# Implementación de un clúster basado en Hadoop y Spark para el entrenamiento de modelos de Machine Learning para la detección de ataques DDoS

José de Jesús Zapata Lara
*Instituto Nacional de Astrofísica, Óptica y Electrónica*
Tonantzintla, Puebla, México
josedj.zapata@inaoe.mx

***Resumen*—El uso de sistemas distribuidos en los últimos años se ha incrementado debido a las capacidades que ofrecen para el procesamiento y almacenamiento masivo de información, así como por los requerimientos no funcionales que proveen. En el presente trabajo se plantea la implementación de un clúster para el entrenamiento de modelos basados en Machine Learning (ML) para la detección de ataques de negación de servicio distribuidos (DDoS) utilizando el sistema de almacenamiento de archivos distribuido HDFS del framework Hadoop y la integración del framework Spark para procesamiento. Los resultados obtenidos demuestran que un clúster de alto rendimiento que utiliza la integración de Hadoop y Spark puede ser utilizado para el entrenamiento de modelos basados en ML en el contexto de ciberseguridad con buen desempeño en cuanto a precisión, tiempo de ejecución, escalabilidad horizontal y tolerancia a fallos.**

***Palabras clave*—Sistema distribuido, Clúster, Hadoop, Spark, Ataques DDoS, Machine Learning**

## I. Introducción

El incremento del uso de sistemas distribuidos en los últimos años está ampliamente relacionado con las capacidades que estos sistemas ofrecen para el procesamiento y almacenamiento masivo de datos [3]. Para la implementación y despliegue de sistemas distribuidos existen modelos de programación y diseño que pueden ser utilizados para atender necesidades específicas de un entorno determinado [11]. Uno de los principales desafíos en el área de ciberseguridad es la detección oportuna de ataques cibernéticos de naturaleza dinámica, como los ataques de negación de servicio distribuidos (DDoS por sus siglas en inglés). Por lo anterior, la tendencia en los últimos años ha sido utilizar modelos basados en Machine Learning (ML) los cuales han demostrado tener la capacidad para detectar patrones de comportamiento anómalo eficientemente sin ser programados de forma explícita [1]. Una de las principales áreas de oportunidad para los modelos basados en ML es realizar su evaluación en escenarios que puedan ser escalables [8]. El insumo principal para los esquemas de detección de ataques DDoS basados en modelos de ML son los conjuntos de datos, los cuales están integrados por archivos que pueden incrementar su volumen rápidamente, por lo cual es importante para estos esquemas utilizar infraestructura de almacenamiento y procesamiento con requerimientos no funcionales como escalabilidad y tolerancia a fallos.

En el estado del arte existen trabajos relacionados con la detección de ataques DDoS mediante cómputo distribuido y modelos basados en ML sin embargo, han sido implementados solo en un equipo de cómputo local, otros han utilizado conjuntos de datos no especializados en ataques DDoS o solamente han sido evaluados para la detección de un ataque DDoS en específico y sin utilizar para la fase de entrenamiento de los modelos las características más relevantes de los ataques DDoS para optimizar la precisión en la detección y para la reducción del tiempo de entrenamiento.

El trabajo realizado plantea una alternativa para almacenar y procesar conjuntos de datos para la fase de entrenamiento de modelos basados en ML para la detección de ataques DDoS en un entorno distribuido con escalamiento horizontal, utilizando las características más relevantes que modelan un ataque DDoS.

Las principales contribuciones del artículo son las siguientes:

- Implementación de un clúster para almacenamiento y procesamiento de las fases de entrenamiento y validación de modelos basados en ML para la detección de ataques DDoS con escalabilidad horizontal y tolerancia a fallos.
- Validación del desempeño de tres modelos basados en ML (Regresión Logística, Random Forest y Gradient Boosted Tree) en un entorno distribuido para la clasificación binaria de ataques DDoS que utilizan los vectores de ataque de Reflexión (DNS y NTP) e Inundación (TCP-SYN y UDP), utilizando las características más relevantes del conjunto de datos.

