Detección de ciberataques mediante redes neuronales

Santiago Ríos Guiral
Facultad de Ingeniería - Universidad de Antioquia
Medellín, Colombia
santiago.riosg@udea.edu.co

***Resumen*—**
***Palabras claves*—Detección de ciberataques**

I. Introducción

La seguridad de las redes de computadoras es una necesidad fundamental que se debe satisfacer en el contexto actual de la evolución de las redes. En los últimos años, se ha producido un aumento en el número de ciberataques, los cuales son más frecuentes, complejos y de mayor volumen. Reportes de seguridad de Microsoft e IBM presentan un aumento de más del 40% en ciberataques entre los años 2021-2022 [1], [2]. Estas alarmantes cifras demuestran la necesidad de desarrollar nuevas estrategias de detección. En aras de mejorar la seguridad de la red se requiere la implementación de nuevas tecnologías como las redes definidas por software (SDN) y las redes autogestionables. El objetivo de este tipo de redes se centra es desplegar redes autónomas capaces de predecir cambios y se que se adapten al comportamiento de la red [3]. La automatización de la red es una característica deseable en el campo de la ciberseguridad de tal forma que las decisiones no dependan de la intervención humana en el proceso de defensa ante la presencia de nuevos ciberataques.

La complejidad de la red y la naturaleza dinámica de sus procesos subyacentes hace de los algoritmos de Machine Learning (ML) una herramienta natural para detectar, diagnosticar y mitigar intrusiones [4]. A través de algoritmos de ML es posible crear sistemas de detección de intrusiones (IDS). La implementación de ML en el desarrollo de aplicaciones permite incrementar la detección de ataques sofisticados, reaccionar más rápido ante una intrusión, mejorar la escalabilidad de la red, y permite automatizar las acciones de seguridad. En la revisión del estado del arte, se observa un incremento en el desarrollo de aplicaciones de seguridad que utilizan ML, incluyendo algoritmos de Deep Learning (DL). Recientemente, ML y DL son implementados para mejorar la seguridad de la red [5]. Consecuentemente, es fundamental la implementación de ML y DL en las redes de computadores con el objetivo de mejorar la eficacia en la detección y mitigación de ataques.

II. Resultados previos

En esta sección se presentan trabajos relacionados con la implementación de redes neuronales en la detección de ataques informáticos. En Li et al. [6], se comparan diferentes modelos de DL con diferentes ventanas de tiempo que agregan los paquetes de red en flujos. Los autores implmeentan CNNs y RNNs donde logran obtener una precisión de 88.6% y 91.5% respectivamente. En [7], los autores proponen DeepIDS. El sistema utiliza una red neuronal recurrente (GRU-RNN) donde se logra obtener una precisión de 90% en la clasificación correcta del tráfico. Esta red neuronal se despliega en un sistema de detección de intrusiones para detectar tráfico anómalo. En [8], los autores utilizan redes neuronales recurrentes (RNN) para detectar ataques DDoS. El estudio presenta una red de memoria a corto plazo (LSTM) para obtener información de características temporales y espaciales del tráfico de red. Los autores obtienen una precisión de 99.9% en la clasificación de tráfico. En [9], Haider et al. proponen una red neuronal híbrida como la construcción de 2 redes neuronales, ya sean redes neuronales convolucionales (CNN) o redes neuronales recurrentes (RNN). El sistema puede utilizar 2 CNN o 2 RNN donde las salidas se combinan como entradas de una tercera red neuronal, la cual define una salida binaria. De esta forma se clasifica el tráfico como benigno o como anómalo de tipo de Denegación distribuida de servicios (DDoS). Los autores afirman lograr una precisión de 99.45% para un montaje con CNN y del 98.68% para un montaje con RNN. En [10], Wang et al. proponen un sistema de defensa el cual implementa una CNN. Para evaluar el rendimiento, los autores modifican la arquitectura cambiando el número de capas convolucionales y/o cambiando el número de capas totalmente conectadas. Se logra obtener una precisión entre el 98.7% y 99%. En [11], los autores presentan un algoritmo basado en RNN. El modelo selecciona las mejoras características de los paquetes de red de acuerdo al modelo OCSA. Posteriormente la RNN utiliza el algoritmo Propagación hacia atrás con retraso de tiempo (BPTT) para obtener una clasificación binaria. El modelo logra una precisión del 98.1% para clasificar entre paquetes normales y ataques. Finalmente, en [12] presentan una red neuronal clásica (NN)implementada en nuevas tecnologías de red como lo son los conmutadores programables. En este sistema se logró una precisión del 96.8%.

