrando ser competitivo con los algoritmos del estado-del-arte. Una de las áreas de aplicación es la detección de virus informáticos [46].

En [18] un algoritmo XGBoost es empleado para predecir anomalías en la red. Los autores del estudio comparan el modelo con otros del estado-del-arte, donde se incluye RandomForest. Al final se demuestra que dicho modelo puede competir y superar a la mayoría de los métodos analizados. La exactitud obtenida por el modelo alcanza un 98,70%.

B Base de Datos.

Para validar la propuesta de la investigación desarrollada en este artículo se empleó el conjunto de datos de flujo de red KDD-Cup99 [47], [48]. Este conjunto de datos está constituido por 44 atributos, donde 43 son atributos descriptivos y uno más para el atributo predictor, el cual está clasificado en dos tipos (normal-0, anomalía-1). El conjunto de datos está constituido por 9711 y 12833 muestras de comportamiento normal y anomalía respectivamente. Con el objetivo de experimentar en datos altamente desbalanceados el conjunto de datos original fue transformado hasta obtener una relación entre clases de 9710 y 971 muestras de comportamiento normal y anomalía, lo que representa un desbalance entre clases de 1/10.

La base de datos empleada en esta investigación, así como los resultados y el código de la experimentación pueden ser consultadas desde nuestro Github de materiales suplementarios[1].

C Diseño Experimental.

La detección de anomalías es un proceso complejo debido a que, en entornos reales, los ataques o cambios en el comportamiento de la red son muy pocos con respecto al comportamiento normal. De hecho el rango de falsas alarmas generados por los métodos puede llegar a ser un problema a tener en consideración [18]. La selección de un modelo predictivo para la detección de anomalías debe cumplir con índices elevados de aciertos y bajos índices de falsas alarmas. Es por ello que, para el desarrollo de la investigación, se siguió el diseño experimental que cuenta de los pasos siguientes:

1) Análisis de la capacidad de predicción de diferentes esquemas de combinación de múltiples clasificadores basados en árboles de decisión, mediante la comparación sus resultados en cuanto a las métricas Pr, Rc, f1.

2) En caso de que los resultados no sean altamente diferenciables realizar un análisis de sus curvas ROC y PRC, para determinar la influencia de los verdaderos y falsos positivos primeramente, unido al balance entre el Pr y el Rc.

3) Analizar el desempeño de los métodos seleccionados junto a métodos basados en aprendizaje profundo, mediante la comparación de sus curvas ROC y PRC, para determinar si constituyen una alternativa frente a técnicas avanzadas de predicción.

D Evaluación de los Resultados.

Los sistemas de detección de anomalías en redes de computadoras requieren de altos índices de detección correctos y bajos índices de falsas alarmas. En general, el desempeño de estos sistemas puede ser evaluado en términos de Precisión y Recuerdo (Recall) [22]. Con el objetivo de evaluar el comportamiento de los métodos seleccionados, se hace necesario contar con los estadígrafos que permitan medir el desempeño en cada uno de estos. Para lograr dicha tarea en todo momento se empleó el procedimiento de validación cruzada y fueron analizadas las magnitudes del Precisión (Pr), Recall (Rc), f1 [49], y las curvas ROC y PRC [50] de los métodos empleados.

III. Análisis y Discusión

Para determinar el poder de detección de anomalías de los esquemas de combinación de múltiples clasificadores basados en árboles de decisión se llevaron a cabo varias etapas descritas a continuación.

A Comparación de Esquemas.

Para analizar el desempeño de los diferentes modelos predictivos enunciados en la sección Materiales y métodos, se emplearon las métricas Precisión, Recall, f1-score. En la TABLA 1 se puede observar un resumen de tales métricas para los métodos j48, AdaBoost, Bagging, Extra-Tree, Random Forest y XGBoost. Con el objetivo de notar su capacidad predictiva fueron analizadas por separado en la tabla ambas clases del problema, comportamiento anomalía y normal.

TABLA 1. RESULTADOS DE LOS MÉTODOS J48, ADABOOST, BAGGING, EXTRA-TREE, RANDOM FOREST Y XGBOOST PARA LAS MÉTRICAS PRECISIÓN (PR), RECALL (RC) Y F1.

Clases ----->	**Anomalía (971)**			**Normal (9710)**		
Matos's/Métricas	**Pr**	**Rc**	**F1**	**Pr**	**Rc**	**F1**
J48	0.90	0.93	0.92	0.99	0.99	0.99
Random Forest	0.94	***0.95***	***0.94***	0.99	0.99	0.99
Bagging	0.93	0.94	***0.94***	0.99	0.99	0.99
Extra-Tree	*0.95*	0.93	***0.94***	0.99	***1.00***	0.99
AdaBoost	0.92	0.80	0.85	0.98	0.99	0.99
XGBoost	***0.97***	0.89	0.93	0.99	***1.00***	0.99

Como se puede observar en la TABLA 1, el desempeño de los métodos es diferente para ambas clases anomalía y normal,

[1]https://github.com/jcquintana87/myGit2Play

donde para la última mencionada el desempeño ronda el 99% de Precisión y Recall. Sin embargo, para la clase anomalía, los métodos alcanzan en general un promedio de 93%, 90% para ambas métricas respectivamente. Esto significa que con respecto al comportamiento normal en la red los métodos no cometen prácticamente falsas alarmas. Al contrario, para la clase anomalía, se dejan de predecir correctamente cerca de un 7% general de este comportamiento y no se recuerdan aproximadamente 10% de estas.