El resto del artículo está organizado de la siguiente manera: en la Sección II se describen trabajos relacionados con el uso de sistemas distribuidos basados en Hadoop y Spark para la detección de ataques DDoS. La Sección III describe los preliminares del trabajo realizado. La Sección IV describe la metodología utilizada para la implementación y la experimentación. En la sección V se presentan los resultados obtenidos y en la sección VI se presenta la discusión sobre las capacidades de evolución del trabajo realizado, así como sus limitaciones. En la sección VII se presentan las conclusiones y finalmente,

en la sección VIII se presenta la perspectiva de trabajo futuro a partir de las áreas de oportunidad identificadas.

## II. Trabajos relacionados

Mediante una revisión sistemática de la literatura se consultaron trabajos relacionados con el uso de Hadoop y Spark para la detección de ataques DDoS. La propuesta del artículo [5] presenta una solución para la detección de ataques DDoS de tipo inundación analizando bitácoras de tráfico de red en tiempo real, lo cual representa una limitante para la detección de ataques DDoS de multiples vectores.

En el trabajo [7] presentan una comparativa del desempeño de los frameworks Hadoop-Mahout y Spark como componente de un sistema para la detección de instrusiones (IDS por sus siglas en inglés), resultando el escenario de Spark con mejor desempeño. La solución propuesta fue entrenada con el conjunto de datos Network Security Lab Knowledge Discovery in Databases (NSL-KDD) el cual incluye características de ataques del tipo Probe, User to Root (U2R), Remote to Local (R2L) y Denial of Service (DoS) y se desarrolló utilizando un entorno de simulación en un equipo de cómputo portátil.

El artículo [6] presenta la propuesta de un modelo para la detección de intrusiones monitoreando el tráfico de red para el entrenamiento de varios modelos basados en ML. En la propuesta se plantea la combinación de multiples fuentes de información y el uso de multiples modelos basados en ML para mejorar el desempeño de un mecanismo de detección de intrusiones, sin embargo no se muestran resultados de la experimentación ni la información sobre el escenario donde fue realizada la implementación.

En el trabajo [12] se presenta una solución basada en Spark implementada en un entorno local para la detección de intrusiones, realizando una comparativa de su desempeño utilizando un conjunto de datos generado por los autores y algunos de los conjuntos de datos más utilizados en el estado del arte, demostrando que la solución desarrollada presenta mejores resultados utilizando el conjunto de datos autogenerado, dado que es más diverso, está actualizado e incluye un mayor número de características.

En el artículo [4] se plantea el despliegue de un escenario distribuido para la clasificación de datos mediante el modelo basado en ML de máquinas de soporte vectorial. En la solución propuesta se plantea la ejecución de procesamiento paralelo utilizando Spark y el almacenamiento de los datos en el sistema de archivos HDFS de Hadoop. El uso de Spark para la detección de ataques DDoS también se plantea de forma similar al trabajo previo en el artículo [9], pero utilizando el modelo Random Forest.

## III. Preliminares

### *III-A. Sistema distribuido tipo clúster*

Un sistema distribuido se define como aquel en el que los componentes de hardware y/o software ubicados en equipos de cómputo interconectados mediante una red se comunican y coordinan sus acciones únicamente mediante el envío de mensajes [2]. Los componentes de software fundamentales para el trabajo realizado son Hadoop y Spark. Hadoop es un marco de código abierto que se utiliza para almacenar y procesar de manera eficiente conjuntos de datos grandes de forma masiva y paralela. Spark es un sistema de código abierto para procesamiento distribuido de cargas de trabajo de gran tamaño optimizado mediante el uso de estructuras para almacenamiento de datos en memoria.

