

Validación de la Técnica de Inteligencia en la detección de ciberataques

Validation of the Intelligence Technique in the detection of cyber attacks

Santiago Ordoñez Tumbo¹  Katerine Márceles Villalba²  Siler Amador Donado³ 

¹Institución Universitaria Colegio Mayor del Cauca, Faculty of Engineering, Computing Engineering Program, I+D in Computing Group, Popayán-Colombia.

²Universidad de Antioquía, Faculty of Engineering, System Engineering Program, In2lab Group, Medellín-Colombia.

³Universidad del Cauca, Faculty of Electronic Engineering and Telecommunications, System Engineering Program, Information Technology Research and Development Group (GTI), Popayán-Colombia.

Resumen

Este artículo presenta el proceso realizado para la evaluación de la técnica de inteligencia más adecuada que permita la identificación de tráfico malicioso con el fin de minimizar el riesgo a un ciberataque. Esto fue realizado a través de cuatro fases empleando la metodología de investigación-acción articulada a una revisión sistemática de literatura y a través de escenarios propuestos permitieron validar ésta.

Abstract

This article presents the process carried out to evaluate the most suitable intelligence technique for the identification of malicious traffic in order to minimize the risk of a cyberattack. This was accomplished through four phases using an action research methodology articulated to a systematic literature review, and through proposed scenarios that allowed for the validation of this approach.

Keywords: cyberattack detection, Intelligence technique, engineering.

Palabras clave: detección de ciberataque, ciberataque, inteligencia técnica, ingeniería.

¿Cómo citar?

Ordoñez, S., Márceles, K., Amador, S. Validación de la Técnica de Inteligencia en la detección de ciberataques. Ingeniería y Competitividad, 2024, 26(3)e-20213800

<https://doi.org/10.25100/iyv.26i3.13800>

Recibido: 26-05-24

Aceptado: 12-08-24

Correspondence:

Katerine.marceles@udea.edu.co

Esta obra está bajo una licencia Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Conflicto de intereses: ninguno declarado



OPEN  ACCESS

¿Por qué se realizó?:

La investigación surgió por la necesidad de desarrollar un marco de ciberseguridad que integrara inteligencia artificial para responder de manera efectiva a los ciberataques. Este requerimiento surgió de la creciente sofisticación y frecuencia de las amenazas digitales. Para abordarlo, se diseñaron y evaluaron múltiples escenarios, empleando diversas técnicas de inteligencia artificial, con el objetivo de optimizar la detección de amenazas y reducir los errores en la identificación de amenazas.

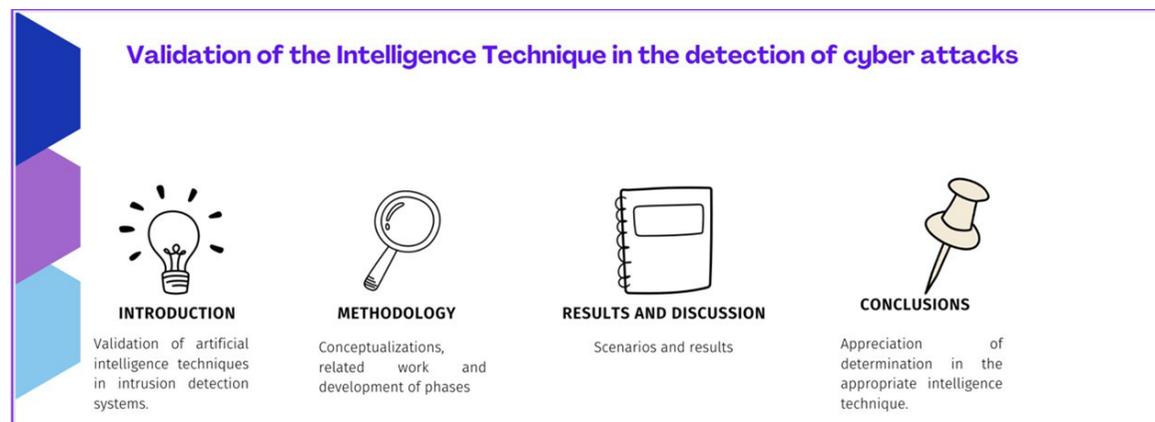
¿Cuáles fueron los resultados más relevantes?

Los resultados más significativos surgieron luego de múltiples pruebas basadas en escenarios de ataque simulados. Se identificó la técnica más efectiva, que no solo mejora la detección de ciberataques sino que también minimiza la incidencia de falsos positivos. Esta técnica demostró ser superior en comparación con los métodos tradicionales y otros nuevos enfoques evaluados durante la investigación.

¿Qué aportan estos resultados?

Los resultados de este estudio son particularmente valiosos porque ofrecen una solución viable y eficiente para empresas medianas y pequeñas, brindando una herramienta crucial para los administradores de red en la detección de anomalías. Este avance representa un apoyo significativo en la mejora de la ciberseguridad, permitiendo una gestión de riesgos más precisa y proactiva.

Graphical Abstract



Introducción

En la actualidad, existen diversas formas de mantener la seguridad en nuestras máquinas o redes. La era digital ha visto un enorme impacto debido a la innovación y las nuevas tecnologías que facilitan a las organizaciones ejecutar sus actividades de manera más eficiente, en gran parte gracias a la adopción del Internet de las cosas (IoT). No obstante, la constante incorporación de estas tecnologías requiere también un enfoque robusto en el aseguramiento de la información. Considerando que los datos almacenados en estos dispositivos son activos críticos, es esencial reconocer que los ciberataques evolucionan diariamente, transformando la seguridad de los sistemas en un desafío constante.

