Empirical Exploration of Machine Learning Techniques for Detection of Anomalies Based on NIDS

Diego Vallejo-Huanga*†, Marco Ambuludi, and Paulina Morillo, *Member, IEEE*

***Abstract*—Computer crimes and attacks on data networks have increased significantly, so it has become necessary to implement techniques that detect these threats and safeguard the information of organizations. Network Intrusion Detection Systems (NIDS) allow detecting anomalies and attacks in real time, by analyzing the local and outgoing traffic of the network. At present, to improve its performance, it has been chosen to use Machine Learning (ML) techniques that automate these processes and improve the detection of an anomaly. This paper implements ML techniques through the use of datasets, in the context of a NIDS, for the detection and prediction of anomalies on networks. Tests were performed with non-supervised and supervised learning algorithms on NSL-KDD and UNSW-NB15 datasets. An exploratory analysis of data together with dimensionality reduction techniques allowed us to understand the nature of the data, prior to the modeling. The results show that the methodology can be extrapolated for real scenarios with different network configurations.**

***Index Terms*—PCA, SVM, ANN, Logistic Regression, Random Forest.**

I. Introducción

El desarrollo de aplicaciones, servicios web y transacciones electrónicas, se ha incrementado debido a que las organizaciones despliegan gran parte de su infraestructura sobre Internet. Esto último crea un escenario de riesgo potencial para ataques que podrían comprometer datos o recursos estratégicos de la organización y de las personas. Frente a esta problemática han surgido varias herramientas que buscan incrementar la seguridad de los sistemas informáticos [1] [2] [3] , sin embargo, la gran mayoría tienden a funcionar bajo reglas y configuraciones determinadas por los administradores, incrementando la posibilidad de cometer errores, debido al factor humano [4] [5] [6].

Por otro lado, el *Machine Learning* (ML) ha hecho posible la automatización de diversos procesos en el campo de la ingeniería, las ciencias sociales, administrativas y ciencias médicas [7]; y es de gran utilidad cuando se desean realizar multitareas automatizadas. Según un informe realizado por Google Cloud a 375 empresas sobre el uso de *Machine Learning*, más del 50% lo usan para el análisis de datos con el fin de obtener una ventaja competitiva [8].

Un Sistema de Detección de Intrusos en la Red (NIDS), por sus siglas en inglés, monitorea el tráfico de la red en busca de actividades sospechosas que, por lo general, son detectadas a través de reglas definidas o patrones de comportamiento conocidos. Un NIDS es considerado la primera línea de defensa en la red, ya que inspecciona los paquetes entrantes, tráfico saliente y tráfico local, en busca de patrones sospechosos con el propósito de categorizar e identificar el tipo de ataque que se encuentra en curso [9]. Bajo esta premisa un NIDS podría trabajar en complemento con algoritmos de aprendizaje automático para lograr una mayor precisión y velocidad en la detección.

La predicción de amenazas relacionadas a datos obtenidos por medio de Sistemas de Detección de Intrusos (IDS), ayuda a las organizaciones a crear planes y mejoras en las políticas de seguridad tecnológica. En muchas ocasiones dichas políticas permiten filtrar el tráfico anómalo además de reconocer y evitar ataques en tiempo real [10]. Las técnicas de aprendizaje automático enfocadas a la predicción de anomalías en un NIDS, han sido exploradas desde el paradigma supervisado [11]. Este artículo abordará la detección de ataques en una red, mediante el uso de técnicas de aprendizaje supervisado y no supervisado.

A. Trabajos Relacionados

Un *dataset* o conjunto de datos es una colección de registros que son tratados colectivamente como una unidad y se caracterizan por estar agrupados, contenidos y relacionados [12] y son la entrada de cualquier algoritmo de aprendizaje automático. A continuación, se detallan los *datasets* utilizados más frecuentemente en tareas de *Machine Learning* y NIDS.

1) DARPA 1998: el Grupo de Tecnología de Sistemas de Información, del Instituto Tecnológico de Massachusetts, en conjunto con la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA) y el Laboratorio de Investigación de las Fuerzas Aéreas (AFRL), recopiló el primer *dataset* que contiene tráfico de red con una amplia colección de registros. Sus datos fueron recolectados durante nueve semanas, donde cada registro de conexión se encuentra etiquetado como normal o anormal, considerando cuatro tipos de ataques: Denegación de Servicio (DoS), Remoto a Local (R2L), Usuario a Root (U2R) y Acceso No Autorizado [13].

Diego Vallejo-Huanga, *Universidad Politécnica Salesiana, IDEIAGEOCA Research Group, Quito, Ecuador, dvallejoh@ups.edu.ec. † Universidad de las Américas, Department of Physics and Mathematics, Quito, Ecuador, diego.vallejo.huanga@udla.edu.ec.

Marco Ambuludi, Universidad Politécnica Salesiana, Quito, Ecuador, mambuludit@est.ups.edu.ec

Paulina Morillo, Universidad Politécnica Salesiana, IDEIAGEOCA Research Group, Quito, Ecuador, pmorillo@ups.edu.ec.

2) KDD'99: construido con instancias del conjunto de datos DARPA 1998, consta de 4900000 registros, etiquetados como normales o anormales. Sus variables pueden ser clasificadas en dos grupos:

(i) Características básicas: encapsulan los atributos que son extraídos desde una conexión TCP/IP.

(ii) Características de tráfico: se computan con respecto a un intervalo de tiempo.

Este conjunto de datos presenta algunos problemas, ya que no se realizó una validación analítica ni experimental, por lo tanto, los datos resultantes en ocasiones no son similares al tráfico real. Además, no se realizó un test para verificar la existencia de paquetes descartados, razón por la cual se presentan valores perdidos y datos duplicados [14].

3) NSL-KDD: es un conjunto de datos que resolvió algunos de los problemas inherentes al *dataset* KDD'99, principalmente se eliminaron instancias duplicadas. Esta nueva versión del conjunto de datos se ha convertido en un *dataset* de referencia, para ayudar a los investigadores a comparar diferentes métodos de detección de intrusos [15].