### *III-B. Ataques DDoS*

Un ataque DDoS es aquel que tiene como objetivo interrumpir el funcionamiento normal de un sistema o servicio, saturándolo con tráfico de red enviado desde múltiples fuentes, haciéndolo inaccesible para los usuarios legítimos. Uno de los principales desafíos en la detección de los ataques DDoS es su naturaleza dinámica, por lo cual un esquema de detección tradicional ha dejado de ser efectivo en la actualidad.

### *III-C. Detección de ataques DDoS utilizando ML*

Los esquemas de detección de ataques DDoS basados en modelos de Machine Learning (ML) han demostrado ser una alternativa efectiva con altos niveles de precisión y rendimiento para la detección de ataques DDoS [1]. Los modelos basados en ML son eficientes para detectar patrones mediante algoritmos de entrenamiento, lo cual los hace versátiles y con capacidades de generalización.

### *III-D. Arquitectura del clúster*

La arquitectura del clúster implementado es del tipo de alto rendimiento y utiliza nodos de cómputo configurados bajo el modelo de memoria distribuida. El clúster está integrado por un nodo maestro el cual administra la operación y asignación de tareas y por cinco nodos trabajadores que realizan las actividades de procesamiento y almacenamiento. La conectividad de los nodos del clúster es provista por una red con tecnología Gigabit Ethernet configurada con direccionamiento IP privado. Cada nodo del clúster tiene almacenamiento independiente administrado por el sistema de archivos distribuido HDFS del framework Hadoop. La administración de recursos y calendarización de trabajos en el clúster se realiza mediante el componente YARN del framework Hadoop y la ejecución de tareas de forma distribuida con procesamiento de datos en memoria es realizada y administrada mediante el framework Spark.

## IV. Metodología

Esta sección describe la metodología utilizada para el desarrollo de la propuesta, la cual estuvo integrada por las etapas de: revisión sistemática de la literatura, la selección y configuración de los elementos de hardware y software, así como la definición del escenario de pruebas para validar su funcionamiento. A continuación se describe el trabajo realizado en cada una de las etapas.

### *IV-A. Revisión sistemática de la literatura*

La revisión realizada consideró artículos de conferencia publicados en las bases de datos: Scopus, IEEE Xplore y SpringerLink. Esta revisión se desarrolló con la finalidad de identificar trabajos relacionados y considerar aquellos que pudieran servir de base.

Las preguntas de investigación utilizadas fueron las siguientes:

1. ¿Un sistema de cómputo distribuido tipo clúster puede ser utilizado para el despliegue eficiente de un escenario para el procesamiento de conjuntos de datos de gran volumen?
2. ¿Existen implementaciones en el estado del arte que utilicen Hadoop y Spark para almacenar y procesar datos en aplicaciones que requieren baja latencia?

Los términos "Hadoop" y "Spark" fueron utilizados en la cadena de búsqueda y la selección de los trabajos se realizó con base en los siguientes criterios de inclusión y exclusión:

- **Criterios de inclusión:** Artículos que utilicen la integración de Hadoop y Spark en el contexto de ciberseguridad y que hayan sido publicados dentro del periodo de 2023 a 2025
- **Criterio de exclusión:** Que el artículo no corresponda a los criterios de inclusión.

El resumen del análisis de la búsqueda se muestra en la tabla I.

Tabla I
ANÁLISIS DE LA BÚSQUEDA

| **Número de trabajos** | | | |
|---|---|---|---|
| **Fuente** | ***Encontrados*** | ***Analizados*** | ***Relacionados al proyecto*** |
| Scopus | 57 | 6 | 5 |
| IEEE Xplore | 21 | 4 | 1 |
| SpringerLink | 61 | 8 | 3 |

### *IV-B. Selección y configuración de elementos del clúster*

En esta sección se describen los elementos de hardware, software y servicios utilizados. El diagrama del clúster se muestra en la figura 1.

Las especificaciones técnicas de los equipos de cómputo utilizados se muestran en la tabla II.