En detalle, el esquema de combinación que obtuvo el peor desempeño en cuanto a la Precisión para la base de datos empleada fue AdaBoost cuyo resultado fue de 92%, superando a un método vaselina (j48). Es necesario destacar que un 2% es relevante para este campo de investigación, donde se requiere de una alta confianza en las herramientas empleadas ya que se protegen datos de usuarios, infraestructura de red. Por otro lado el método de mejor desempeño fue XGBoost superando en un 2% al segundo método de mejor desempeño en cuanto a esta métrica, Extra-Tree. Donde con un 97% y 100% de precisión para ambas clases anomalía y normal, XGBoost muestra que puede ser una alternativa para la predicción de anomalías.

En cuanto a la métrica Recall, el método que obtuvo el mejor desempeño fue Random Forest, seguido de Bagging y Extra-Tree, lo que muestra la capacidad de recuerdo de esquemas de construcción de múltiples clasificadores. Aunque el método vaselina empleado logra un índice de recuerdo similar al de un esquema de construcción de múltiples clasificadores simple, el cual solamente aplica muestreo con remplazamiento. Esto sugiere que emplear árboles de decisión, incluso estrategias relativamente simples, puede garantizar recordar comportamientos anómalos en la red. Sin embargo, algoritmos más complejos como AdaBoost y XGBoost no superan el 90% de la capacidad de recuerdo, mostrando el primero de estos un índice de 80%. Este comportamiento puede estar relacionado con su estrategia de construcción, en la cual se aprende de manera adaptativa, asignando pesos a los ejemplos mal clasificados, para entonces en las iteraciones siguientes prestar mayor atención. Tomando en consideración que solamente se cuentan con 971 ejemplos de anomalías, debido a que con fines experimentales se deseaba obtener un conjunto de datos altamente desbalanceados, dichos métodos pueden estar siendo desfavorecidos frente a estrategias altamente aleatorias o estrategias más simples que con pocos datos logran modelar mejor el espacio de soluciones del problema.

B Selección de Estrategias.

Para seleccionar los esquemas de combinación adecuados para la detección de anomalías, y teniendo como precedente el estudio anterior, se tomaron en cuenta los esquemas que presentaron los mejores índices en cuento a Precisión y Recall. Se emplearon diferentes gráficos que describiesen su desempeño en cuanto a dichas métricas combinadas, lo que permite seleccionar la estrategia que menor nivel de falsas alarmas pueda cometer con respecto a ambas clases, comportamiento normal y anomalía. Similar a ROC, la curva Precisión-Recall (PRC) es una herramienta adecuada para mostrar y analizar el desempeño de diferentes algoritmos [50]. Especialmente es útil frente a conjuntos de datos altamente sesgados, donde puede ser muy informativa. El principal objetivo en la PRC es posicionarse lo más al tope en la esquina derecha del gráfico, lo que marca el mejor comportamiento. La Fig. 1 muestra la curva PRC para la clase normal arrojada para los métodos J48, Bagging, Random Forest, Extra-Tree, AdaBoost y XGBoost.

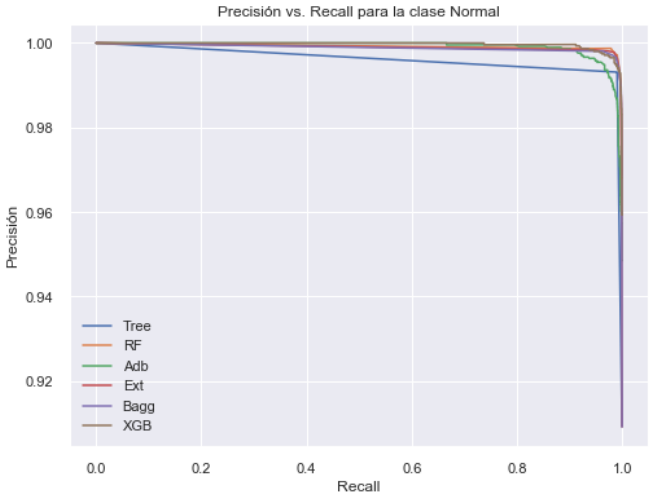

Fig. 1 Resultado de la curva PRC para la clase normal de los métodos J48 (Tree), Bagging (Bagg), Random Forest (RF), Extra-Tree (Ext), AdaBoost (Ada) y XGBoost (XGB).

En la Fig. 1 se puede observar que el comportamiento de la mayoría de los esquemas de combinación de múltiples clasificadores es similar para la clase normal. Solamente AdaBoost tiene un comportamiento inferior con respecto a la relación Precisión y Recall. Por otro lado, cuando con respecto a la clase anomalía, la Fig. 2 muestra la curva PRC.

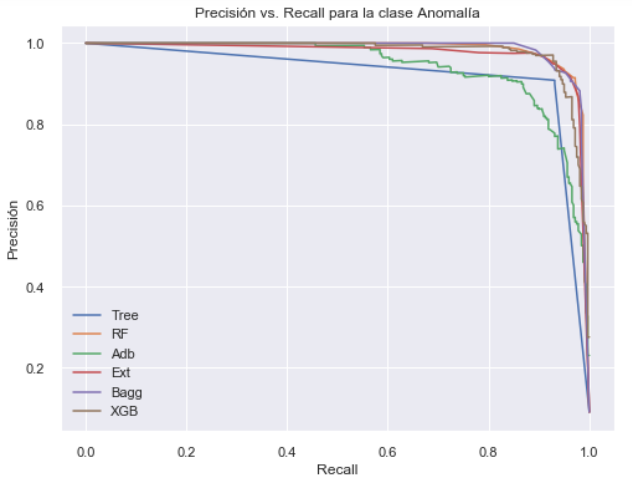

Fig. 1 Resultado de la curva PRC para la clase anomalía de los métodos J48 (Tree), Bagging (Bagg), Random Forest (RF), Extra-Tree (Ext), AdaBoost (Ada) y XGBoost (XGB)