En consecuencia, a través de estos artículos, y otros más que se encuentran en la literatura, se materializa la importancia de aplicar modelos de ML y DL en el área de la ciberseguridad. En el proyecto final del curso de Deep Learning se espera implementar un modelo capaz de detectar la ocurrencia de ciberataques de tal forma que se pueda brindar la seguridad necesaria a una red de computadores.

III. Objetivo de Machine Learning

Con el propósito de mejorar el rendimiento de los sistemas de detección de intrusiones (IDS), se ha recurrido a la implementación de algoritmos de Machine Learning (ML) y Deep Learning (DL). Estos sistemas tienen la función de clasificar el tráfico de red, estableciendo clases para el tráfico anómalo y benigno. La inclusión de algoritmos de ML o DP busca inferir sobre el tráfico que llega a una red, permitiendo así la predicción de su clase. En caso de tratarse de tráfico normal, el sistema permite su libre paso a través de la red hasta su destino, mientras que, en presencia de tráfico malicioso, se busca detectar el ataque y bloquear su acceso a la posible víctima. En este sentido, para el proyecto del curso se pretende desarrollar un clasificador haciendo uso de una base de datos con tráfico de red referente en el área de ciberseguridad. Además, se espera crear un modelo de ML que permita la clasificación del tráfico mediante diferentes características extraídas (características estadísticas, propias del tráfico, entre otras). La base de datos se utilizará para el entrenamiento y evaluación del modelo. En resumen, el modelo de ML o DP puede ser implementado como parte fundamental de un sistema de ciberseguridad robusto capaz de detectar diferentes tipos de ataques.

IV. Base de datos CICIDS2017

En este proyecto se va a utilizar la base de datos *Intrusion Detection Evaluation Dataset* (CIC-IDS2017) [13]. El desarrollo de una base de datos confiable de tráfico de red es fundamental para la evaluación y mejora de los sistemas de ciberseguridad. En particular, los sistemas de detección de intrusiones (IDS) son herramientas críticas para contrarrestar los ataques cada vez más sofisticados y frecuentes. En este contexto, la base de datos CIC-IDS2017 se presenta como una solución valiosa para la comunidad de la ciberseguridad, ya que proporciona una base de datos confiable para la evaluación de IDS y otros sistemas de defensa.

CIC-IDS2017 contiene tráfico benigno y anómalo, los cuales se asemejan a paquetes reales encontrados en la redes de computadores. Esta base de datos organiza el tráfico en archivos PCAP. Asimismo, esta base de datos también se encuentra en formato CSV, donde se utilizó el extractor CICFlowMeter [14] para generar estos archivos. El extractor captura los paquetes del archivo PCAP y genera flujos de paquetes los cuales se encuentran debidamente etiquetados en un archivo CSV.

El periodo de captura de datos comenzó el lunes 3 de julio de 2017 y finalizó el viernes 7 de julio de 2017, lo que supone un total de cinco días. Incluye tráfico normal y los ataques implementados incluyen FTP de fuerza bruta, SSH de fuerza bruta, DoS, Heartbleed, ataque web, infiltración, botnet y DDoS.

En el proyecto de detección de ataques se va a utilizar la versión de CIC-IDS2017 organizada por el extractor CICFlowMeter [14]. Cada flujo de tráfico está etiquetado, ya sea como normal o ataque e incluye 84 características por cada flujo generado. La base de datos está organizada en ocho archivos. En la tabla I se presenta la estructura de la base de datos.