Por otro lado, las organizaciones implementan medidas de seguridad para mitigar los riesgos, pero a menudo sin considerar que los métodos de ataque están en constante evolución, lo que podría representar un problema significativo para los responsables de la seguridad de los sistemas. En este contexto, se utilizan los Sistemas de Detección de Intrusiones Colaborativos (CIDS), que permiten compartir información entre detectores para ampliar la base de conocimiento y mejorar la capacidad de respuesta ante posibles amenazas. Sin embargo, uno de los principales desafíos de los CIDS radica en su capacidad para adaptarse a nuevas amenazas. La actualización de reglas suele requerir intervención manual, lo que puede ser una tarea ardua para las organizaciones que necesitan mantener una base de conocimientos actualizada y procesar grandes volúmenes de datos para prevenir nuevos ataques, aunque esto nunca es 100 % efectivo.

Es aquí donde la inteligencia artificial se vuelve crucial, ya que ofrece la posibilidad de generar detecciones de anomalías de manera inteligente y basada en datos reales, alcanzando un índice de precisión del 95% en la detección de comportamientos anómalos en los dispositivos conectados a la red de una organización.

Para el desarrollo de esta investigación, fue fundamental seguir una serie de pasos estructurados. En este proyecto se empleó la metodología de investigación-acción (I-A), lo que permitió tener resultados alineados a la necesidad presentada en cuanto a la validación de la técnica de inteligencia artificial adecuada a la adaptación a un IDS (Detector de intrusos).

Metodología

Antes de iniciar con el desarrollo metodológico, es importante tener en cuenta algunos conceptos y antecedentes que sirvieron de referencia para el desarrollo de este trabajo:

Ciberseguridad: La ciberseguridad permite salvaguardar equipos de cómputo, dispositivos móviles, los servicios electrónicos y las redes de datos frente a ataques maliciosos [1]. Los dispositivos electrónicos se caracterizan por interrelacionarse entre sí, ya sea de manera directa e indirecta por medio de las redes de datos o por dispositivos de almacenamiento externos.

Sistema Detector de Intrusos (IDS): Un IDS es un dispositivo o aplicación de software que se encarga de monitorear la red en busca de actividad maliciosa [2]. El sistema detector de intrusos está de manera continua analizando el tráfico que pasa por la red de datos, con el fin de identificar anomalías basándose en patrones y heurísticas.

IoT: Se refiere a que los dispositivos están conectados en red para identificar, monitorear y controlar el mundo físico [3]. La IoT tiene como función interconectar y transferir datos.

NetFlow: Es un protocolo de red desarrollado por Cisco System [4] que se encarga de la recolección de información específica del tráfico de red por medio de las IP y que permite solo seleccionar unos paquetes, convirtiéndola en un grupo de datos que contiene una serie de campos de información.

Aprendizaje profundo: En la actualidad, la inteligencia artificial ha tomado un gran auge; está siendo aplicada en muchos campos de la informática. Uno de sus componentes fundamentales

es el aprendizaje profundo (*Deep Learning*). Se puede definir al aprendizaje profundo como una clase de algoritmos de aprendizaje automático [2]. La idea principal del aprendizaje profundo es la resolución de problemas a partir de redes neuronales profundas que buscan imitar la forma en la que el cerebro toma decisiones. Las redes neuronales en este caso poseen un gran número de capas ocultas; en comparación a las redes neuronales tradicionales, esta tecnología busca obtener patrones o características simples a partir de entradas complejas.

Amenaza: Se caracteriza por ser un incidente no deseado, que puede causar daños a un sistema o a una organización [5], aprovechándose de una vulnerabilidad para atacar la seguridad de un sistema de información.

Vulnerabilidad: Es una debilidad o fallo en un sistema de información que puede ser aprovechado por una amenaza [5], que a su vez pone en riesgo la seguridad de la información en cuanto a la integridad, disponibilidad o confidencialidad de esta.

Internet de las cosas industrial (IIoT): Con la aparición del Internet de las cosas, la industria identificó que se podía aprovechar esa tecnología en sus operaciones; para ello aparece el IIoT, el cual supone la integración de la computación, redes y objetos físicos para la industria, donde los dispositivos están conectados en red para detectar, monitorizar y controlar el mundo físico [6].

Aprendizaje supervisado: Se caracterizan por requerir de unas etiquetas iniciales. Estas etiquetas se refieren al valor final de un consecutivo de datos que ya tienen el valor de su objetivo [7]. Esto permite al algoritmo poder aprender de sus errores y aciertos en base a estos resultados ya etiquetados previamente, por lo general utilizados cuando se solicita un resultado tanto numérico como categórico.

Aprendizaje no supervisado: Su aprendizaje se basa en datos no etiquetados. Su experiencia depende casi en totalidad del agrupamiento de datos llamados "clústeres", que permiten en el transcurso del aprendizaje agrupar los suficientes datos para que en las nuevas iteraciones conozca mejor los datos de entrenamiento. Estos métodos de agrupamiento se dividen en dos ramas, los jerárquicos que se basan en la puntuación jerárquica que pone el modelo, y los no jerárquicos que son los que cualquier tipo de flujo puede generar clúster.

Aprendizaje semi supervisado: Para hablar de los algoritmos semi supervisados se tiene que conocer previamente la estructura de los aprendizajes supervisados y no supervisados, ya que este aprendizaje utiliza una parte de ambos [8]. Dado que en la parte de reconocimiento de datos se utilizan los etiquetados propios de los supervisados, y para la parte de la toma de decisión final y el aprendizaje se utiliza un sistema no supervisado basado en los clústeres que generan los no supervisados.