En el conjunto de datos NSL-KDD existe un atributo que indica si una instancia de conexión, es normal o no, mientras que las 41 variables restantes han sido categorizadas en cuatro grupos [16]:

(i) Variables básicas (B): variables de conexiones TCP individuales.

(ii) Variables de contenido (C): atributos dentro de una conexión sugerida por el conocimiento del dominio.

(iii) Variables de tráfico (T): atributos calculados bajo una ventana temporal de dos segundos

(iv) Variables de host (H): atributos que evalúan ataques que tardan más de dos segundos.

4) UNSW-NB15: conjunto de datos creado en 2015, por la herramienta IXIA PerfectStorm, en el laboratorio Cyber Range, del Centro Australiano de Seguridad Cibernética (ACCS) [17] con el objetivo de generar un híbrido de actividades normales y comportamientos sintéticos de ataques. Es uno de los *datasets* más recientes, desarrollado con fines investigativos enfocados al análisis de tráfico en redes .

Los registros incluyen nueve tipos de ataques modernos en comparación con los cuatro que tiene el conjunto de datos KDD'99. Los nueve tipos de ataques son: reconocimiento, *shellcode*, *exploit*, *fuzzers*, gusano, DoS, *backdoor*, análisis y genéricos. Por otro lado, sus 49 variables pueden ser categorizadas como nominales, básicas, de contenido o de tiempo [18]. Las variables 1 a 35 representan la información adquirida de los paquetes de datos, las características 36 a 40 son consideradas de propósito general, mientras que desde la variable 41 a la 47 representan las características de conexión. Finalmente, la variable 48 representa el tipo de ataque y la variable 49 codifica al tipo de ataque, con 0 si es un registro normal y 1 si no lo es.

Existen varios trabajos que usan técnicas de *Machine Learning* para la detección de ataques en NIDS. En [19], por ejemplo, se utilizó el conjunto de datos NSL-KDD para el modelamiento de técnicas de ML. Con el fin de reducir la dimensionalidad del conjunto de datos, previo a la aplicación de los algoritmos de aprendizaje automático, se utilizó una selección de características basadas en correlación, obteniendo una reducción a 13 variables de las 42 originales. Se realizaron experimentos de clasificación utilizando seis técnicas de ML, donde el clasificador que obtuvo el mejor rendimiento fue el algoritmo *Random Forest* (RF) [20] y el peor fue Naïve Bayes [21].

En [22] se expone la importancia de las Redes Neuronales Artificiales (ANN) como técnica de clasificación en la detección de intrusos, ya que permiten el procesamiento eficaz de información, detección de ataques en tiempo real y capacidad de procesamiento con alta precisión, incluso cuando los datos de entrada están incompletos o son imprecisos. Además, se detalla un modelo de predicción de ataques, cuyo funcionamiento consiste en el uso de los registros *tcpdump* del *dataset* de entrenamiento DARPA 1998 con una exactitud (*accuracy*) de tan solo el 24%.

El conjunto de datos KDD'99 fue usado en [23] para encontrar métodos de pre-procesamiento de datos utilizando ANN y Funciones de Base Radial (RBF) [24]. En la experimentación se codificaron las variables cualitativas (TCP, ICMP o UDP) en cuantitativas. Esta investigación muestra la importancia del pre-procesamiento de los datos, antes de la aplicación de una o más técnicas de *Machine Learning*.

El trabajo realizado en [25] busca la identificación de intrusos en NIDS, con el uso del conjunto de datos NSL-KDD. Se realizó un pre-procesamiento de los datos para la conversión de variables cualitativas a numéricas y se redujo la dimensionalidad de las variables del *dataset*, mediante la evaluación de ganancia de atributos y algoritmos de correlación, obteniendo un total de 29 variables de 42 existentes. Este artículo detalla el proceso de clasificación mediante redes neuronales. Los resultados alcanzaron un 81.2% de exactitud para la detección de una intrusión, y un 79.9% de exactitud en la clasificación por tipo de ataque específico.

Chandrasekhar en [26] propuso distintos modelos para la detección de intrusos en la red con el uso de distintas técnicas de ML. Para la experimentación utilizó el *dataset* KDD'99, y realizó una reducción de variables a 34 con el objetivo de disminuir el tiempo de procesamiento computacional. En la clasificación se utilizaron todas las instancias del *dataset* por cada ataque (DoS, PROBE, R2L, U2R), sin embargo, para el tipo de ataque DoS se trabajó con una porción reducida, para mejorar el desempeño. Los resultados finales obtuvieron niveles sobre el 98% de exactitud de detección para cada tipo de ataque.

En [27] se realizó una evaluación en la detección de intrusos en la red, mediante el uso de ANN, sobre el *dataset* KDD'99. Este trabajo se llevó a cabo en distintas fases. En la fase de preprocesamiento de los datos, se convirtieron todos los valores categóricos a numéricos. Los resultados fueron analizados mediante matrices de confusión por cada tipo de ataque y los resultados con mayor tasa de predicción correcta fueron los relacionados a R2L y U2R.

Las Máquinas de Soporte Vectorial (SVM) [28] son algoritmos de aprendizaje supervisado, que en el contexto de NIDS, han mostrado la posibilidad de clasificar los diferentes tipos de ataques en tiempo real. Praneeth en [29] utilizó SVM para la detección de intrusos en la red con el *dataset* KDD'99.