TABLE I
Estructura de la base de datos

Nombre de los flujos recolectados	**Tamaño del archivo (MB)**	**Flujos**
Monday-WorkingHours.pcap	268.6	529918
Tuesday-WorkingHours.pcap	174.7	445909
Wednesday-workingHours.pcap	258.6	692703
Thursday-WorkingHours-Afternoon-WebAttacks.pcap	92	170366
Thursday-WorkingHours-Afternoon-Infilteration.pcap	108.7	288602
Friday-WorkingHours-Morning.pcap	75.4	191033
Friday-WorkingHours-Afternoon-PortScan.pcap	101.9	286467
Friday-WorkingHours-Afternoon-DDos.pcap	96.1	225745

V. Métricas de desempeño

Para los modelos de clasificación implementados con ML y DL es posible utilizar la matriz de confusión presentada en la tabla II. Esta ofrece una gama de medidas de desempeño útiles en la evaluación del modelo.

TABLE II
Matriz de Confusión

Matriz de Confusión		Valores predichos	
		Benigno	Ataque
Clase	Benigno	TP	FN
	Ataque	FP	TN

TP: Positivos verdaderos.
FP: Falsos verdaderos.
FN: Falsos negativos.
TN: Positivos negativos.

Para evaluar la clasificación se utiliza la ecuación 1 que describe la precisión (*Acc*) en la discriminación del tráfico de red. Esta ecuación indica el porcentaje de registros que fueron clasificados correctamente con respecto a todas las muestras de entrada.

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

También existen métricas que permiten evaluar la clasificación con respecto a las clases de salida. A continuación se presentan las medidas de desempeño que permiten tener un mejor conocimiento del tráfico malicioso que se esta detectando.

La ecuación 2 indica la proporción de aciertos de la clase positiva sobre el numero total de predicciones. Por lo tanto indica el porcentaje de ataques predichos correctamente.

$$TPR = \frac{TP}{TP + FN} \tag{2}$$

La ecuación 3 indica la proporción de clasificación de la clase negativa que fueron predichas como positivos. Esto nos

da información de ataques que fueron clasificados como tráfico benigno.

$$FNR = \frac{FN}{FN + TP} \tag{3}$$

La ecuación 2 indica la proporción de clasificación de la clase positiva que fueron predichas como positivos. Esta medida indica el porcentaje de tráfico benigno que fue clasificado como malicioso.

$$FPR = \frac{FP}{TN + FP} \tag{4}$$

En el contexto de ciberseguridad se proyecta obtener un rendimiento alto del sistema de defensa (indicador de gestión de la defensa). Se espera una detección rápida de los ciberataques, así como disminuir el nivel de afectación de la red ante la ocurrencia de estos (actuación del proveedor de la defensa). También se espera dar un grado de confianza a los usuarios de las redes de computadores donde se están desarrollando las medidas de protección de esa información privada (satisfacción del usuario o cliente).

VI. Análisis de los datos

Antes de proceder con la implementación de los diversos modelos de clasificación, resulta apropiado llevar a cabo un análisis exploratorio de los datos con el propósito de obtener una comprensión detallada de las variables contenidas en la base de datos, así como identificar la variable objetivo. La variable objetivo define la naturaleza del tráfico de la red, distinguiendo entre tráfico benigno y los diversos tipos de ataques detectados.