Entre los antecedentes que se tuvieron como referentes están:

Internet of Things: A survey on machine learning-based intrusion detection approaches [9]. Esta investigación se centra en la indagación rigurosa y de vanguardia sobre los temas, el aprendizaje automático aplicado en Internet de las cosas y la detección de intrusiones para la seguridad de las redes informáticas.

El trabajo tiene como objetivo la investigación reciente y en profundidad de trabajos relevantes que abordan diversas técnicas inteligentes y sus arquitecturas de detección de intrusiones aplicadas en redes informáticas, con énfasis en el Internet de las cosas y el aprendizaje automático. Este artículo aporta al trabajo unos resultados rigurosos sobre el aprendizaje profundo y las técnicas más adecuadas.

Detecting Internet of Things attacks using distributed deep learning. [8] En este documento, se propone un marco de aprendizaje profundo distribuido, basado en la nube para el phishing y el ataque con *botnet* para su detección y mitigación. El modelo comprende dos mecanismos de seguridad clave que funcionan de manera cooperativa, a saber: (1) un modelo de red neuronal convolucional distribuida (DCNN) integrado como un complemento de microseguridad de

dispositivo IoT activado para detectar ataques de phishing y DDoS en la capa de aplicación; y (2) una nube temporal a largo-corto plazo con modelo de red de memoria (LSTM) alojado en el *back-end* para detectar ataques de *botnet* e ingerir incrustaciones de CNN (red neuronal convolucional) para detectar ataques de phishing distribuidos en varios dispositivos de IoT. El modelo de CNN distribuido, integrado en un motor de aprendizaje automático, en el dispositivo de IoT del cliente, permite detectar y defender el dispositivo de IoT de los ataques de phishing en el punto de origen. Se crea un conjunto de datos que consta de URL de phishing y no phishing para capacitar a los modelos de seguridad complementarios de CNN y se selecciona el conjunto de datos *N_BaIoT* para entrenar el modelo LSTM de *back-end*. Este artículo aporta mucha información sobre redes neuronales y su implementación. Además, es de resaltar el modelo que implementan con dos redes neuronales, el cual será de mucha utilidad para evidenciar el funcionamiento de éstas, ya que es otra perspectiva que se debe tener en cuenta antes de proceder con la implementación.

Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things [10]. En este documento, se presenta un sistema de detección de intrusiones basado en aprendizaje automático distribuido en Internet de las cosas (IoT) que utiliza tecnología *blockchain*. En particular, se propone la partición espectral para dividir la red de IoT en sistemas autónomos (AS) que permitan la monitorización del tráfico para la detección de intrusiones (ID) por parte de los nodos del área fronteriza de AS seleccionados de forma distribuida. El sistema de identificación se basa en el aprendizaje automático, en el que se entrena un algoritmo de máquina en el vector de soporte utilizando conjuntos de datos de IoT destacados y se proporciona la detección de los atacantes. Además, la integridad de la lista de atacantes se ofrece mediante el uso de la tecnología *blockchain*, que permite una distribución de la información de los atacantes nodos.

El aporte a la presente propuesta es la especificación de vulnerabilidades presentadas y la utilización de *blockchain* en las tecnologías IoT. También se puede tener en cuenta la manera de entrenamiento del algoritmo de aprendizaje automático.

Para la construcción de toda propuesta se deben seguir una serie de pasos. En este sentido, para el desarrollo de este proyecto se hace uso de la metodología investigación-acción (I-A) [11], la cual consiste en unir la teoría con la práctica de tal forma que el investigador pueda sacar conclusiones acertadas sobre las prácticas realizadas. Debido a que este tipo de metodología busca solucionar problemas concretos, logrando entender e interpretar continuamente para mejorar a partir de ellas, a continuación, se describen las fases.

Fase 1: Selección de técnica de inteligencia.

Inicialmente se realizó una revisión de las posibles técnicas de *Deep Learning* que se ajustan a los requisitos que son la detección de ciberataques y/o anomalías en tráfico de red, teniendo en cuenta los artículos que se tomaron como primarios luego de una caracterización, realizando así una selección de la primera revisión de aproximadamente 100 artículos en la fase 1, de tal manera que permite dar una visión de qué características se deben tomar para su posterior selección.

En la tabla 1 se puede observar el listado de artículos que tienen alguna técnica relacionada con inteligencia artificial y que fueron revisados detalladamente para obtener las tecnologías que fueron utilizadas o referenciadas para la construcción de cada uno de ellos.

Tabla 1. Listado de artículos *Machine Learning*.

#	Año publicación	Título del artículo o estudio
1	2020	CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques [12]
2	2018	Machine learning based mobile malware detection using highly imbalanced network traffic [13]
3	2019	EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques [14]
4	2019	Análisis de seguridad en tráfico de redes empleando minería de datos [15]
5	2020	Modelado probabilístico basado en aprendizaje profundo para la detección de anomalías en el tráfico de red [16]
6	2018	Detecting and classifying malicious TLS network traffic using machine learning [17]
7	2019	Indicadores para la detección de ataques ransomware [18]
8	2019	Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection [19]
9	2020	Detecting Malicious DNS over HTTPS Traffic in Domain Name System using Machine Learning Classifiers [20]
10	2020	Machine Learning for Traffic Analysis: A Review [21]
11	2019	Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms [22]
12	2020	Cyber forensics framework for big data analytics in IoT environment using machine learning [23]
13	2019	Android Malware Identification Based on Traffic Analysis [24]
14	2020	Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city [25]

Fuente: elaboración propia de los autores.