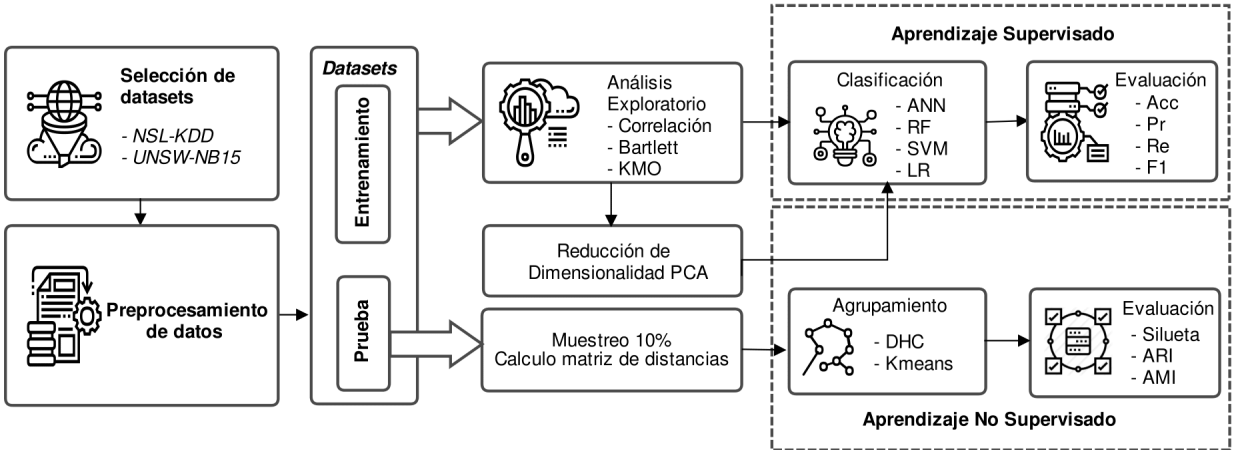

Fig. 1. Esquema de la metodología experimental utilizada en el análisis de las técnicas de aprendizaje supervisado y aprendizjae no supervisado sobre los conjuntos de datos NSL-KDD y UNSW-NB15.

El entrenamiento lo realizó con el 10% de los datos y las pruebas con el 100% de los datos. Para la reducción de dimensionalidad de variables se utilizó Análisis de Componentes Principales (PCA) [30] y para mejorar la tasa de detección se utilizó RBF. Los resultados de detección fueron comparados sin PCA y con PCA, obteniendo tasas de exactitud del 99.4% y 99.7%, respectivamente.

En [31] se utilizó el conjunto de datos KDD'99, en su versión reducida, para la clasificación de cada uno de los ataques mediante SVM, utilizando el *kernel* RBF. El resultado final mostró una exactitud del 89.85%.

Yuan-Cheng y Wang en [32] utilizaron el conjunto de datos KDD'99 con SVM para crear un clasificador binario de datos anormales y normales, sin importar el tipo de ataque del *dataset*. Con el fin de mejorar la precisión, mejorar la velocidad de procesamiento y obtener una fácil clasificación, se utilizó Extracción de Características Kernel PCA (KPCA) [33]. Los resultados fueron presentados, con y sin el uso de KPCA, obteniendo tasas de *accuracy* del 95% y 98.85%, respectivamente.

El documento presentado por Gharaee y Hossein [34], propone un IDS mediante el uso Máquinas de Soporte Vectorial de Mínimos Cuadrados (LSSVM) [35] y Algoritmos Genéticos [36]. Utiliza un método para la selección de características, mediante la función *fitness* de Algoritmos Genéticos, con el fin de obtener variables que reduzcan la dimensión de los datos, incrementen los verdaderos positivos y simultáneamente ayuden a clasificar de manera más eficiente el tráfico por cada tipo de ataque. Para las pruebas se utilizaron los conjuntos de datos KDD'99 y UNSW-NB15 obteniendo predicciones acertadas superiores al 90%.

En [37] se presenta una propuesta basada en SVM y PCA para reconocer e identificar intrusiones dentro de un sistema. Se utilizó el 10% del conjunto de datos KDD'99 para la experimentación. En busca de un óptimo tiempo de respuesta y mejor rendimiento del sistema se redujo la dimensión del *dataset* a 17 componentes. Los resultados fueron categorizados con y sin el uso de PCA, obteniendo *accuracies* del 91% y 90%, respectivamente.

II. Materiales y Métodos

La metodología *Knowledge Discovery in Databases* (KDD) propuesta por [38] es una de las más robustas y se ha convertido en un estándar de facto, para el procesamiento y explotación de los datos. Este artículo usará KDD para evaluar e interpretar patrones y modelos en la toma de decisiones. KDD consta de una secuencia interactiva e iterativa de pasos para la extracción de conocimiento en conjuntos de datos, los cuales son: comprensión del dominio del estudio, selección del *dataset*, minería de datos, interpretación de los resultados y utilización del conocimiento. La Figura 1 resume la metodología empleada para el diseño de los experimentos y las técnicas ejecutadas en este trabajo.

Para esta investigación se han utilizado dos conjuntos de datos. El primer *dataset* seleccionado fue NSL-KDD por sus ventajas respecto a sus predecesores (DARPA 1998 y KDD cup'99) y el segundo UNSW-NB15, debido a que modela escenarios de tráfico tradicionales y modernos [18]. Ambos conjuntos de datos tienen características (variables) en común como: tipo de protocolo, servicio, duración, bytes de origen, bytes de destino y algunas variables que, a pesar de no tener el mismo nombre, representan cualidades similares. Por otro lado, la principal diferencia entre estos dos *datasets* es su cardinalidad. Los autores de los dos conjuntos de datos los han dividido en dos particiones, una de entrenamiento y otra de prueba. En la Tabla I se presentan algunas características de los conjuntos de datos utilizados en este trabajo.

TABLA I
Descripción de los conjuntos de datos NSL-KDD y UNSW-NB15

		NSL-KDD	UNSW-NB15
Número de instancias por partición	**Entrenamiento**	125973	175341
	Prueba	22543	82332
	Total	148516	257673
Porcentaje de instancias por clase (entrenamiento)	**Normales (0)**	53.46	31.94
	Anómalas (1)	46.54	68.06
Porcentaje de instancias por clase (prueba)	**Normales (0)**	43.07	44.94
	Anómalas (1)	56.93	55.06
ID de variables utilizadas	**Independientes**	1, 5-19, 21-42	2, 6-26, 28-38, 40-43, 45
	Dependiente	42	45

Después de seleccionar los *datasets*, se realizó el pre-

procesamiento de los datos con el fin de curar y depurar los conjuntos de datos. Luego, se ejecutó un análisis exploratorio sobre el total de instancias de cada *dataset*, para mostrar las relaciones que existen entre las variables de cada conjunto y tener una visión general, previa al modelado.