En la figura 1 se presentan las cinco categorías con mayor cantidad de observaciones dentro de la base de datos. Se puede observar como los flujos benignos tienen una mayor

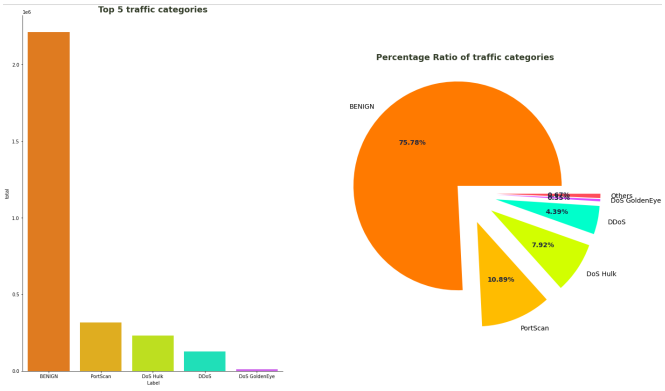

Fig. 1. Categorías de tráfico con más observaciones.

Asimismo, se presentan las categorías disponible en la variable objetivo:

- BENIGN
- PortScan
- DoS Hulk
- DDoS
- DoS GoldenEye
- FTP-Patator
- SSH-Patator
- DoS slowloris
- DoS Slowhttptest
- Web Attack Brute Force
- Web Attack XSS
- Infiltration
- Web Attack Sql Injection
- Heartbleed

Con el fin de abordar la amplia variedad de ataques informáticos y siguiendo el objetivo establecido para la construcción de los modelos de clasificación, se pretende agrupar los ataques informáticos mencionados en una única categoría que indique la presencia de un ataque en la red de computadores. Con este propósito, en la Figura 2 se muestra el resultado obtenido tras realizar dicha agrupación. Se puede observar un desbalance, donde el tráfico benigno corresponde al 75.28% de los datos, y los ataques equivalen al 24.42%.

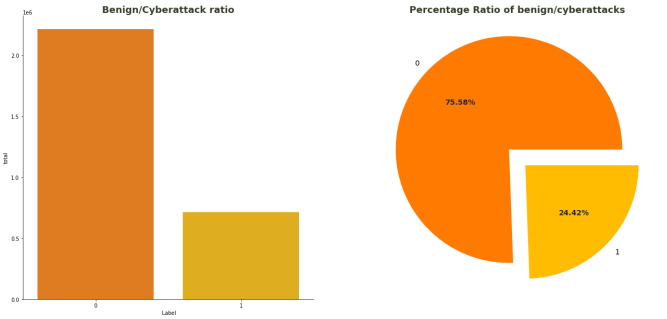

Fig. 2. Categorías de tráfico indicando la cantidad de flujos benignos y anómalos.

Finalmente, tras examinar detalladamente la variable de respuesta, se llevó a cabo un estudio de las características presentes en la base de datos. Esta base de datos abarca información relacionada con los nodos lógicos y físicos de la red de computadores, así como variables de naturaleza tanto categórica como numérica.

VII. Implementación

En esta sección se presenta el desarrollo de los modelos de clasificación de flujos de computadores. Inicialmente se presenta la etapa de procesamiento de datos, se construye un modelo inicial tomando como base un perceptrón multicapas, y por último se presenta la implementación de un modelo que despliega temática de las redes neuronales recurrentes.

A. Procesamiento de los datos

Antes de implementar los modelos de clasificación, es primordial realizar un procesamiento de los datos con el objetivo de garantizar la calidad y adecuación de los datos de entrada, lo que a su vez puede ayudar a mejorar la precisión y eficacia del modelo resultante. A continuación se presentan los pasos seguidos:

1) Definir dos clases para la variable respuesta: benigno (0) y ciberataque (1). Para esto, se codificaron las etiquetas correspondientes al tipo de tráfico, donde los diferentes ataques fueron agrupados en una misma categoría.
2) Se realizo una limpieza de los datos al eliminar observaciones donde las variables de protocolo, puerto fuente y puerto destino tenían valores iguales a cero. Debido a que es un tipo de tráfico de red especial que no es usual en las comunicaciones cotidianas.
3) Se descartaron observaciones (filas) de la base de datos con la presencia de valores de tipo NaN. Ante la presencia de valores infinitos, se procedió a reemplazar con NaN y posteriormente se descartaron dichas observaciones.
4) Se descartaron variables que deben ser privatizadas como las direcciones IP. Además se descartaron variables como la identificación del flujo, el protocolo y la estampa de tiempo.
5) Se realizo el balanceo de la base de datos para tener igual cantidad de observaciones normales y de ataques.
6) Mediante un análisis de correlación se descartaron ocho variables que se mantienen constantes en todas las observaciones. Además, se descartaron tres variables que no aportan a la variabilidad por tener un valor de correlación perfecto con otras variables.
7) Se realizó la normalización de las variables mediante la función de *StandardScaler()*.
8) Se realizó la partición de la base de datos, donde el 70% se utilizo para entrenamiento y el 30% para evaluación.