Posteriormente, se realizó la lectura de cada uno de los artículos presentes en la tabla 1, donde se obtuvieron las técnicas que se describen a continuación en la siguiente tabla.

Tabla 2. Descripción de técnicas.

Técnicas	Descripción
BiLSTM-BIDIRECCIONAL	Es un algoritmo bidireccional de memoria a corto plazo y largo plazo, el cual presenta una versión extendida de RNs. Se caracteriza por no ser capaz de aprender información contextual durante un periodo de tiempo prolongado, esto ocasionado por el problema de gradiente de fuga. Las LSTM emplean puertas de unidades relacionadas, el cual supera el problema de las gradientes que desaparecen; por lo tanto, preserva la información por periodos más largos y así mantenerlos para el análisis [26].
Swarm(pso)	La optimización de enjambres de partículas (PSO) es un algoritmo EC, se puede emplear para dar solución a problemas de optimización sin conocimientos de dominio [27].
Aprendizaje profundo de mlp	El algoritmo de una percepción multicapa (MLP), suele ser una extensión del algoritmo de regresión logística [28].
Máquinas de boltzmann restringidas (rbm)	Se define como una red neuronal que coloca al Deep Learning una distribución de probabilidad sobre su conjunto de entradas. Las RBM están relacionadas con la variante de las máquinas de Boltzmann quienes son la restricción de sus neuronas se forma un grafo bipartito, con nodos comúnmente llamados visible y oculto respectivamente [29].
Perceptrón multicapa (mlp)	El algoritmo está formado por múltiples capas, lo que permite una buena alternativa para solucionar problemas que no son linealmente separables, siendo esto una limitación a la hora de aplicar el Deep Learning, por ello se le llama perceptrón simple[30].

Fuente: elaboración propia de los autores.

Luego de identificar las diferentes técnicas de aprendizaje, se realizó una comparación teniendo en cuenta los criterios básicos del funcionamiento de la inteligencia profunda o Deep Learning. Para poder determinar la técnica de aprendizaje profundo que mejor se adapte a la identificación de tráfico malicioso o cualquier anomalía, se tuvieron que establecer características para puntuarlas con valores cuantitativos. Por esta razón, se hace uso de los criterios de selección presentes en la tabla 3.

Tabla 3. Criterios de selección para la técnica de aprendizaje profundo

Descripción	Característica	Valor Cualitativo	Valor Cuantitativo
1 Trabaja bajo predicción de datos	Predicción	Si	1
		No	0
		ALTO	1
2 Nivel de precisión frente al entrenamiento final	Precisión	MEDIO	0.5
		BAJO	0
3 Permite identificar si la tecnología es de código libre	Open source	Si	1
		No	0
4 Comportamiento basado en aprendizaje	Aprendizaje	Si	1
		No	0
		ALTO	1
5 La capacidad de producir resultados con situaciones nunca antes vistas	Reconocimiento de tendencias	MEDIO	0.5
		BAJO	0

Fuente: elaboración propia de los autores.

Con base a lo anterior, se creó la tabla 4, donde se especifican las características esenciales a tener en cuenta para así seleccionar la técnica más adecuada.

Tabla 4. Técnicas de aprendizaje profundo con valores cualitativos

Técnica Aprendizaje Profundo	Predicción	Precisión	Open source	Aprendizaje	Reconocimiento de tendencias
Aprendizaje profundo bidireccional de memoria a corto plazo a largo plazo (BiLSTM)	SI	ALTO	SI	SI	ALTO
Neuronal profunda (DNN)	SI	ALTO	SI	SI	ALTO
Swarm(PSO)	SI	ALTO	SI	NO	ALTO
Aprendizaje profundo de MLP	SI	MEDIO	SI	SI	MEDIO
Máquinas de Boltzmann restringidas (RBM)	NO	MEDIO	SI	SI	MEDIO
Perceptrón multicapa (MLP)	SI	BAJO	NO	NO	MEDIO

Fuente: elaboración propia de los autores.

En la tabla 4, se muestra la caracterización cualitativa de las técnicas de aprendizaje profundo [8], el cual dará una posición más centrada frente a los resultados obtenidos según cada característica.

Tabla 5. Técnicas de aprendizaje profundo con valores cuantitativos.

Técnica Aprendizaje Profundo	Predicción	Precisión	Open source	Aprendizaje	Reconocimiento de tendencias	Promedio
Aprendizaje profundo bidireccional de memoria a corto plazo a largo plazo (BiLSTM)	1	1	1	1	1	1
Neuronal profunda (DNN)	1	1	1	1	1	1
Swarm(PSO)	1	1	1	0	1	0.8
Redes neuronales profundas de MLP	1	0.5	1	1	0.5	0.8
Máquinas de Boltzmann restringidas (RBM)	0	0.5	1	1	0.5	0.6
Perceptrón multicapa (MLP)	1	1	0	0	0.5	0.5

Fuente: elaboración propia de los autores.

A partir de los resultados obtenidos en la tabla 5, se puede observar la clasificación y las técnicas mejores valoradas, por lo que se puede deducir cuales son las más adecuadas que se ajustan a las necesidades de esta propuesta.