Adicionalmente, se utilizó PCA como técnica de reducción de dimensionalidad, con el fin de transformar una gran cantidad de variables interrelacionadas a unas pocas componentes que describan de manera concisa la variabilidad de los datos.

Para la detección de intrusos en la red, en la fase de experimentación, se utilizaron algoritmos de aprendizaje supervisado y no supervisado. En el caso de las técnicas supervisadas, se experimentó con cuatro algoritmos: Redes Neuronales Artificiales, Máquinas de Soporte Vectorial, *Random Forest* y Regresión Logística (LR, por sus siglas en inglés) [39]. Estas técnicas se ejecutaron sobre el total de instancias de la partición de entrenamiento, tomando en cuenta dos escenarios: con PCA y sin PCA. La evaluación de rendimiento de los algoritmos se ejecutó con la totalidad de instancias de la partición de prueba.

En el caso de las técnicas no supervisadas, se empleó un algoritmo de particionamiento (K-Means) [40] y un algoritmo de Clustering Jerárquico Divisivo (DHC) [41]. Ambos algoritmos no necesitan una partición de datos de entrenamiento para el modelado, por lo tanto, se decidió utilizar la partición de datos de prueba para la ejecución de los algoritmos y la evaluación de los resultados. El algoritmo DHC requiere el cálculo de la matriz de distancias euclídeanas entre los elementos (instancias), para cada conjunto de datos, respectivamente. El cálculo de esta matriz es muy costosa computacionalmente ya que depende del número de objetos, por esta razón, se utilizó una muestra aleatoria del 10% del total de datos de la partición de prueba de cada *dataset*, manteniendo la proporción porcentual de prevalencia de clases.

Finalmente, se realizó la evaluación de rendimiento de las técnicas de aprendizaje supervisado y aprendizaje no supervisado. Para evaluar el rendimiento de los algoritmos supervisados se calcularon cuatro métricas: exactitud o *accuracy* (Acc), precisión (Pr), sensibilidad o *recall* (Re) y la medida F1 (F1). Por otro lado, para evaluar los resultados de los algoritmos de aprendizaje no supervisado, se utilizaron tres métricas: el coeficiente de silueta para cada grupo $s(i)$ [42], el Índice Aleatorio Ajustado (ARI) [43] y el índice de Información Mutua Ajustada (AMI) [44].

Los experimentos se ejecutaron en un computador portátil Asus K501UB con sistema operativo Windows 10, procesador Intel(R) Core (TM) i7-6500U, CPU de 4 Núcleos a 2.5GHz, con 8GB de memoria RAM y el lenguaje de programación R v.3.6.1. Los scripts para la verificación y reproducción de los experimentos se encuentran alojados en un repositorio de código, de acceso libre: https://github.com/dievalhu/NIDS.

III. Resultados y Discusión

A. Pre-procesamiento de los Datos

Tanto el conjunto de datos NSL-KDD como UNSW-NB15, tienen variables cuantitativas y cualitativas. NSL-KDD contiene cuatro variables cualitativas y 38 cuantitativas, mientras que, UNSW-NB15 posee cuatro cualitativas y 41 cuantitativas. Para esta investigación se utilizarán solo las variables cuantitativas. La variable dependiente a predecir, por los algoritmos de aprendizaje supervisados, en ambos *datasets* corresponde a la presencia o no de una anomalía en la red y han sido codificadas de forma booleana. Así, un 1 representa una anormalidad, independientemente del tipo de ataque a la red, y un 0 representa datos de tráfico normal. Dado que el atributo a predecir es una variable cuantitativa discreta, el problema a resolver por los algoritmos de aprendizaje supervisados, se traduce en un problema de clasificación. En la partición de prueba del conjunto de datos, donde se evalúa el rendimiento de los algoritmos supervisados, se observa que ambos conjuntos de datos están desbalanceados, i.e., tienen un mayor número de unos sobre ceros. Esto es, para NSL-KDD 12833 unos y 9710 ceros, con un 56.93% de prevalencia de la clase anómala y en el *dataset* UNSW-NB15 existen 45332 unos y 37000 ceros, con un 55.06% de prevalencia de la clase anómala.

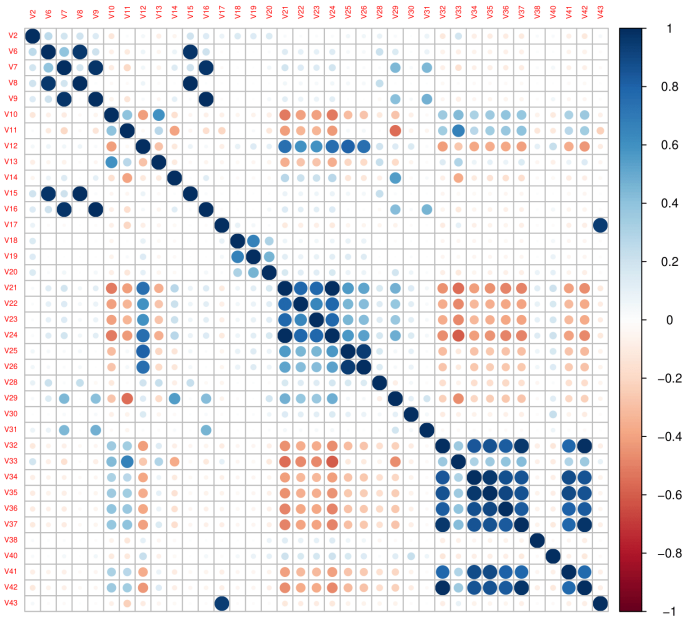

Fig. 2. Matriz de correlación del dataset UNSW-NB15. Los ejes vertical y horizontal muestran las 37 variables cuantitativas independientes, la correlación se mide por el tamaño e intensidad de color de los círculos. El color rojo indica correlación negativa (inversamente proporcional), mientras que el color azul corresponde a una correlación positiva (directamente proporcional).