Para los nodos de cómputo del clúster se utilizó el sistema operativo Linux Ubuntu Server 22.04.5 LTS y fueron interconectados mediante una red privada con el segmento 192.168.2.0/24. Se habilitó el acceso SSH entre los nodos de cómputo mediante llaves para la autenticación sin contraseña y se realizó la instalación y configuración de Hadoop 3.4.1 y de Spark 3.5.6, ambos en modo clúster.

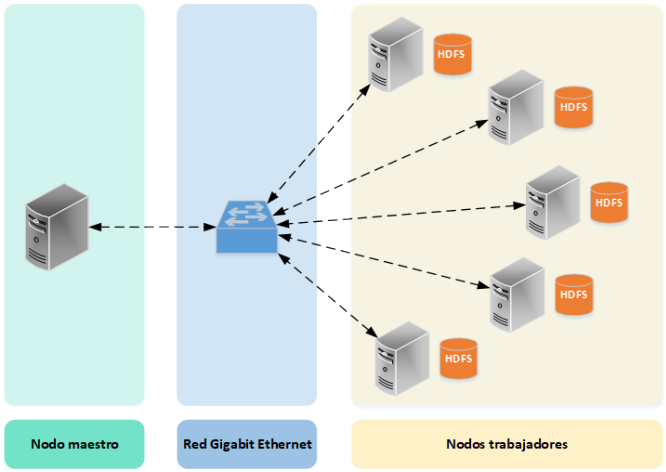


Figura 1. Diagrama de la implementación del clúster.

Tabla II
HARDWARE CLÚSTER HADOOP-SPARK

| **Especificaciones técnicas** | | | |
|---|---|---|---|
| **Descripción** | ***Procesador*** | ***Memoria*** | ***Disco*** |
| Nodo maestro | Intel Xeon X5550 (4 cores) | 12 GB | 1 TB |
| Nodos trabajadores | Intel Xeon X5550 (8 cores) | 16 GB | 1 TB |
| **Cantidades totales** | **44 cores** | **92 GB** | **6 TB** |

### *IV-C. Escenario de pruebas*

El conjunto de datos utilizado fue CICDDoS-2019 [10], el cual está integrado por capturas de tráfico de ataques DDoS con 80 características y 15 etiquetas. Del conjunto de datos seleccionado se utilizaron los ataques DDoS con los vectores de ataque de reflexión (DNS y NTP) e inundación (TCP SYN y UDP Flood). Los modelos basados en ML seleccionados para la clasificación binaria de ataques DDoS fueron: Regresión Logística (LR), Random Forest (RF) y Gradient Boosted Tree (GBT), con una distribución del 80 % del conjunto de datos para el entrenamiento y el 20 % restante para la validación. Las características utilizadas para el entrenamiento de cada modelo seleccionado se muestran en la tabla III.

Para la evaluación del desempeño de los modelos basados en ML se utilizaron las mediciones Accuracy, Precision, Recall y F1-Score. La descripción de las métricas utilizadas se presenta en la tabla IV.

## V. RESULTADOS

La detección de ataques DDoS fue realizada utilizando tres escenarios: el primero mediante el uso de un solo clasificador (LR), el segundo utilizando un ensamble de clasificadores tipo Bagging (RF) y el tercero mediante un ensamble de clasificadores de tipo Boosting (GBT). Se ha demostrado que el uso de un ensamble de clasificadores presenta mejor desempeño que el uso de un solo clasificador y mayor capacidad de generalización en la detección de ataques DDoS [8]. La comparativa de los modelos seleccionados se realizó para