Al terminar esta categorización de resultados con base a las técnicas de Deep Learning asociadas a este proceso, se concluyó que la técnica de aprendizaje profundo neuronal profunda (DNN) [31] es la más adecuada, teniendo en cuenta la gran precisión en la respuesta y la solución al problema de gradiente, lo que puede presentar un inconveniente a futuro, el cual se soluciona con los accesos para estos nodos. Su buen rendimiento y bajo costo de recursos son tomados como definitivos para darla como la técnica de Deep Learning adecuada.

Luego de toda la investigación realizada, y teniendo en cuenta que la tecnología avanza a pasos agigantados frente a la innovación y la integración de nuevas propuestas, se puso en marcha la implementación del algoritmo de Deep Learning, en el cual se presentaron algunos inconvenientes, como el alto consumo de recursos hardware que es común en este tipo de tecnologías y la problemática de los altos costos en el pre y pos procesamiento de datos, ya que la fuente principal de base de datos contaría en muy poco tiempo con una saturación.

La caracterización es específica hacia la necesidad final del proceso de identificación, por lo que se tienen en cuenta las características, como ser un algoritmo semi supervisado que ayuda a la agilización de los procesos al dividir en 2 fases el análisis. Supervisado para el entrenamiento y la identificación de anomalías y no supervisado para la creación de clústeres para tener un mejor dominio de los datos [32].

Fase 2: Caracterización y construcción del dataset.

Cada contexto en un proyecto tiene un propósito al cual dirigir su algoritmo de inteligencia artificial, teniendo en cuenta unas características y unas necesidades específicas a las cuales dar prioridad; para este caso la prioridad es la precisión en el análisis del flujo de tráfico para encontrar posibles ataques. Luego de esta fase, es importante conocer los datos para el entrenamiento inicial, dado que se deben definir las variables de entrada que se van a utilizar y posteriormente dividir un porcentaje de datos para entrenamiento y otro para pruebas. Por lo tanto, de acuerdo con lo anterior, es necesario contar con un buen dataset de entrada que permita cumplir con todos los requerimientos antes mencionados.

Es importante mencionar que para la implementación del algoritmo es necesario tener un buen dataset que cumpla con el propósito de llegar a un porcentaje mayor al 90%. Esto solo se puede lograr encontrando un buen balance entre el tráfico benigno y tráfico maligno; que por cuestiones de optimización se sugiere que el dataset de entrenamiento contenga un porcentaje de al menos el 20% de ataques y un porcentaje de 80% para el tráfico normal o categorizado como benigno, como se sugiere en [33], donde propone siempre colocar un porcentaje mayor al tráfico benigno en este caso, por lo cual se decidió el porcentaje mencionado. Así las neuronas aprenderán a identificar el tráfico benigno con mayor facilidad, dejando lo nuevo y no identificado como dato anómalo listo para ser comparado por los clústeres no supervisados. Así mismo, de manera supervisada, el algoritmo aprende a identificar los datos benignos y los malignos para poder encontrar los patrones necesarios para la creación de neuronas lo suficientemente óptimas que permitan la toma de decisiones.

El dataset es el conjunto de datos almacenados y tabulados, donde cada columna representa una variable y las filas un grupo de datos que identifican una misma información. Esta parte es muy importante para la buena implementación del algoritmo de aprendizaje, ya que sí se cuenta con un dataset completo y con datos bien definidos, la parte de la normalización de los datos será más sencilla y así se logrará que el algoritmo aprenda de una manera óptima y de esa forma poder obtener los mejores resultados cuando se realicen las pruebas respectivas.

El tipo de tráfico que se seleccionó para generar los datos de entrada tanto para el modelo de aprendizaje como para el modelo de predicción fue netflow, que utiliza los datos basados en UDP o SCTP a un servidor recolector de datos que, en este caso, sería una raspberry pi 3 que estaría recolectando la mayor cantidad de paquetes netflow durante determinado tiempo para que después el IDS suricata pueda obtener la información de dichos paquetes y de esa forma, cuando se tenga dicha información, se pueda enviar hacia el algoritmo propuesto para que este pueda realizar el análisis del tráfico y a su vez generar una predicción que permita identificar si dicho tráfico es anómalo.

El dataset que se utilizó para la fase de entrenamiento es un dataset con una arquitectura de tráfico netflow, que fue organizado con base a un dataset de prueba recuperado del repositorio de Queensland [34] en Australia, donde se encuentran varias versiones del dataset.

Posteriormente, el dataset fue adaptado a las necesidades de la prueba, incluyendo el tráfico malicioso que se encuentra etiquetado y el tráfico benigno igualmente etiquetado. También es importante decir que la información que se necesitó fue de tipo numérica, por lo que se procedió a generar un script dentro del modelo de aprendizaje, el cual logró esa transformación y normalización con el dataset de aprendizaje.

Para la normalización de los datos se deben tener en cuenta algunas reglas, como por ejemplo: que las variables sean de tipo numérico o bien de tipo float, lo cual es esencial a la hora de crear el modelo de aprendizaje, ya que esto evitará la generación de problemas y que el modelo se ejecute con normalidad. El dataset contiene un total de un millón de datos distribuidos en 12 columnas que contienen datos distribuidos entre anómalos y benignos que van a significar los datos de entrenamiento inicial y con los cuales el algoritmo va a aprender a identificar los posibles ataques que van a ser transformados en nuevas variables de aprendizaje y de alimentación para el dataset de entrenamiento. De igual manera, las neuronas implementadas serán capaces de identificar esos nuevos casos, que es lo que se espera de un algoritmo de Deep Learning.

Fase 3: Escenario de evaluación de técnicas de inteligencia para la detección de ciberataques.