Ambos *datasets* poseen varias decenas de atributos (variables aleatorias), por lo tanto, es importante identificar, fusionar o eliminar aquellas variables que sean irrelevantes para el estudio, con la finalidad de mejorar el rendimiento computacional y reducir la complejidad de implementación y comprensión del modelo. Mediante un análisis exploratorio de las variables, en ambos *datasets*, se ha identificado que existen variables con una gran cantidad de valores nulos. Bajo estas condiciones, se realizó un análisis de dispersión para cada variable, y se encontró que algunas variables de los conjunto de datos NSL-KDD y UNSW-NB15 presentan varianzas nulas o casi nulas, razón por la cual se decidió no tomarlas en cuenta para la experimentación, con el fin de evitar resultados

de clasificación sesgados. La Tabla I, muestra las variables utilizadas en la experimentación para cada conjunto de datos.

La correlación estadística de variables permite determinar relaciones entre las variables de los conjuntos de datos y su intensidad de relación. Cuando los conjuntos de datos poseen varias decenas de atributos, es menester identificar las variables más explicativas y que contribuyen con mayor información a los modelos que se aplicarán a-posteriori. En la Figura 2 se puede observar la matriz de correlaciones de las variables independientes del *dataset* UNSW-NB15, donde se identifican variables que podrían ser fusionadas al estar fuertemente relacionadas. Se ejecutó el test de esfericidad de Barlett [45], en ambos *datasets*, y se obtuvo un p-valor de $2.2x10^{-16}$. En ambos casos, el p-valor es mucho menor que el nivel de significancia $\alpha = 0.05$, por lo que se rechaza la hipótesis nula H_0 y se determina que es posible aplicar análisis factorial. De manera paralela se ejecutó el test de homogeneidad de la varianza de Kayser Meyer y Olkin (KMO) [46] y se obtuvieron valores de 0.75 y 0.82, para NSL-KDD y UNSW-NB15, respectivamente. Ergo, de acuerdo a [47], un valor de KMO > 0.6 debe considerar fuertemente la posibilidad de utilizar análisis factorial para la reducción de dimensionalidad en los dos *datasets*, ratificándose los resultados del test de esfericidad de Barlett.

El Escalamiento Multidimensional (MDS), permite representar geométricamente un conjunto de objetos mediante puntos, donde las distancias de los puntos (proximidades) corresponden a las diferencias entre los objetos [48], permitiendo así, representar estructuras de datos de alta dimensionalidad en un modelo espacial $\mathbb{R}^2$. Así, MDS permitirá descubrir la estructura de datos y observar de manera tácita la distribución de las clases de la variable dependiente (normal y anormal/anómalo). La Figura 3 muestra la distribución de los objetos, en función de la variable dependiente, para el conjunto de datos UNSW-NB15, donde se observa la distribución de las dos clases.

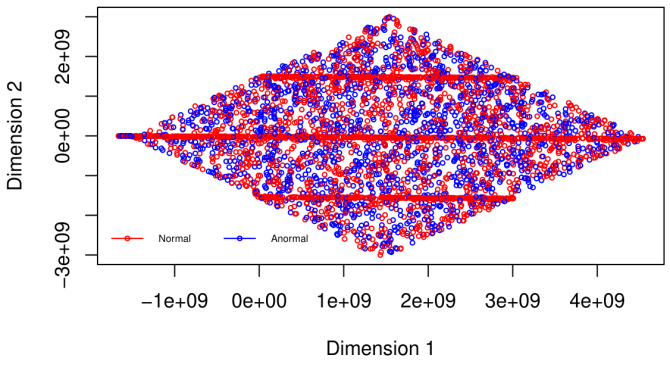

Fig. 3. Escalamiento Multidimensional (MDS) del dataset UNSW-NB15

B. NIDS con Aprendizaje No Supervisado

En la mayoría de los trabajos relacionados con Sistemas de Detección de Intrusos en la Red, no se aborda el problema desde el paradigma de aprendizaje de máquina no supervisado. Una de las técnicas más comunes asociadas al aprendizaje no supervisado son los algoritmos de *clustering* que permiten buscar patrones dentro de los conjuntos de datos. En este trabajo, se exploraron dos tipos de algoritmos de agrupamiento: K-Means y DHC. La experimentación se realizó, en ambos conjuntos de datos, sobre el 10% de las particiones de prueba, i.e., con 2254 instancias en NSL-KDD (971 normales y 1283 anómalas) y 8233 instancias en UNSW-NB15 (3705 normales y 4528 anómalas).

El algoritmo K-Means, tomó como entrada la partición de prueba muestreada con $k = 2$, que se encuentra en consonancia con los dos tipos de clases existentes en el *dataset*: normal y anormal. Por otro lado, el algoritmo DHC, tomó como entrada la matriz de distancias entre los elementos (instancias) que conforman el conjunto de datos de prueba muestreado. Esta matriz permitió observar la similitud que existe entre las unidades de las variables cuantitativas en ambos conjuntos de datos y se configuró el corte del dendograma, del algoritmo DHC, para que genere dos grupos diferentes $k = 2$.

La Tabla II, resume los resultados de evaluación de rendimiento de los dos algoritmos de aprendizaje de máquina no supervisados y muestra los tamaños de los grupos resultantes, en comparación con el tamaño original de los dos grupos, para cada conjunto de datos.