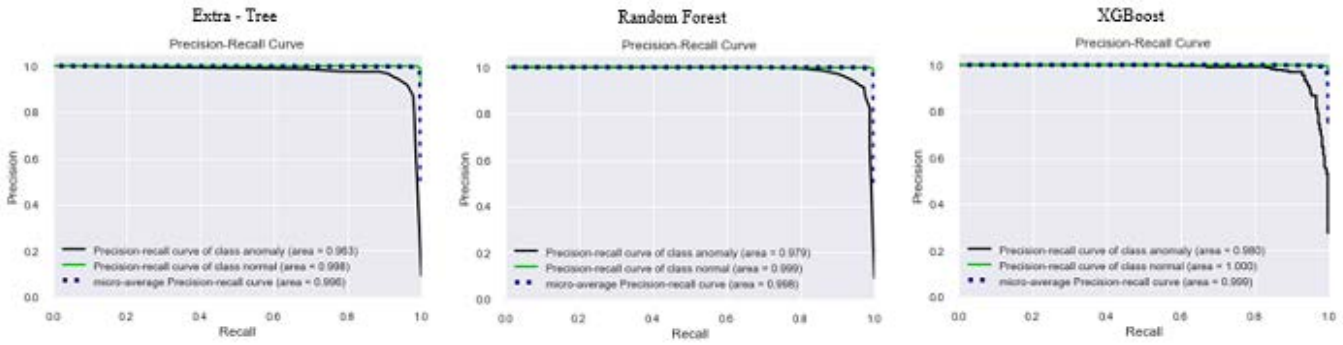

Fig. 2 Curvas Precisión vs Recall para los métodos Extra-Tree, Random Forest y XGBoost

En la Fig. 2 se puede observar que el comportamiento de la mayoría de los esquemas de combinación de múltiples clasificadores muestra mayores diferencias entre ellos. Aunque se mantiene con un desempeño inferior AdaBoost. Para analizar por separado los métodos que mejor se desempeñas se muestran sus gráficos para la curva PRC por separado.

La Fig. 3 muestra el desempeño PRC de los métodos Extra-Tree, RandomForest y XGBoost. Donde la línea negra, verde y discontinua, representan la clase anomalía, normal y el micro-average entre estas. Los valores para la clase anomalía muestran un índice de 96%, 97% y 98% para los métodos Extra-Tree, RandomForest y XGBoost respectivamente.

Por otro lado, AdaBoost muestra el peor desempeño de los esquemas de combinación de clasificadores. En la Fig. 4 se muestra el resultado de la curva PRC de dicho método.

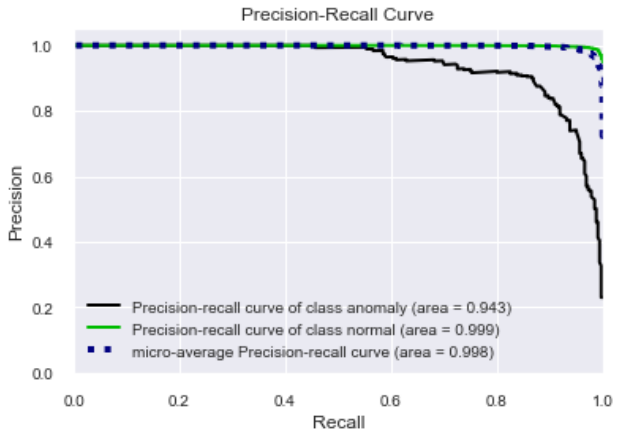

Fig.3 Curva Precisión vs Recall para el método AdaBoost. Las líneas verde, negra y discontinua representan las clases, normal, anomalía y el micro-average respectivamente.

En la Fig. 4 se observa que el desempeño obtenido por el método es de 94% para la clase anomalía. Aunque muestra el peor desempeño, tales niveles son similares a los obtenidos por métodos del estado-del-arte.

A pesar de los resultados previamente analizados no se puede llegar a una conclusión clara sobre cuál de los esquemas de combinación de múltiples clasificadores seleccionar o si todas constituyen una alternativa. Para determinar si existen diferencias entre el desempeño de los métodos, se requiere de pruebas de significación estadísticas [51], [52]. En este caso se aplica Friedman con su correspondiente corrección Iman & Davenport con un α=0.05. La TABLA 2 muestra el resultado de ambas pruebas.

TABLA 2.
PRUEBAS ESTADISTICAS DE FRIEDMAN E IMAN & DAVENPORT

Prueba	p-value	Hipótesis
Friedman	0.7605	Aceptada
Iman & Davenport	0.7605	Aceptada

Mediante estas pruebas se comprueba la hipótesis nula de que no existen diferencias entre el desempeño de los métodos por tanto cualquiera de estos puede ser empleado para la predicción de anomalías. Dicha hipótesis que fue aceptada, con un *p-value* de 0,83. Por tal razón a pesar de los resultados mostrados en los gráficos, cualquiera de los esquemas ya sea Random Forest, Extra-Tree, XGBoost y AdaBoost pueden ser empleados para la predicción de anomalías.

C Comparación con Tendencias.