En esta actividad se plantean algunos escenarios controlados, donde la idea principal es probar el funcionamiento del modelo y retroalimentar a su vez el algoritmo de Deep Learning.

Este apartado aborda el cómo la red neuronal logra dar los resultados que se esperaban teniendo como base el dataset de entrenamiento, toda la estructura y todos los cambios adoptados en el transcurso de la investigación.

Para determinar la exactitud del algoritmo, se evaluó mediante una matriz de confusión que será útil para la evaluación de modelos, en este caso, que funciona con base a los principios de las redes neuronales, donde solicitan unos datos de entrada, divididos entre entrenamiento y test, utilizando así el test del modelo para determinar las variables a tener en cuenta por la matriz de confusión. En la figura 1 se muestra una representación de la matriz 2*2 que será la que se tendrá en cuenta para el análisis del funcionamiento del algoritmo.

		Actual Values	
		Yes	No
Predicted Values	Yes	True Positive	False Positive
	No	False Negative	True Negative

Figura 1. Matriz de confusión. Fuente : Samhain Labs | samhain. Samhain Labs. <https://www.la-samhna.de/samhain/> (accedido el 15 de mayo de 2022)

Teniendo en cuenta la figura 1, se puede evidenciar cómo se va a calcular la exactitud de las predicciones del algoritmo. También se puede tener en cuenta la figura 2 que contiene la fórmula que permite determinar la exactitud de manera matemática, donde se calcula la suma de las predicciones correctas sobre las predicciones totales.

$$Accuracy = \frac{\# \text{ of correct predictions}}{\text{total \# of predictions}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figura 2. Fórmula de precisión. Fuente : Samhain Labs | samhain. Samhain Labs. <https://www.la-samhna.de/samhain/> (accedido el 15 de mayo de 2022)

Con base en la fórmula descrita en la figura 2, se adapta a los resultados del proceso de predicción, como se muestra en la figura 3, donde se logra ver un total de aciertos favorables, que dan como resultado un 95.44% de efectividad.

Resultados y discusión

Teniendo en cuenta los resultados que muestra la figura 3, es importante precisar el significado de las variables que se tienen en cuenta para la matriz de confusión, que a continuación se expresan:

Y = Son los datos etiquetados, que tendrá la matriz a comparar con los resultados de la predicción del algoritmo.

y = Son los datos de las predicciones o resultados del algoritmo y su modelo de predicción.

Se encontraron un total de 413511 verdaderos positivos, 0 falsos positivos, 0 falsos negativos y 20495 verdaderos negativos.

Así mismo, los resultados que muestra la figura 4 son arrojados por el modelo, quién, como se señaló anteriormente, ejecuta la predicción de una cantidad de datos ya preparados para los test de predicción y dando como resultado una precisión del 95.44% en los aciertos. Se puede deducir que el algoritmo se comportó de manera adecuada a los datos de entrenamiento y que las neuronas entrenadas están listas para ser puestas a prueba con tráfico real.

```
Epoch 5/5
13563/13563 [=====] - 17s 1ms/step - loss: 0.0393 -
binary_accuracy: 0.9549
13563/13563 [=====] - 12s 846us/step - loss: 0.0393 -
binary_accuracy: 0.9544
binary_accuracy: 95.44%
```

Figura 4. Precisión del algoritmo. Fuente: elaboración propia del autor

Esto da una perspectiva del funcionamiento del algoritmo, que dado a su buen entrenamiento inicial y su estructura de algoritmo de red neuronal puede llegar a mejorar aún más ese porcentaje, teniendo en cuenta que en adelante, siempre que llegue nuevo tráfico y más ataques, será aún más robusto su arsenal de detección. Por consiguiente, se puede evidenciar el tiempo de detección que tuvo el algoritmo a partir de tres escenarios de pruebas realizados, donde el algoritmo detectó los 3 ataques durante el tiempo de su ejecución, ver Tabla 6.

Tabla 6. Resultados de detección de ataques por algoritmo de Machine Learning.

Escenario	Ataque	Proceso		Tiempo de ejecución
1	Backdoor	Exitoso	Sí	3 Min
2	DDOS	Exitoso	Sí	8 Min
3		Exitoso	Sí	5 Min

Fuente: elaboración propia del autor

A continuación, se describen los escenarios que se diseñaron para las pruebas de detección del algoritmo:

Escenario 1. En este escenario se realiza un ataque backdoor que genera una gran cantidad de tráfico malicioso y de comportamiento anómalo, donde el algoritmo tuvo el reto de identificar esa conexión maligna.

Backdoor

Cuando se habla del ataque backdoor, se puede decir que es un tipo de ataque informático que es diseñado para dar acceso de manera remota al atacante. Para este escenario se preparó una conexión directa entre el atacante y la víctima, teniendo al atacante sobre una máquina virtual con el sistema operativo Kali Linux y la víctima igualmente con una máquina virtual con un sistema operativo de Windows 7. Es importante mencionar que no se llevó a cabo el escenario con un sistema operativo de Windows en su versión 10 y/o 11, debido a que la gran mayoría de las pequeñas y microempresas cuentan con máquinas de recursos limitados para sus actividades diarias y son pocas las que actualizan el sistema operativo dado al rendimiento de la máquina; por ello se seleccionó la versión del Windows 7.

Escenario 2. En este escenario se realizó un ataque DDOS que, al igual que el backdoor, generó una gran cantidad de tráfico malicioso y de comportamiento anómalo, que representó un reto para el algoritmo, ya que, al denegar servicios, se pudo encontrar la anomalía para esto. Al igual que algunos otros ataques, se entrenó previamente el algoritmo.