TABLA II
EVALUACIÓN DE RENDIMIENTO DE LOS ALGORITMOS DE APRENDIZAJE NO SUPERVISADO

	NSL-KDD		**UNSW-NB15**	
	K-Means	**DHC**	**K-Means**	**DHC**
Tamaños Grupos Originales	(971,1283)		(3705,4528)	
Tamaños Grupos Clustering	(78,2176)	(1,2253)	(3366,4867)	(2885,5348)
ARI	-0.011019	0.000285	0.093803	0.080482
AMI	0.019543	0.000547	0.063898	0.052890
s(i)	(0.066,0.966)	(0,0.993)	(0.381,0.841)	(0.361,0.722)

Los resultados muestran, para ambos *datasets*, índices de evaluación interna aceptables, ya que los valores de $s(i)$ son cercanos a uno. Una de las desventajas al usar técnicas de *clustering* para agrupar objetos, es la dificultad de evaluar el rendimiento del agrupamiento de los objetos con respecto a los grupos existentes (*ground truth*), i.e., realizar validación externa. En el caso del conjunto de datos NSL-KDD, se observa que los tamaños de los grupos creados por ambos algoritmos de *clustering* están muy desbalanceados, e.g. con DHC el primer grupo tiene una sola instancia y el resto de instancias están en el segundo grupo. Los tamaños de los grupos resultantes distan de los valores reales y por lo tanto los valores de los índices de validación externa (ARI y AMI) son bajos, cercanos a cero. Esto sucede ya que ambos algoritmos de *Machine Learning* no permiten restringir, a-priori, la cantidad de instancias por cada grupo, por lo tanto, los algoritmos buscarán el mejor agrupamiento en función de las características de los datos y sin tomar en cuenta el *ground truth*.

C. NIDS con Aprendizaje Supervisado

Para abordar la tarea de clasificación de anomalías, se usó el número total de instancias de cada conjunto de datos;

la partición de datos de entrenamiento fue usada para la generación de los modelos y la partición de datos de prueba para la evaluación de los resultados. Debido a que ambos *datasets* presentan un gran número de variables, se utilizó PCA como técnica de reducción de dimensionalidad. El número de componentes principales seleccionados para los *datasets* NSL-KDD y UNSW-NB15, considerando un umbral de varianza acumulada superior al 90% fue de 19 y 16, respectivamente. Para mostrar los resultados de clasificación con los cuatro algoritmos: LR, ANN, SVM y RF se han utilizado matrices de confusión, utilizando PCA y sin el uso de PCA (Tabla III).

TABLA III
MATRICES DE CONFUSIÓN PARA LOS ALGORITMOS DE APRENDIZAJE SUPERVISADO

				Clase Actual			
				Sin PCA		Con PCA	
				N	A	N	A
Clase Predicha	NSL-KDD	RL	N	9031	5026	9494	5083
			A	679	7807	216	7750
		ANN	N	9036	5499	4598	4649
			A	674	7334	5112	8184
		SVM	N	9060	4751	8144	5154
			A	650	8082	1566	7679
		RF	N	9454	4857	8672	5909
			A	256	7976	1038	6924
	UNSW-NB15	RL	N	21140	2903	15589	13339
			A	15860	42429	21411	31993
		ANN	N	32074	16188	5795	4530
			A	4926	29144	31205	40802
		SVM	N	22032	253	19593	21075
			A	14968	45079	17407	24257
		RF	N	44306	1026	4391	4164
			A	9761	27239	32609	41168

En primera instancia se utilizó el algoritmo de Regresión Logística para la clasificación binomial por su versatilidad, eficiencia computacional y explicabilidad del modelo. La regresión logística permite estimar la probabilidad de la respuesta binaria Normal (N) o Anormal(A), en función de una o más variables independientes (atributos). El resultado de clasificación tomó en cuenta un umbral superior o igual a 0.5 para la clase anómala y menor a 0.5 para la clase normal.

Por otro lado, la ANN se configuró con una capa de entrada, una capa oculta y una capa de salida. La capa de entrada tiene tantas neuronas como variables independientes posee el *dataset*, la capa oculta permite resolver el problema no lineal y finalmente la capa de salida tiene una sola neurona, ya que el problema de clasificación tiene únicamente dos clases a predecir. En el caso de SVM se utilizó un *kernel* radial, debido a que puede capturar relaciones más complejas entre los puntos (instancias) y a que el límite divisorio de las clases no es lineal. Finalmente, se exploró el rendimiento del algoritmo RF, el modelo se parametrizó con 100 árboles aleatorios y con $\sqrt{p}$ caracterísitcas analizadas en cada nodo, donde p es el número de variables independientes de cada *dataset*.

La Tabla IV resume los resultados de la evaluación de cada clasificador. Se ha añadido una columna adicional, para mostrar los tiempos de entrenamiento (Te), en minutos, de cada algoritmo. El mayor porcentaje de exactitud en la predicción, para los dos *datasets*, fue alcanzado por los

TABLA IV
EVALUACIÓN DE RENDIMIENTO DE LOS ALGORITMOS DE APRENDIZAJE SUPERVISADO

			Acc	Pr	Re	F1	Te [min]
NSL-KDD	RL	Sin PCA	74.69	93.01	64.25	75.99	**0.14**
		Con PCA	76.49	**97.78**	65.13	78.18	1.69
	ANN	Sin PCA	72.62	93.06	62.17	74.54	64.81
		Con PCA	56.70	47.35	49.72	48.51	17.61
	SVM	Sin PCA	76.04	93.31	65.60	77.04	8.85
		Con PCA	70.19	83.87	61.24	70.79	9.31
	RF	Sin PCA	**77.32**	97.36	**66.06**	**78.71**	9.01
		Con PCA	69.18	89.31	59.48	71.40	12.33
UNSW-NB15	RL	Sin PCA	77.21	57.14	87.93	69.26	0.33
		Con PCA	57.79	42.13	53.89	47.29	**0.13**
	ANN	Sin PCA	74.36	**86.69**	66.46	75.24	0.31
		Con PCA	56.60	15.66	56.13	24.49	5.16
	SVM	Sin PCA	81.51	59.55	**98.86**	74.32	46.10
		Con PCA	53.26	52.95	48.18	50.45	39.15
	RF	Sin PCA	**86.90**	73.62	96.37	**83.47**	44.71
		Con PCA	55.34	11.87	51.32	19.28	35.03

modelos RF sin PCA con un porcentaje de 77.32% para el NSL-KDD y 86.90% para el UNSW-NB15. Los resultados de la exactitud muestran que los modelos logran porcentajes más altos cuando se aplican sobre el conjunto de datos UNSW-NB15, sin PCA. En cuanto a la precisión, para el *dataset* NSL-KDD, el algoritmo RL con PCA consiguió el valor más alto con un 97.78%, seguido por el algoritmo RF sin PCA que obtuvo un 97.36%. La peor precisión, para este conjunto de datos, fue del algoritmo ANN con PCA con un valor del 47.35%, este resultado se contrasta con la presición obtenida por el mismo modelo sin PCA, pero aplicado sobre el *dataset* UNSW-NB15, donde se obtuvo la mejor precisión igual a 86.69%. Los resultados muestran que los modelos alcazaron mayores valores de presición, al ser aplicados sobre el conjunto de datos NSL-KDD.