Tabla III
ATAQUES DDOS Y SUS CARACTERÍSTICAS

| **Ataque DDoS** | ***Características*** |
|---|---|
| Reflexión DNS | Longitud máxima del paquete, longitud máxima del paquete hacia el destino, longitud mínima del paquete hacia el destino, tamaño promedio de paquete y longitud mínima del paquete. |
| Reflexión NTP | Promedio de bytes en un subflujo hacia el origen, longitud total de los paquetes hacia el destino, desviación estándar de la longitud del paquete hacia el destino, tamaño mínimo del segmento hacia el destino y tiempo mínimo entre dos paquetes consecutivos en un flujo. |
| Inundación TCP-SYN | Número de paquetes con el indicador ACK establecido en el encabezado TCP, tamaño inicial de la ventana TCP hacia el destino en bytes, tamaño mínimo del segmento hacia el destino, tiempo total entre dos paquetes enviados hacia e destino y secuencia de paquetes entre origen y destino. |
| Inundación UDP | Secuencia de paquetes entre origen y destino, desviación estándar de la longitud del paquete hacia el destino desviación estándar de la longitud del paquete, tamañ mínimo del segmento hacia el destino y el identificado del protocolo. |

Tabla IV
MEDICIÓN DEL DESEMPEÑO DE MODELOS BASADOS EN ML

| **Medición** | ***Descripción*** | ***Fórmula*** |
|---|---|---|
| Accuracy | Proporción de todas las clasificaciones que fueron correctas, ya sean positivas o negativas | $\frac{TN+TP}{TN+TP+FN+FP}$ |
| Precision | Proporción de todas las clasificaciones positivas del modelo que son realmente positivas | $\frac{TP}{TP+FP}$ |
| Recall | Proporción de todos los positivos reales que se clasificaron correctamente como positivos | $\frac{TP}{TP+FN}$ |
| F1-Score | Media armónica de Precision y Recall, utilizada para medir con mayor precisión la exactitud del modelo en conjuntos de datos desequilibrados | $2 * \frac{Precision*Recall}{Precision+Recall}$ |

validar su desempeño en un entorno distribuido y de esta manera explorar su viabilidad de integración en una solución escalable. En los resultados obtenidos se pudo observar que los modelos basados en el ensamble Random Forest y Gradient Boosted Tree presentan un mejor rendimiento en general en comparativa con el modelo de Regresión Logística, debido a que están integrados por múltiples clasificadores que retroalimentan al modelo y de esta manera obtienen mejores resultados en sus predicciones. Al comparar los dos modelos de tipo ensamble fue posible identificar que Random Forest es más rápido que Gradient Boosted Tree, ya que utiliza de forma paralela múltiples clasificadores mediante árboles de decisión, los cuales realizan su entrenamiento sobre muestras aleatorias del conjunto de datos (subconjuntos), obteniendo su predicción final a partir del promedio de las predicciones individuales de cada clasificador, minimizando de esta manera el porcentaje de error. Sin embargo, el modelo basado en el ensamble Gradient Boosted Tree el cual también utiliza árboles de decisión como clasificadores débiles sobre el conjunto de datos completo, por su naturaleza de retroalimentación secuencial a partir de los errores identificados en los clasificadores previos, presentó mejor resultado en cuanto a precisión. Por lo anterior, es importante mencionar que una solución basada en un ensamble de clasificadores en el escenario utilizado, ofrece una mejor solución y a partir de los resultados obtenidos se puede tomar la decisión de utilizar determinado ensamble considerando un balance entre rapidez y precisión a partir de las necesidades específicas del entorno.

Los resultados obtenidos de la evaluación del desempeño de cada modelo se muestran en la gráfica de la figura 2.

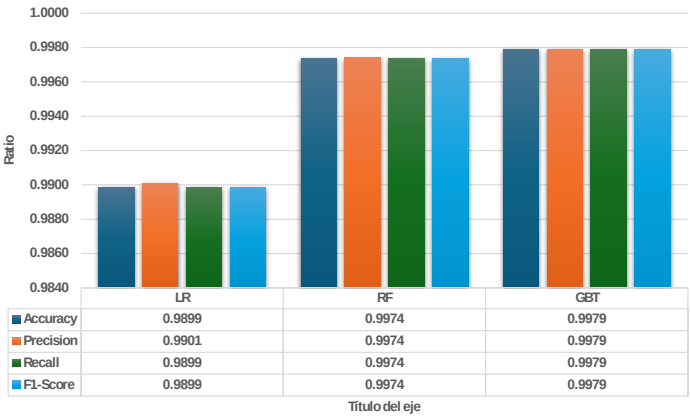


Figura 2. Desempeño de los modelos basados en ML utilizados para la detección de ataques DDoS.