Posterior al procesamiento de los datos, se da inicio a la ejecución de los diferentes modelos de clasificación que se presentan a continuación.

B. Perceptrón multi capas

Un primer enfoque consistió en emplear un modelo con el perceptrón multicapas. Mediante su creación se buscó definir una base de comparación para evaluar el proceso de detección de ataques.

La arquitectura del modelo consiste en una red neuronal secuencial con tres capas: entrada, capa oculta, y salida. La tabla III presenta la arquitectura de este modelo.

TABLE III
PERCEPTRÓN MULTI CAPA

Capa	Cantidad de Neuronas	Función de Activación
Densa	5	Relu
Densa	2	Relu
Densa	1	Sigmoid

Para este modelo se utilizo el optimizador Stocastic Gradient Descendent (SGD) y se realizó el entrenamiento con 20 épocas con 10 observaciones por cada lote. El objetivo de implementar este modelo era evaluar la detección de ataques con un enfoque compacto.

C. Implementación de redes neuronales recurrentes

El tráfico de red se refiere a la transferencia de datos que ocurre en un instante específico a través de una red de computadoras. Este tráfico se encapsula predominantemente en entidades conocidas como paquetes de red, los cuales contienen tanto la información necesaria para el transporte de datos como los datos mismos que se están transmitiendo. Asimismo, es posible generar flujos de tráfico, los cuales agregan paquetes de red que comparten características en común dentro de una ventana de tiempo. Los flujos se generan a partir de una tupla de 5 características, las cuales son: Protocolo, puerto destino, puerto fuente, dirección IP destino, y dirección IP fuente. Durante la transmisión de datos en la red, se puede evidenciar que los paquetes o flujos de datos comparten algunas características con información previamente transmitida. De esta forma es posible clasificar el tipo de tráfico que se está transmitiendo, así como clasificar la información ya sea en tráfico benigno o anómalo. Debido a este comportamiento del tráfico de red, se define implementar redes neuronales recurrentes y asociadas con el objetivo de clasificar el tráfico al extraer información que caracteriza el comportamiento de los flujos y las variables extraídas.

D. Recurrent Neural Network (RNN)

En la tabla IV se presenta la arquitectura de la red neuronal recurrente donde se implemento una capa recurrente para guardar la información de las variables de los flujos de red. Además, se utilizo un optimizador Adam y para las capas se utiliza la función de activación tangente hiperbólica (tanh). El modelo es entrenado por 30 épocas con 1000 observaciones cada una.

TABLE IV
RECURRENT NEURAL NETWORK

Capa	Cantidad de Neuronas	Función de Activación
SimpleRNN	68	tanh
Dropout (0.2)	-	-
Densa	10	tanh
Densa	1	Sigmoid

E. Gare Recurrent Unit (GRU)

Continuando con la exploración de redes neuronales recurrentes, también se implementó una red neuronal con un capa de entrada de tipo GRU. Similarmente, se utilizó un optimizador Adam y se utilizaron funciones de activación tanh. Además, el modelo fue entrenado con 50 épocas, cada una con 1000 observaciones. La tabla V presenta el resumen del modelo.