Por otra parte, la sensibilidad alcanzada por los modelos sobre el *dataset* NSL-KDD, no supera el 70%. El valor más alto fue 66.06% y lo obtuvo el algoritmo RF sin PCA. Para el conjunto de datos UNSW-NB15, en cambio, se alcanzaron valores de sensibilidad superiores al 90%. Los algoritmos con mejor *recall* fueron SVM y RF (todos ellos sin PCA), con valores de 98.86% y 96.37%, respectivamente. Para la medida F1, en ambos *datasets*, se destaca el modelo RF sin PCA, con un valor promedio de 81.09%.

Finalmente, al analizar los tiempos de entrenamiento de cada modelo, se observa que el modelo de Regresión Logística tiene el menor tiempo de entrenamiento, para los dos conjuntos de datos con y sin PCA. El mayor tiempo de entrenamiento, igual a 64.81 minutos, fue alcanzado por las Redes Neuronales Artificiales sobre el *dataset* NS-KDD sin PCA. Para el conjunto UNSW-NB15, los modelos que tomaron mayor tiempo de entrenamiento fueron las Máquinas de Soporte Vectorial y *Random Forest*, con valores superiores a los 30 minutos.

IV. CONCLUSIONES

Este trabajo ha analizado distintas tecnicas de *Machine Learning* que permiten predecir anomalías en la red mediante el uso de conjuntos de datos, en el contexto de un NIDS.

En las técnicas de aprendizaje no supervisado, los valores de los índices de validación externa (ARI y AMI) fueron bajos en contraste con el índice de validación interna ($s(i)$), esto se debe fundamentalmente a que los tamaños de los grupos obtenidos por los modelos, no coinciden con el tamaño de los grupos (clases) originales. Sin embargo, los índices de silueta cercanos a uno, muestran que los elementos de cada grupo, creados por ambos algoritmos, están fuertemente cohesionados entre sí. Aunque, este último hecho sea favorable para los resultados del *clustering*, no favorece la identificación de patrones discriminantes que permitan la detección de anomalías en la red.

Como resultado de los experimentos en el paradigma de aprendizaje supervisado, se pudo evidenciar que el algoritmo con mejor desempeño en cuanto a exactitud, sensibilidad y medida F1 fue *Random Forest* sin el uso de Análisis de Componentes Principales. Sin embargo, los tiempos de entrenamiento fueron relativamente altos en comparación con el algoritmo Regresión Logística sin PCA, que obtuvo valores similares de Acc, Re y F1, especialmente en el conjunto de datos NSL-KDD. Los algoritmos ANN y SVM (ambos con PCA), en la mayoría de los experimentos, obtuvieron los porcentajes de rendimiento más bajos con tiempos de entrenamiento más altos. Aunque, a-priori, la reducción de la dimensionalidad a través de PCA podría verse como una ventaja, para mejorar el tiempo de entrenamiento, los resultados muestran que ninguno de los modelos analizados obtuvieron estas mejoras.

En general, los resultados de los experimentos han mostrado que la metodología planteada es extrapolable a otros conjuntos de datos y la configuración de los algoritmos ha permitido alcanzar tiempos de procesamiento computacional aceptables. Las configuraciones de los algoritmos usados en los experimentos fueron básicas, con la intensión de mostrar resultados preliminares de la capacidad aprendizaje y de reconocimiento de patrones de los modelos. El desempeño de los modelos puede incrementarse si se mejora el ajuste de los parámetros, de cada algoritmo, como se evidencia en la revisión de la literatura.

Como trabajo futuro se plantea analizar más técnicas de aprendizaje automatico, en dispositivos con mejores prestaciones computacionales y con técnicas de computación en paralelo, que permitan disminuir el tiempo de entrenamiento y aumentar el número de parámetros de los modelos. Adicionalmente, se pueden incorporar en el análisis técnicas de aprendizaje semi-supervisado, que permitan restringir el tamaño de los grupos y mejoren la discriminación de las clases.

AGRADECIMIENTO

Este trabajo fue financiado por el Grupo de Investigación IDEIAGEOCA de la Universidad Politécnica Salesiana de Quito, Ecuador.

REFERENCIAS

[1] M. M. Yamin, B. Katt, and V. Gkioulos, "Cyber ranges and security testbeds: Scenarios, functions, tools and architecture," *Computers & Security*, vol. 88, p. 101636, 2020.

[2] M. M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the trustworthiness in the industrial iot networks through a reliable cyberattack detection model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6154–6162, 2020.

[3] R. Ortalo, Y. Deswarte, and M. Kaâniche, "Experimenting with quantitative evaluation tools for monitoring operational security," *IEEE Transactions on Software Engineering*, vol. 25, no. 5, pp. 633–650, 1999.

[4] Z. Wang, L. Sun, and H. Zhu, "Defining social engineering in cybersecurity," *IEEE Access*, vol. 8, pp. 85 094–85 115, 2020.

[5] L. Allodi, M. Cremonini, F. Massacci, and W. Shim, "Measuring the accuracy of software vulnerability assessments: experiments with students and professionals," *Empirical Software Engineering*, vol. 25, no. 2, pp. 1063–1094, 2020.

[6] B. Shin and P. B. Lowry, "A review and theoretical explanation of the 'cyberthreat-intelligence (cti) capability'that needs to be fostered in information security practitioners and how this can be accomplished," *Computers & Security*, vol. 92, p. 101761, 2020.