Las arquitecturas de aprendizaje profundo (*Deep Learning*, DL), están compuestas por múltiples niveles de operaciones no-lineales, muy similar a como se estructuran las capas de las redes neuronales [53]. Esta forma de procesamiento por capas permite a los métodos aprender jerarquías de rasgos desde los niveles superiores hasta los niveles inferiores de representación. De esta forma, automáticamente se pueden aprender rasgos a múltiples niveles de abstracción permitiendo a los modelos aprender complejas funciones de mapeo en los datos. Al final, este tipo de arquitectura de aprendizaje puede procesar información para la extracción y transformación de rasgos, supervisado y no-supervisado, además es altamente competitiva en el análisis y clasificación de patrones con respecto a los modelos presentes en el estado-del-arte [54], [55], [56], [57]. Las arquitecturas DL se han aplicado en diferentes campos de investigación, tales como el tratamiento de imágenes [58], clasificación de audio y video [59], en la Bioinformática [60] y en el análisis de sentimientos y texto [61], [56], solo por

mencionar.

En la ciber-seguridad han sido empleadas en diversos casos [28], [4], [3]. Un método para analizar anomalías en el tráfico de la red mediante aprendizaje profundo es desarrollado en [22]. En este trabajo solo se seleccionan seis rasgos de 41 existentes en el conjunto de datos y se emplea una red profunda simple (cuenta con varias capas completamente conectadas). Al obtener cerca de un 75.75% de exactitud en la predicción de anomalías, los autores del trabajo concluyen que el aprendizaje profundo puede ser una vía alternativa para la detección de anomalías e incluso empleando pocos rasgos.

Para la experimentación se construyó una red neuronal totalmente conectada, implementada mediante el *framework* Keras, en Python 3.7. La red cuenta con tres capas de neuronas, una de entrada, una oculta y una de salida. El método de entrenamiento empleado fue un Gradiente Descendente Estocástico (*Stochastic Descending Gradient*, SGD) con un rango de aprendizaje (Learning rate) de 0.01, durante 1000 épocas. La computadora donde fue desarrollada la experimentación fue una Intel Core i-5 primera generación, con 3 G de Ram. Para realizar la comparación fueron empleados gráficos de barras, las curvas ROC y PRC y finalmente una prueba de significación estadística. La Fig. 5 muestra la arquitectura de la red.

Layer (type)	Output Shape	Param #
dense_14 (Dense)	(None, 512)	19968
dropout_9 (Dropout)	(None, 512)	0
dense_15 (Dense)	(None, 64)	32832
dropout_10 (Dropout)	(None, 64)	0
dense_16 (Dense)	(None, 2)	130

Total params: 52,930
Trainable params: 52,930
Non-trainable params: 0

Fig. 5 Arquitectura de la red neuronal implementada con Keras en Python 3.7.

Para analizar el desempeño de los esquemas de combinación de múltiples clasificadores basados en árboles de decisión y métodos basados en aprendizaje profundo fue empleado el mismo conjunto de datos. Para el entrenamiento de la red neuronal, el conjunto de datos fue normalizado, basándose en la media y la desviación estándar. La Fig. 6 muestra el desempeño mediante gráficos de barras para las métricas Precisión, Recall y F1 de la red neuronal implementada y XGBoost para la clase anomalía.

En la Fig. 6 se puede observar las tres métricas, Precisión, Recall y F1 para ambos métodos comparados. Donde en cuanto a la métrica precisión, ambos métodos se comportan de manera similar, obteniendo un 97% promedio para el conjunto de datos empleado. Dicho resultado está en concordancia con los valores para está métrica obtenidos por los métodos del estado-del-arte. Tales valores de precisión sugieren que ambos métodos pueden ser aplicados como alternativas, resaltando la alta capacidad generalizadora de las redes neuronales. No obstante, un esquema de combinación de árboles d decisión también se muestra como una alternativa dado ese resultado. Alternativa que es viable dada la facilidad de implementación y entrenamiento de estos. Ya que pueden ser fácilmente adaptados a rasgos nuevos sin la necesidad de realizar grandes cambios.

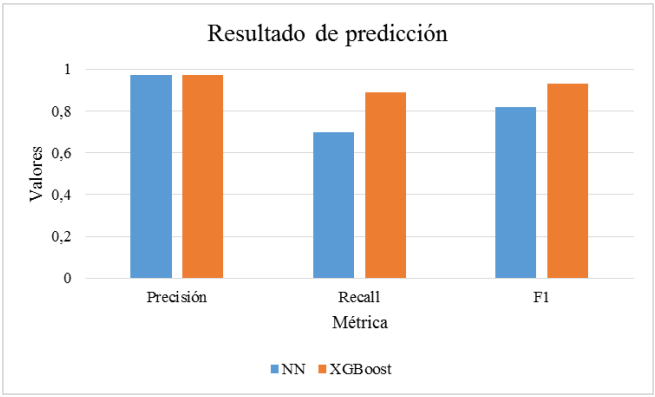

Fig. 6 Resultados del desempeño para la clase anomalía de la Red neuronal (NN) y XGBoost

En cuanto a la métrica Recall, si existe una diferencia marcada, donde XGBoost y NN obtienen un 89% y 70% respectivamente. Como se puede observar existe cerca de un 9% de diferencia entre ambos paradigmas de aprendizaje automatizado. Lo que sugiere que el esquema de combinación de clasificadores adaptativo logra recordar mejor las anomalías. Es necesario resaltar que XGBoost fue el de mejor precisión con respecto al resto de los esquemas basados en árboles, pero fue uno de los que obtuvo el menor desempeño en cuanto a recuerdo. Esto sugiere que la red neuronal, puede llegar a comportarse para los datos empleados en la experimentación peor que Random Forest y Extra-Tree.