TABLE V
GATE RECURRENT UNIT

Capa	Cantidad de Neuronas	Función de Activación
GRU	68	tanh
Dropout (0.2)	-	-
Densa	10	tanh
Densa	1	Sigmoid

F. Long Short Term Memory (LSTM)

En la tabla VI se presenta el modelo que utiliza una capa de tipo LSTM con el objetivo de memorizar la información de los diferentes flujos de tráfico de red. En este modelo se utiliza un optimizador Adam, se entrena por 50 épocas con 1000 observaciones, y las funciones de activación son tanh.

TABLE VI
LONG SHORT TERM MEMORY

Capa	Cantidad de Neuronas	Función de Activación
GRU	68	tanh
Dropout (0.2)	-	-
Densa	10	tanh
Densa	1	Sigmoid

VIII. RESULTADOS

A. Comparación de modelos

La tabla VII presenta una recopilación de los modelos implementados. Se realiza una comparación del desempeño de acuerdo a las métricas establecidas en la sección V. El objetivo de la clasificación se centra en detectar la mayor cantidad de ataques, para esto se busca que las métricas de tasa de falsos negativos y la tasa de falsos positivos tengan el menor valor. De igual forma es importante tomar el valor de precisión como métrica fundamental para evaluar la clasificación en general.

TABLE VII
COMPARACIÓN DE LOS MODELOS PROPUESTOS

Modelos propuestos	ACC	TPR	FNR	FPR	Loss	Val Loss
Perceptrón V1	0.974	0.979	0.021	0.031	0.091	0.087
Perceptrón V2	0.965	0.975	0.025	0.045	0.117	0.118
RNN	0.983	0.993	0.006	0.027	0.047	0.044
LSTM	0.985	0.992	0.008	0.022	0.042	0.040
GRU	0.988	0.997	0.003	0.021	0.036	0.034

Según la tabla comparativa (Tabla VII), se procede a realizar una comparación de los resultados obtenidos mediante la implementación de cinco modelos distintos. Se puede apreciar que, en términos de precisión en la clasificación, el modelo basado en capas GRU presenta el desempeño más destacado, alcanzando un valor del 98.8%. Asimismo, este mismo modelo exhibe los resultados más favorables en cuanto a la clasificación correcta de los ataques, logrando un porcentaje del 99.7%. En resumen, es evidente que este modelo exhibe un desempeño sobresaliente en todas las métricas de desempeño previamente establecidas. Por otra parte, el modelo de perceptrón multicapa, al emplear la técnica de detección temprana del entrenamiento, se destaca por presentar una menor precisión, ya que no logra caracterizar adecuadamente el tráfico normal, resultando en clasificaciones erróneas como ataques. En este caso, se obtuvo una precisión general de 96.5%.

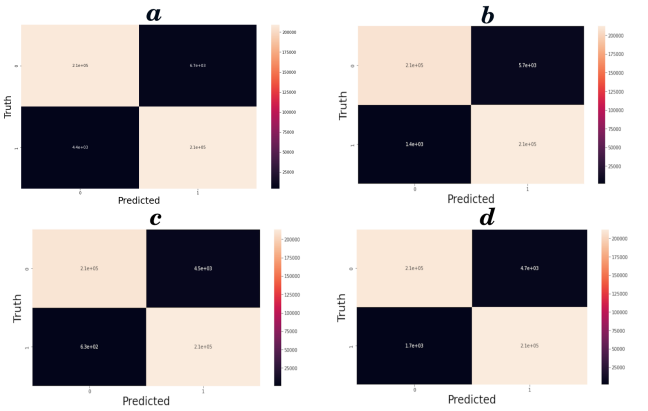

Fig. 3. Comparación de las matrices de confusión de los modelos: (a) Perceptrón, (b) RNN (c) GRU, (d) LSTM.