[7] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[8] M. Learning, "The new proving ground for competitive advantage," Technical report, Mit Technology Review, Tech. Rep., 2017.

[9] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[10] A. Handa, A. Sharma, and S. K. Shukla, "Machine learning in cybersecurity: A review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 4, p. e1306, 2019.

[11] R. G. M. Helali, "Data mining based network intrusion detection system: A survey," in *Novel Algorithms and Techniques in Telecommunications and Networking*. Springer, 2010, pp. 501–505.

[12] C. Snijders, U. Matzat, and U.-D. Reips, "Big data: big gaps of knowledge in the field of internet science," *International journal of internet science*, vol. 7, no. 1, pp. 1–5, 2012.

[13] R. Lippmann, R. K. Cunningham, D. J. Fried, I. Graf, K. R. Kendall, S. E. Webster, and M. A. Zissman, "Results of the darpa 1998 offline intrusion detection evaluation." in *Recent advances in intrusion detection*, vol. 99, 1999, pp. 829–835.

[14] K. Siddique, Z. Akhtar, F. A. Khan, and Y. Kim, "Kdd cup 99 data sets: A perspective on the role of data sets in network intrusion detection research," *Computer*, vol. 52, no. 2, pp. 41–51, 2019.

[15] C. I. for Cybersecurity, "Nsl-kdd dataset," 2009, https://www.unb.ca/cic/datasets/nsl.html.

[16] P. Aggarwal and S. K. Sharma, "Analysis of kdd dataset attributes-class wise for intrusion detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015.

[17] N. Moustafa and J. Slay, "Unsw-nb15 dataset for network intrusion detection systems," https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

[18] Moustafa and Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[19] S. Revathi and A. Malathi, "A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[22] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, M. Embrechts *et al.*, "Network-based intrusion detection using neural networks," *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 12, no. 1, pp. 579–584, 2002.

[23] A. Özkaya and B. Karlık, "Protocol type based intrusion detection using rbf neural network," *Int. J. Artif. Intell. Expert Syst*, vol. 3, no. 4, pp. 90–99, 2012.

[24] Q. Que and M. Belkin, "Back to the future: radial basis function network revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[25] B. Ingre and A. Yadav, "Performance analysis of nsl-kdd dataset using ann," in *2015 international conference on signal processing and communication engineering systems*. IEEE, 2015, pp. 92–96.

[26] A. Chandrasekhar and K. Raghuveer, "Confederation of fcm clustering, ann and svm techniques to implement hybrid nids using corrected kdd cup 99 dataset," in *2014 International Conference on Communication and Signal Processing*. IEEE, 2014, pp. 672–676.
[27] S. Kumar and A. Yadav, "Increasing performance of intrusion detection system using neural network," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*. IEEE, 2014, pp. 546–550.
[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[29] P. Nskh, M. N. Varma, and R. R. Naik, "Principle component analysis based intrusion detection system using support vector machine," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2016, pp. 1344–1350.
[30] P. Besse and J. O. Ramsay, "Principal components analysis of sampled functions," *Psychometrika*, vol. 51, no. 2, pp. 285–311, 1986.
[31] M. V. Kotpalliwar and R. Wajgi, "Classification of attacks using support vector machine (svm) on kddcup'99 ids database," in *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE, 2015, pp. 987–990.
[32] Y.-C. Li and Z.-Q. Wang, "An intrusion detection method based on svm and kpca," in *2007 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 4. IEEE, 2007, pp. 1462–1466.
[33] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
[34] H. Gharaee and H. Hosseinvand, "A new feature selection ids based on genetic algorithm and svm," in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 139–144.
[35] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.
[36] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
[37] M. C. Raja and M. M. A. Rabbani, "Combined analysis of support vector machine and principle component analysis for ids," in *2016 International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2016, pp. 1–5.
[38] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.
[39] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1-2, pp. 167–179, 1967.
[40] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.
[41] A. Guénoche, P. Hansen, and B. Jaumard, "Efficient algorithms for divisive hierarchical clustering with the diameter criterion," *Journal of classification*, vol. 8, no. 1, pp. 5–30, 1991.
[42] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
[43] D. Steinley, "Properties of the hubert-arable adjusted rand index." *Psychological methods*, vol. 9, no. 3, p. 386, 2004.
[44] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
[45] M. S. Bartlett, "Properties of sufficiency and statistical tests," *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, vol. 160, no. 901, pp. 268–282, 1937.
[46] H. F. Kaiser, "An index of factorial simplicity," *Psychometrika*, vol. 39, no. 1, pp. 31–36, 1974.
[47] C. D. Dziuban and E. C. Shirkey, "When is a correlation matrix appropriate for factor analysis? some decision rules." *Psychological bulletin*, vol. 81, no. 6, p. 358, 1974.
[48] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

Diego Vallejo-Huanga obtuvo su grado de Ingeniería en Electrónica y Telecomunicaciones por la Escuela Politécnica Nacional (2012) y su posgrado en Gestión de la Información por la Universitat Politècnica de València (2016). Actualmente trabaja como profesor e investigador en la Universidad Politécnica Salesiana (UPS) y como profesor tiempo parcial en la Universidad San Francisco de Quito (USFQ) y la Universidad de las Américas (UDLA). Posee experiencia en los campos de inteligencia artificial, matemática computacional y recuperación de la información.

Marco Ambuludi obtuvo su grado de Ingeniería de Sistemas por la Universidad Politécnica Salesiana (2019). Actualmente trabaja como desarrollador y posee experiencia en varios lenguajes de programación con diferentes frameworks, orientados a la resolución de problemas de aprendizaje de máquina.

Paulina Morillo se graduó de Ingeniería en Electrónica y Telecomunicaciones en la Escuela Politécnica Nacional (2011) en Ecuador y obtuvo su maestría en la Universitat Politècnica de València (2016) en España. Actualmente trabaja como investigadora en el grupo IDEIAGEOCA. Su campo de interés se enmarca en áreas como Machine Learning y métodos numéricos.