Como parte del análisis para medir el desempeño fueron seleccionados los gráficos de curvas ROC y PRC de ambos modelos predictivos, los cuales permiten observar el balance entre los falsos y verdaderos positivos por un lado, y el balance entre Precisión y Recall por otro lado, respectivamente. El gráfico *Receiver Operating Characteristics* (ROC) puede ser empleado para seleccionar clasificadores y visualizar su desempeño [62]. Según el estado-del-arte, ROC posee propiedades que lo hacen especialmente útil para trabajar en dominios de datos donde la distribución y costo de error de predicción de las clases sea diferente. La Figura 7 y Figura 8 muestran la curva ROC para la red neuronal creada y XGBoost, separadas por clases, comportamiento anomalía y normal.

En la Fig. 7 se observa un gráfico combinado para las curvas ROC referentes a las clases Anomalía, Normal, el micro y macro-average para ambas clases. Donde el valor obtenido para la clase anomalía y normal es de 98%, mostrando un buen desempeño. Por otro lado, en la Fig. 8 se muestra la curva ROC

para XGBoost, donde se obtiene 99% para ambas clases. Valor que es en 1% superior a la red neuronal.

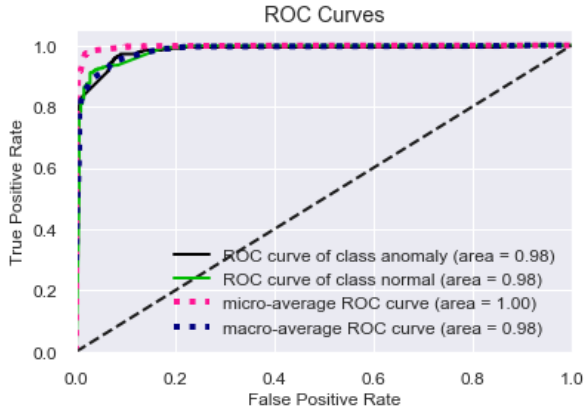

Fig. 7 Curva ROC resultante de la red neuronal.

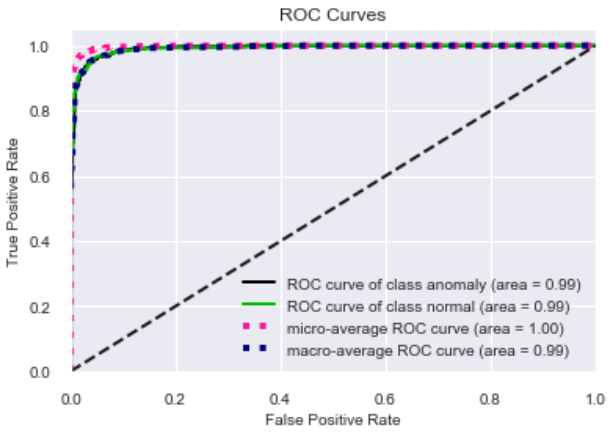

Fig. 8 Curva ROC resultante de XGBoost.

Cuando analizamos los gráficos PRC, la Fig. 9 muestra el desempeño de la red neuronal, donde se puede observar que la curva para la clase anomalía no presenta un comportamiento suave, y se obtiene un valor de 91%.

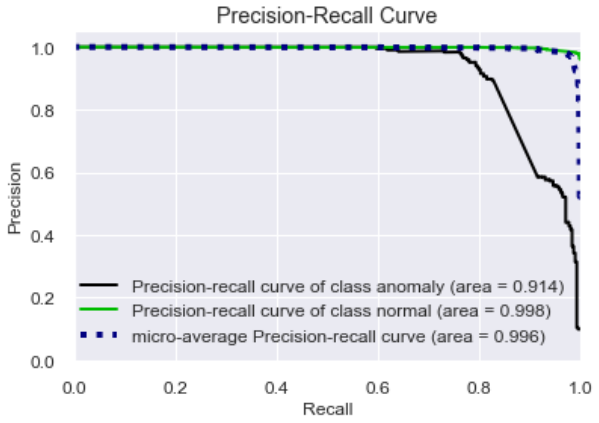

Fig. 9 Curva PRC resultante de la red neuronal.

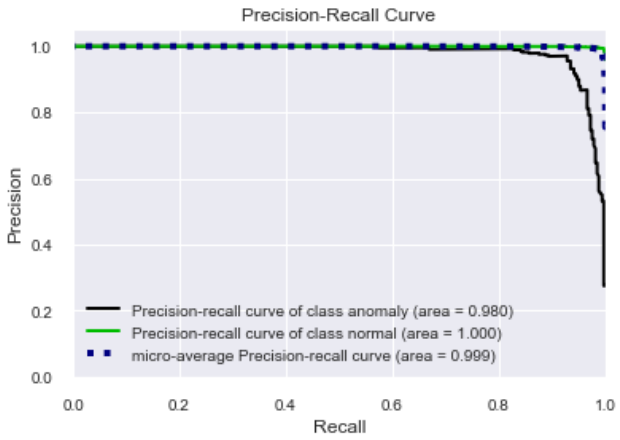

Fig. 10 Curva PRC resultante de XGBoost.

Por otro lado, en la Fig. 10, se muestran las curvas PRC para las clases anomalía y normal obtenidas por el modelo XGBoost, donde se obtiene un 98% para la clase anomalía, cerca de un 7% superior al valor obtenido por la red neuronal, mostrado de esta forma un mejor desempeño con respecto al modelo conexionista.