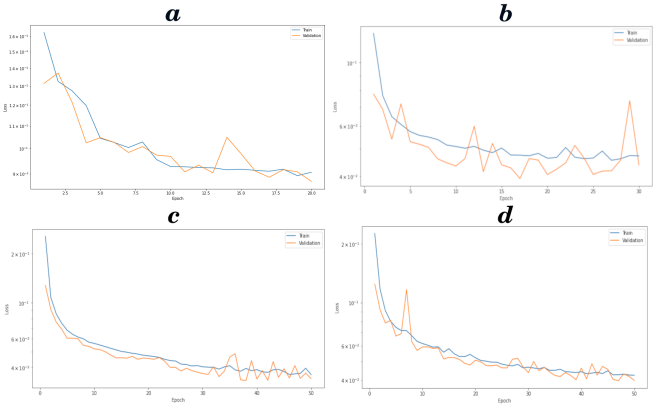

Fig. 4. Comparación de las funciones de perdida de los modelos: (a) Perceptrón, (b) RNN (c) GRU, (d) LSTM.

B. Funciones de perdida

En la Figura 4 se presenta el contraste de las funciones de pérdida obtenidas durante el proceso de entrenamiento de los modelos. En líneas generales, se puede inferir que todavía existe un margen de mejora para dichos modelos. Se plantea la posibilidad de extender el entrenamiento mediante un mayor número de épocas, ya que hasta el momento de la captura de las funciones no se ha manifestado un fenómeno de sobreajuste. Además, se aprecia cómo la función de pérdida de entrenamiento de los modelos que incorporan capas GRU y LSTM converge hacia un punto de estabilidad, en contraste con la función de pérdida de validación. Esto sugiere que los modelos propuestos se ajustan de manera adecuada a la clasificación del tráfico de red. En particular GRU, LSTM y RNN son modelos adecuados para detectar tráfico anómalo

C. Matrices de confusión

Para concluir la exposición de los resultados, se incluye en la Figura 3 una comparativa de las matrices de confusión correspondientes a los modelos evaluados. Se destaca que el modelo que emplea capas GRU logra la correcta clasificación de un mayor número de instancias tanto en tráfico benigno como en

tráfico anómalo. Además, el modelo con capa RNN exhibe un desempeño satisfactorio en la clasificación de ataques. Por último, se observa que el modelo basado en el Perceptrón multicapa presenta los resultados más desfavorables en cuanto a la clasificación de ataques, lo cual desaconseja continuar con el refinamiento de dicho modelo.

IX. Conclusiones

En las redes de computadores satisfacer la necesidad de seguridad es una tarea imperativa a llevar a cabo. En los últimos años ha incrementado la cantidad de ciberataques ejecutados, los cuales son más complejos y de mayor volumen. Esta alarmante realidad demuestra la necesidad de desarrollar sistemas de detección de ataques capaces de combatir las vulnerabilidades explotadas por los criminales. Además, es fundamental la automatización de las decisiones de seguridad para aumentar la eficiencia en las tareas de defensa. En este sentido la implementación de técnicas de Machine Learning (ML) y Deep Learning (DL) son herramientas naturales para la automatización del proceso de detección y mitigación de ciberataques.