DDOS

El ataque DDOS o (Ataque de denegación de servicio) es básicamente una forma de hacer caer servicios en servidores, páginas, etc. con un número exagerado de peticiones a una dirección IP, de tal manera que los servidores son incapaces de procesar las peticiones, generando errores y reinicios, para lo cual solo se necesita una víctima. Para este ataque se tuvieron en cuenta los puertos de una máquina virtual, la cual se preparó previamente para el ataque. Aquí el algoritmo encontró los patrones del ataque por denegación.

Escenario 3. Para este escenario se realizó un ataque de envenenamiento por ARP (Address Resolution Protocol), conocido por ser uno de los protocolos fundamentales de redes IPv4, por lo que al momento de enviar el tráfico se generó un reto para el algoritmo, ya que su rastro fue muy parecido a los protocolos de la IPv4.

Ataque de envenenamiento por ARP. Es el ataque que permite interceptar una conversación o transmisión de datos entre dos máquinas conectadas, las cuales estarán enviando información al atacante.

	positivo	negativo
positivo	3	2
negativo	0	2

Figura 5. Matriz de confusión de tráfico real. Fuente: Elaboración propia de los autores

En la figura 5, se muestra de forma gráfica, mediante una matriz de confusión, el resultado de las evaluaciones, comparando los resultados en ataques ejecutados, donde se evidencia que los tres ataques ejecutados fueron detectados; asimismo se detectaron dos ataques que no eran, generando así dos falsos positivos. La efectividad final del algoritmo demostró ser eficiente frente a escenarios reales y la capacidad de adaptarse a nuevos retos.

Conclusiones

Teniendo en cuenta los resultados obtenidos por la clasificación de técnicas de aprendizaje automático, se puede determinar que siempre que se va a seleccionar una técnica o un algoritmo, se debe tener en cuenta el tipo de datos que se van a manejar en todo momento como entradas, ya que esto determinará el porcentaje de aciertos y de falsos positivos.

El uso de las nuevas tecnologías y la infinidad de nuevos frameworks que aparecen pueden facilitar el uso del Machine Learning, haciendo que la programación y entrenamiento de modelos sean mucho más efectivos y fáciles de aplicar; uno de ellos, por ejemplo, es sklearn, que contiene

la mayoría de las librerías, listas para usar en los proyectos de Machine Learning, entre otras tecnologías.

Finalmente, a manera de recomendación, es importante precisar los recursos tanto de software como hardware que podrían ser necesarios para la ejecución adecuada del algoritmo.

Agradecimientos

Gracias a la Universidad del Cauca, especialmente al grupo de investigación GTI, a la Institución Universitaria Colegio Mayor del Cauca y al grupo in2lab de la Universidad de Antioquía, por el apoyo para el desarrollo de esta propuesta.

Referencias

- [1] Kaspersky, «¿Qué es la Ciberseguridad?» Latam Kaspersky, [En línea]. Available: <https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security>. [Último acceso: 12 01 2021].
- [2] J. P. Sifre, «IDS de red para la detección de ataques sobre SSH y FTP,» Universidad de Alicante, España, 2020.
- [3] B. Y. Navarro, «Blockchain y sus aplicaciones,» 2017. [En línea]. Available: <https://docplayer.es/74398078-Blockchain-y-sus-aplicaciones.html>. [Último acceso: 1 2021].
- [4] T. ©. 2021, «TELCO manager,» Telcomanager, 2021. [En línea]. Available: <https://www.telcomanager.com/es/blog/que-es-el-netflow/>. [Último acceso: 12 1 2021].
- [5] J. S. A. Enrique Javier Santiago, «Riesgos de Ciberseguridad en las Empresas,» Tecnología y Desarrollo, vol. 15, pp. 3-33, 2017.
- [6] P. P. Angie Valencia, «Internet Industrial de las Cosas (IIOT): Nueva Forma de Fabricación Inteligente,» Fundación Universitaria de Popayán - FUP, Popayán, 2020.
- [7] H. G. W. W. Y. G. Yi Zeng, «A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework,» IEEE Xplore, vol. 7, n° Doi:10.1109/ACCESS.2019.2908225, pp. 45182 - 45190, 01 04 2019.
- [8] P. R. K.-K. R. C. N. B. Gonzalo De La Torre Parra, «Detecting Internet of Things attacks using distributed Deep Learning,» Journal of Network and Computer Applications, vol. 163, n° <https://doi.org/10.1016/j.jnca.2020.102662>, 01 08 2020.
- [9] J. P. P. C. d. O. L. R. M. V. H. C. d. A. Kelton Pontara Augusto da Costa, «Internet of Things: A survey on machine learning-based intrusion detection approaches,» Computer Networks, vol. 151, n° <https://doi.org/10.1016/j.comnet.2019.01.023> Get rights and content , pp. 147-157, 2019.
- [10] H. K. Q. C. C. M. L. Muhammad Asaad Cheema, «Utilizing Blockchain for Distributed Machine Learning based Intrusion Detection in Internet of Things,» IEEE Xplore, vol. 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), n° Doi: 10.1109/DCOSS49796.2020.00074, 2020.
- [11] J. E. L. Emilio Berrocal de Luna, «El proceso de investigación educativa II: Investigación - Acción,» Universidad de Granada, España.
- [12] Z. T. A. K. B. X. D. M. G. Muhammad Shafiq, «CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques,» IEEE Xplore.
- [13] Q. Y. H. H. S. W. L. P. L. W. B. Y. Zhenxiang Chen, «Machine learning based mobile malware detection using highly imbalanced network traffic,» Information Sciences, n° <https://doi.org/10.1016/j.ins.2017.04.044>, pp. 346-364, 2018.
- [14] T. J. L. Ayush Kumar, «EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques,» IEEE Xplore, 22 07 2019.
- [15] F. U. R. A. Y. A. T. F. Toshiro Nagata Bolivar, «Análisis de seguridad en tráfico de redes empleando minería de datos,» Revista Ibérica de Sistemas e Tecnologías de Informação, vol. 21, pp. 314-326, 2019.
- [16] C. C. J. G. Santiago Eguren, «Modelado probabilístico basado en aprendizaje profundo para la detección de anomalías en el tráfico de red,» XXI Workshop de Investigadores en Ciencias de la Computación, n° <http://sedici.unlp.edu.ar/handle/10915/77280>, pp. 1-4, 2019.
- [17] G. A. G. Montes, «Detecting and classifying malicious TLS network traffic using machine learning,» E.T.S. de Ingenieros Informáticos (UPM), Madrid, 2018.