Para finalizar con la comparación entre ambos modelos predictivos se llevó a cabo un análisis estadístico para determinar si existen diferencias significativas entre el desempeño de ambos métodos. Para lograr tal tarea, se tomaron los valores de las predicciones realizadas en el conjunto de datos y se compararon mediante una prueba estadística de *Wilcoxon Signed Ranks*, con un α=0.05. Como resultado de la prueba se obtuvo que existen diferencias significativas con un p-value[2] menor que 0.05, por tanto el desempeño de XGBoost es estadísticamente superior a la red neuronal. De esta forma se corrobora que los esquemas de combinación de clasificadores basados en árboles de decisión pueden constituir una alternativa para la detección de anomalías, ya que son competitivos con algunos de los mejores algoritmos del estado-del-arte.

IV. Conclusiones

En el presente artículo se muestra un estudio de la capacidad de detección de anomalías de diferentes esquemas de combinación de clasificadores basados en árboles de decisión. Se pudo constatar que diferentes estrategias de construcción múltiples clasificadores tales como adaptativas XGBoost y muy aleatorias como Random Forest y Extra-Tree presentan comportamiento estadísticamente similar, lo que sugiere que diferentes formas de construcción pueden ser una alternativa. Además, una comparación con respecto a un método basado en aprendizaje profundo muestra la competitividad de los esquemas de combinación de clasificadores basados en árboles de decisión. En general para los métodos empleados en la experimentación, la Precisión y Recall obtenidos rondan los 97% y 99% respectivamente. Dichos valores que son similares a los mostrados por los métodos del estado-del-arte.

[2] p-value = 6.995847403238189e-13

AGRADECIMIENTOS

Los autores expresan y agradecen el enorme apoyo de la Facultad de Informática y Ciencias Exactas de la Universidad "Máximo Gómez Báez" de Ciego de Ávila y de la Universidad Central "Marta Abreu" de las Villas, Cuba.

V. REFERENCIAS

[1] F. Hussain, R. Hussain, S. A. Hassan, y E. Hossain, «Machine Learning in IoT Security: Current Solutions and Future Challenges», *ArXiv190405735 Cs Stat*, mar. 2019, Accedido: nov. 14, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1904.05735.

[2] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, y L. T. Yang, «Data Mining for Internet of Things: A Survey», *IEEE Commun. Surv. Tutor.*, vol. 16, n.º 1, pp. 77-97, 2014, doi: 10.1109/SURV.2013.103013.00206.

[3] B. Dong y X. Wang, «Comparison deep learning method to traditional methods using for network intrusion detection», en *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, China, 2016, pp. 581-585, doi: 10.1109/ICCSN.2016.7586590.

[4] H. Liu y B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey.pdf», *Appl. Sci.*, sep. 2019, doi: doi:10.3390/app9204396.

[5] D. E. Denning, «An Intrusion-Detection Model», *IEEE Trans. Softw. Eng.*, vol. VOL. SE-13, n.º NO. 2, pp. 222-232, feb. 1987.

[6] S. Agrawal y J. Agrawal, «Survey on Anomaly Detection using Data Mining Techniques», *Procedia Comput. Sci.*, vol. 60, pp. 708-713, 2015, doi: 10.1016/j.procs.2015.08.220.

[7] H. A. Nguyen y D. Choi, «Application of Data Mining to Network Intrusion Detection: Classifier Selection Model», en *Challenges for Next Generation Network Operations and Service Management*, vol. 5297, Y. Ma, D. Choi, y S. Ata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 399-408.

[8] H. Altwaijry y S. Algarny, «Bayesian based intrusion detection system», *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 24, n.º 1, pp. 1-6, ene. 2012, doi: 10.1016/j.jksuci.2011.10.001.

[9] R. F. Fouladi, C. E. Kayatas, y E. Anarim, «Frequency based DDoS attack detection approach using naive Bayes classification», en *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, Vienna, Austria, jun. 2016, pp. 104-107, doi: 10.1109/TSP.2016.7760838.

[10] F. E. Heba, A. Darwish, A. E. Hassanien, y A. Abraham, «Principle components analysis and Support Vector Machine based Intrusion Detection System», en *2010 10th International Conference on Intelligent Systems Design and Applications*, Cairo, Egypt, nov. 2010, pp. 363-367, doi: 10.1109/ISDA.2010.5687239.

[11] V. Kosamkar y S. S. Chaudhari, «Improved Intrusion Detection System using C4.5 Decision Tree and Support Vector Machine», University of Mumbai, India, Dissertation, 2013.

[12] S. Teng, N. Wu, H. Zhu, L. Teng, y W. Zhang, «SVM-DT-based adaptive and collaborative intrusion detection», *IEEECAA J. Autom. Sin.*, vol. 5, n.º 1, pp. 108-118, ene. 2018, doi: 10.1109/JAS.2017.7510730.

[13] C. Chen, Y. Gong, y Y. Tian, «Semi-supervised learning methods for network intrusion detection», en *2008 IEEE International Conference on Systems, Man and Cybernetics*, Singapore, Singapore, oct. 2008, pp. 2603-2608, doi: 10.1109/ICSMC.2008.4811688.

[14] Y. Li, B. Fang, L. Guo, y Y. Chen, «Network anomaly detection based on TCM-KNN algorithm», en *Proceedings of the 2nd ACM symposium on Information, computer and communications security - ASIACCS '07*, Singapore, 2007, p. 13, doi: 10.1145/1229285.1229292.