En este proyecto se presenta la comparación de 5 modelos de DL con el objetivo de clasificar el tráfico presente en la base de datos CIC-IDS 2017 en aras de utilizar los modelos en sistemas de detección de intrusiones. Inicialmente se presentaron dos modelos con Perceptron multicapa donde uno de estos utiliza una parada temprana cuando se valida la función de pérdida. Con estos modelos se obtiene una precisión de 97.4% y 96.5%. En general son buenos porcentajes de precisión, sin embargo, presentan resultados pobres en la detección de ataques, por lo que no son muy adecuados para desplegar en sistemas de defensa. Por lo tanto, el siguiente paso consistió en la implementación de modelos más robustos. Para esto se utilizaron herramientas brindadas por las Redes Neuronales Recurrentes (RNN). Con estos modelos se aprovecha de la capacidad inherente de las RNN para modelar tanto dependencias a corto plazo como a largo plazo en la secuencia de flujos de red. De esta manera, se logró capturar patrones y correlaciones temporales presentes en la secuencia del tráfico de red. Además, las RNN permitieron capturar información contextual de cada observación, lo que posibilitó la incorporación de un estado oculto que almacenará información relevante de muestras previas para facilitar la toma de decisiones en instancias futuras. Para este tipo de arquitecturas (RNN) se implementaron tres modelos que utilizan las capas RNN, GRU, LSTM. Con los tres tipos de capas se logro analízar de manera más eficiente las dependencias temporales. Con la capa RNN se logro una precisión de 98.3%, sin embargo presenta el peor resultado el proceso de detección de anomalías. Asimismo, los modelos LSTM y GRU permitieron mejorar el manejo de las dependencias a largo plazo y en la reducción del problema del desvanecimiento de gradientes. Por lo tanto, con estos dos últimos modelos se obtuvieron los mejores resultados. Para el modelo implementado con LSTM se obtuvo una precisión del 98.5% y una precisión en la detección de ataques del 99.2%. Por su parte, con GRU se obtuvo una precisión del 98.8% y una clasificación correcta de los ataques del 99.7%. Por lo tanto ambos modelos son ideales para clasificar el tráfico de la red en aras de detectar ciberataques. También existe un margen de mejora para incrementar la eficiencia de ambos modelos sin llegar al punto de sobreajuste. Sobre todo en el proceso de caracterización del tráfico benigno. Finalmente, los modelos basados en Redes Neuronales Recurrentes si es posible obtener información temporal del comportamiento de los ataques y son arquitecturas ideales para desplegar en sistemas de detección de ataques.

References

[1] IBM, "Cost of a data breach 2022: A million-dollar race to detect and respond," https://www.ibm.com/reports/data-breach.

[2] Microsoft, "Microsoft digital defense report 2022," https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report-2022.

[3] A. S. Jacobs, R. J. Pfitscher, R. A. Ferreira, and L. Z. Granville, "Refining network intents for self-driving networks," in *Proceedings of the Afternoon Workshop on Self-Driving Networks*, 2018, pp. 15–21.

[4] N. Feamster and J. Rexford, "Why (and how) networks should run themselves," *arXiv preprint arXiv:1710.11583*, 2017.

[5] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on sdn based network intrusion detection system using machine learning approaches," *Peer-to-Peer Networking and Applications*, vol. 12, pp. 493–501, 2019.

[6] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of ddos attack–based on deep learning in openflow-based sdn," *International Journal of Communication Systems*, vol. 31, no. 5, p. e3497, 2018.

[7] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, M. Ghogho, and F. El Moussa, "Deepids: Deep learning approach for intrusion detection in software defined networking," *Electronics*, vol. 9, no. 9, p. 1533, 2020.

[8] X. Liang and T. Znati, "A long short-term memory enabled framework for ddos detection," in *2019 IEEE global communications conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[9] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K.-K. R. Choo, and J. Iqbal, "A deep cnn ensemble framework for efficient ddos attack detection in software defined networks," *Ieee Access*, vol. 8, pp. 53 972–53 983, 2020.

[10] L. Wang and Y. Liu, "A ddos attack detection method based on information entropy and deep learning in sdn," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1. IEEE, 2020, pp. 1084–1088.

[11] R. SaiSindhuTheja and G. K. Shyam, "An efficient metaheuristic algorithm based feature selection and recurrent neural network for dos attack detection in cloud computing environment," *Applied Soft Computing*, vol. 100, p. 106997, 2021.

[12] F. Paolucci, L. De Marinis, P. Castoldi, and F. Cugini, "Demonstration of p4 neural network switch," in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*. IEEE, 2021, pp. 1–3.

[13] "Canadian institute for cybersecurity. intrusion detection evaluation dataset (cic-ids2017)," https://www.unb.ca/cic/datasets/ids-2017.html.

[14] "Canadian institute for cybersecurity. cicflowmeter (formerly iscxflowmeter)," https://www.unb.ca/cic/research/applications.html#CICFlowMeter.