- [18] L. I. B. L. Á. L. V. C. M. B. H. Á. Freddy Daniel Bazante Veloz, «Indicadores para la detección de ataques ransomware,» *Revista Ibérica de Sistemas e Tecnologías de Informação*, nº 19, pp. 493-506, 2019.
- [19] J. B. A. D. Omar M. K. Alhawi, «Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection,» *Cyber Threat Intelligence. Advances in information Security*, vol. 70, nº https://doi.org/10.1007/978-3-319-73951-9_5, pp. 1-11, 24 04 2018.
- [20] P. K. R. Sunil Kumar Singh, «Detecting Malicious DNS over HTTPS Traffic Using Machine Learning,» *IEEE Xplore*, nº Doi:10.1109/3ICT51146.2020.9312004, 08 01 2021.
- [21] Q. Y. Nour Alqudah, «Machine Learning for Traffic Analysis: A Review,» *Procedia Computer Science*, vol. 170, pp. 911-916, 2020.
- [22] B. P. J. B. B. N. Robert Ian McKay, «Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms,» *ACM Digital Library*, nº <https://doi.org/10.1145/3314545.3314569>, pp. 31-35, 03 2019.
- [23] V. P. S. M. S. Gural Singh Chhabra, «Cyber forensics framework for big data analytics in IoT environment using machine learning,» *Multimed Tools Appl*, vol. 79, nº <https://doi.org/10.1007/s11042-018-6338-1>, p. 15881–15900, 2020.
- [24] Y. L. W. F. Rong Chen, «Android Malware Identification Based on Traffic Analysis,» *Artificial Intelligence and Security. ICAIS 2019. Lecture Notes in Computer Science()*, vol. 11632, nº https://doi.org/10.1007/978-3-030-24274-9_26, p. 293–303, 2019.
- [25] Z. S. X. D. M. G. Muhammad Shafiq, «Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,» *Future Generation Computer Systems*, vol. 107, nº <https://doi.org/10.1016/j.future.2020.02.017>, pp. 433-442, 06 2020.
- [26] H. Y. D. W. J. D. M. Z. Wanting Yu, «SL-BiLSTM: A Signal-Based Bidirectional LSTM Network for Over-the-Horizon Target Localization,» *Journals Hindawi*, vol. 2021, nº <https://doi.org/10.1155/2021/9992120>, 2021.
- [27] N. N. y. A. B. Miloud Besnassi, «Face detection based on evolutionary Haar filter,» *Pattern Analysis and Applications*, vol. 23, nº <https://doi.org/10.1007/s10044-019-00784-5>, pp. 309-330, 12 02 2020.
- [28] R. D. Gómez, «Introducción y optimización estocástica de redes neuronales profundas MLP,» *Universitat De Barcelona, Barcelona*, 2020.
- [29] E. D. d. I. R. Montero, «Máquinas restringidas de Boltzmann para el modelado de sistemas no lineales,» *Centro de Investigación y de Estudios avanzados del Instituto Politécnico Nacional, Tesis para Doctorado en Ciencias*, 2018.
- [30] M. F. I. M. Gilbert Pla Martinez, «Clasificador automático de imágenes de muestras de sangre basado en redes neuronales profundas,» *Revista Ingeniería Electrónica, Automática y Comunicaciones*, vol. 40, nº 1, pp. 18-30, 2019.
- [31] B. S. S. A. M. J. F. D. B. G. N. Chao Liang, «Intrusion Detection System for Internet of Things based on a Machine Learning approach,» *IEEE Xplore*, nº Doi: 10.1109/ViTECoN.2019.8899448, 2019.
- [32] L. A. A. M. T. S. V. S. M. Cristian Cardellino, «Convolutional Ladder Networks for Legal NERC and the impact of Unsupervised Data in Better Generalizations,» *The Thirty-Second International Florida Artificial Intelligence Research Conference (Flairs - 32)*, pp. 155-160, 2019.
- [33] E. d. d. m. d. a. p. e. Azure, «Microsoft Azure,» 2020. [En línea]. Available: <https://docs.microsoft.com/es-es/azure/architecture/reference-architectures/ai/training-deep-learning>. [Último acceso: 02 02 2021].
- [34] N. f. Cybersecurity, «Cisco Press,» 03 10 2017. [En línea]. Available: <https://www.ciscopress.com/articles/article.asp?p=2812391&seqNum=5>. [Último acceso: 02 02 2021].