[15] H. F. Eid, M. A. Salama, A. E. Hassanien, y T. Kim, «Bi-Layer Behavioral-Based Feature Selection Approach for Network Intrusion Classification», en *Security Technology*, vol. 259, T. Kim, H. Adeli, W. Fang, J. G. Villalba, K. P. Arnett, y M. K. Khan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 195-203.

[16] H. Chae, B. Jo, S.-H. Choi, y T. Park, «Feature Selection for Intrusion Detection using NSL-KDD», en *Recent Advances in Computer Science*, 2013, p. 4.

[17] N. Farnaaz y M. A. Jabbar, «Random Forest Modeling for Network Intrusion Detection System», *Procedia Comput. Sci.*, vol. 89, pp. 213-217, 2016, doi: 10.1016/j.procs.2016.06.047.

[18] S. Dhaliwal, A.-A. Nahid, y R. Abbas, «Effective Intrusion Detection System Using XGBoost», *Information*, vol. 9, n.º 7, p. 149, jun. 2018, doi: 10.3390/info9070149.

[19] J. Kim, J. Kim, H. L. T. Thu, y H. Kim, «Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection», en *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, feb. 2016, pp. 1-5, doi: 10.1109/PlatCon.2016.7456805.

[20] B. J. Radford, L. M. Apolonio, A. J. Trias, y J. A. Simpson, «Network Traffic Anomaly Detection Using Recurrent Neural Networks», *ArXiv180310769 Cs*, mar. 2018, Accedido: nov. 13, 2019. [En línea]. Disponible en: http://arxiv.org/abs/1803.10769.

[21] T. Ma, F. Wang, J. Cheng, Y. Yu, y X. Chen, «A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks», *Sensors*, vol. 16, n.º 10, p. 1701, oct. 2016, doi: 10.3390/s16101701.

[22] M. T. Tang, D. L. Mhamdi, y D. D. McLernon, «Deep Learning Approach for Network Intrusion Detection in Software Defined Networking», *IEEE*, p. 6, 2016, doi: 978-1-5090-3837-4.

[23] C. Min, X. Qian, L. Jianming, L. Wenyin, L. Qing, y W. Jianping, «MS-LSTM: A multi-scale LSTM model for BGP anomaly detection», en *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, Singapore, nov. 2016, pp. 1-6, doi: 10.1109/ICNP.2016.7785326.

[24] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, y M. Sheikhan, «Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm», en *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*, Philadelphia, PA, USA, jul. 2013, pp. 76-81, doi: 10.1109/ICDCSW.2013.40.

[25] R. A. Sadek, M. S. Soliman, y H. S. Elsayed, «Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction», *Int. J. Comput. Sci. Issues*, vol. 10, n.º 6, p. 7, 2013.

[26] B. L. P. Lourdes y T. A. L. Santiago, «Desarrollo de un algoritmo de redes neuronales artificiales aplicado a la predicción de tráfico de la infraestructura de comunicaciones de redes corporativas», ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, Riobamba-Ecuador, 2018.

[27] Á. Martín, J. M. Álvarez, y J. M. Gómez, «Detección de anomalías en red utilizando técnicas de Machine Learning», Grado de Ingeniería Informática, Universidad Carlos III de Madrid, España, 2017.

[28] K. Soman y M. Alazab, «A Comprehensive Tutorial and Survey of Applications of Deep Learning for Cyber Security», 2020.

[29] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, y A. E. Hassanien, «Hybrid Intelligent Intrusion Detection Scheme», en *Soft Computing in Industrial Applications*, vol. 96, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, y L. Costa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 293-303.

[30] A. Javaid, Q. Niyaz, W. Sun, y M. Alam, «A Deep Learning Approach for Network Intrusion Detection System», en *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, New York City, United States, 2016, doi: 10.4108/eai.3-12-2015.2262516.

[31] L. Breiman, «Bagging predictors», *Mach. Learn.*, vol. 24, n.º 2, pp. 123-140, 1996.

[32] D. P. Gaikwad y R. C. Thool, «Intrusion Detection System Using Bagging with Partial Decision TreeBase Classifier», *Procedia Comput. Sci.*, vol. 49, pp. 92-98, 2015, doi: 10.1016/j.procs.2015.04.231.

[33] L. Breiman, «Random forests», *Mach. Learn.*, vol. 45, n.º 1, pp. 5-32, 2001.

[34] A. Dehzangi, S. Phon-Amnuaisuk, y O. Dehzangi, «Using Random Forest for Protein Fold Prediction Problem: An Empirical Study.», *J Inf Sci Eng*, vol. 26, n.º 6, pp. 1941-1956, 2010.

[35] Y. Li, Y. Fang, y J. Fang, «Predicting residue-residue contacts using random forest models», *Bioinformatics*, vol. 27, n.º 24, pp. 3379-3384, dic. 2011, doi: 10.1093/bioinformatics/btr579.

[36] N. F. F. da Silva, E. R. Hruschka, y E. R. Hruschka, «Tweet sentiment analysis with classifier ensembles», *Decis. Support Syst.*, vol. 66, pp. 170-179, oct. 2014, doi: 10.1016/j.dss.2014.07.003.

[37] Y. Zhao y Y. Zhang, «Comparison of decision tree methods for finding active objects», *Adv. Space Res.*, vol. 41, n.º 12, pp. 1955-1959, 2008.

[38] P. Geurts, D. Ernst, y L. Wehenkel, «Extremely randomized trees», *Mach. Learn.*, vol. 63, n.º 1, pp. 3-42, abr. 2006, doi: 10.1007/s10994-006-6226-1.

[39] F. Almaguer-Angeles, J. Murphy, L. Murphy, y A. O. Portillo-Dominguez, «Choosing Machine Learning Algorithms for Anomaly Detection in Smart Building IoT Scenarios», en *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, abr. 2019, pp. 491-495, doi: 10.1109/WF-IoT.2019.8767357.

[40] O. Manzanilla-Salazar, F. Malandra, H. Mellah, C. Wette, y B. Sanso, «A Machine Learning framework for Sleeping Cell Detection in a Smart-city IoT Telecommunications Infrastructure», *ArXiv191001092 Cs Eess*, feb. 2020, Accedido: feb. 25, 2020. [En línea]. Disponible en: http://arxiv.org/abs/1910.01092.

[41] Y. Freund y R. E. Schapire, «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting», *J. Comput. Syst. Sci.*, vol. 55, n.º 1, pp. 119-139, ago. 1997, doi: 10.1006/jcss.1997.1504.

[42] A. Yulianto, P. Sukarno, y N. A. Suwastika, «Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset», *J. Phys. Conf. Ser.*, vol. 1192, p. 012018, mar. 2019, doi: 10.1088/1742-6596/1192/1/012018.

[43] M. Mazini, B. Shirazi, y I. Mahdavi, «Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms», *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, n.º 4, pp. 541-553, oct. 2019, doi: 10.1016/j.jksuci.2018.03.011.

[44] T. Chen y C. Guestrin, «XGBoost: A Scalable Tree Boosting System», en *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, San Francisco, California, USA, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.

[45] R. Mitchell y E. Frank, «Accelerating the XGBoost algorithm using GPU computing», *PeerJ Comput. Sci.*, vol. 3, p. e127, jul. 2017, doi: 10.7717/peerj-cs.127.

[46] J. Wang, B. Li, y Y. Zeng, «XGBoost-Based Android Malware Detection», en *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, dic. 2017, pp. 268-272, doi: 10.1109/CIS.2017.00065.

[47] K. Siddique, Z. Akhtar, F. Aslam Khan, y Y. Kim, «KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research», *Computer*, vol. 52, n.º 2, pp. 41-51, feb. 2019, doi: 10.1109/MC.2018.2888764.

[48] A. Divekar, M. Parekh, V. Savla, R. Mishra, y M. Shirole, «Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives», *2018 IEEE 3rd Int. Conf. Comput. Commun. Secur. ICCCS*, pp. 1-8, oct. 2018, doi: 10.1109/CCCS.2018.8586840.

[49] D. M. Powers, «Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation», 2011.

[50] J. Davis y M. Goadrich, «The relationship between Precision-Recall and ROC curves», en *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240, Accedido: oct. 19, 2015. [En línea]. Disponible en: http://dl.acm.org/citation.cfm?id=1143874.

[51] J. Demšar, «Statistical comparisons of classifiers over multiple data sets», *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[52] B. Calvo y G. Santafé, «scmamp: Statistical Comparison of Multiple Algorithms in Multiple Problems», *R J.*, vol. 8, n.º 1, p. 248, 2016, doi: 10.32614/RJ-2016-017.

[53] R. Vargas, A. Mosavi, y R. Ruiz, «Deep Learning: A Review», MATHEMATICS & COMPUTER SCIENCE, preprint, oct. 2018. doi: 10.20944/preprints201810.0218.v1.

[54] Y. Bengio, «Learning Deep Architectures for AI», *Found. Trends® Mach. Learn.*, vol. 2, n.º 1, pp. 1-127, 2009, doi: 10.1561/2200000006.

[55] L. Deng, «Deep Learning: Methods and Applications», *Found. Trends® Signal Process.*, vol. 7, n.º 3-4, pp. 197-387, 2014, doi: 10.1561/2000000039.

[56] Y. LeCun, Y. Bengio, y G. Hinton, «Deep learning», *Nature*, vol. 521, n.º 7553, pp. 436-444, may 2015, doi: 10.1038/nature14539.

[57] J. Schmidhuber, «Deep learning in neural networks: An overview», *Neural Netw.*, vol. 61, pp. 85-117, 2015.

[58] T.-W. Huang *et al.*, «Age estimation from brain MRI images using deep learning», en *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, Melbourne, Australia, abr. 2017, pp. 849-852, doi: 10.1109/ISBI.2017.7950650.

[59] L. Deng y J. C. Platt, «Ensemble deep learning for speech recognition», en *Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH)*, 2014, Accedido: oct. 19, 2015. [En línea]. Disponible en: http://193.6.4.39/~czap/letoltes/IS14/IS2014/PDF/AUTHOR/IS140245.PDF.

[60] P. D. Lena, K. Nagata, y P. F. Baldi, «Deep spatio-temporal architectures and learning for protein structure prediction», en *Advances in Neural Information Processing Systems*, 2012, pp. 512–520, Accedido: oct. 13, 2015. [En línea].

[61] N. Mohamed Ali, M. M. A. El Hamid, y A. Youssif, «Sentiment analysis for movies reviews dataset using deep learning models», *Int. J. Data Min. Knowl. Manag. Process*, vol. 09, n.º 03, pp. 19-27, may 2019, doi: 10.5121/ijdkp.2019.9302.

[62] T. Fawcett, «An introduction to ROC analysis», *Pattern Recognit. Lett.*, vol. 27, n.º 8, pp. 861-874, jun. 2006, doi: 10.1016/j.patrec.2005.10.010.