Los resultados con el tiempo de ejecución de cada modelo se muestran en la tabla V.

Tabla V
TIEMPOS DE EJECUCIÓN

| **Modelo ML** | ***Tiempo de ejecución (seg)*** |
|---|---|
| LR | 57.8 |
| RF | 58.3 |
| GBT | 85.5 |

En las figuras 3, 5 y 4 se muestran las matrices de confusión del rendimiento de los modelos basados en ML que fueron utilizados, a través de las cuales fue posible realizar la comparativa de la eficacia de cada modelo en la detección de ataques TCP-SYN.

## VI. DISCUSIÓN

El desarrollo de la implementación de un clúster para el entrenamiento de modelos basados en ML para la detección de ataques DDoS puede ser considerado como una alternativa eficiente y escalable para líneas de investigación que requieren capacidades de cómputo de alto desempeño y escalabilidad, pero es importante mencionar que el trabajo realizado presenta diversas áreas de oportunidad. Una limitación identificada es

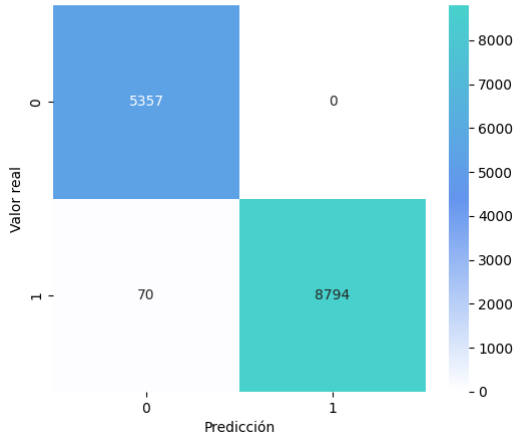


Figura 3. Regresión Logística (SYN).

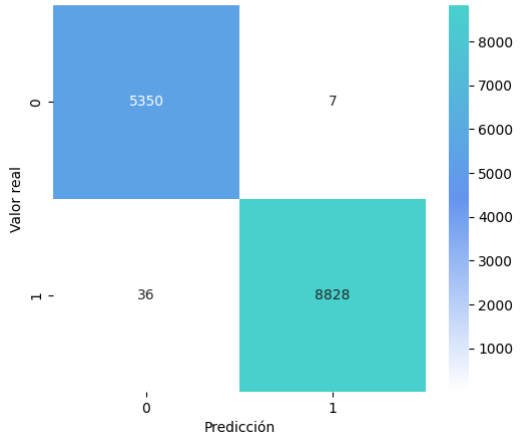


Figura 4. Gradient Boosted Tree (SYN).

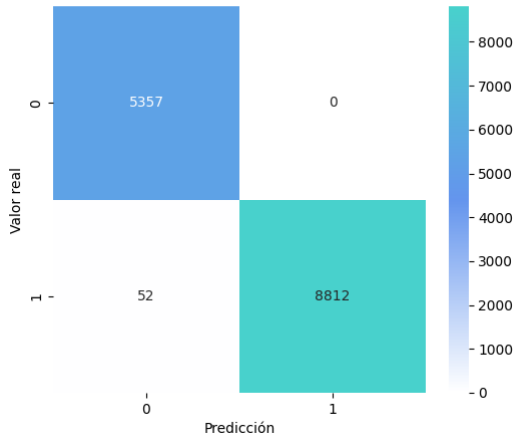


Figura 5. Random Forest (SYN).

que solo se utilizó el dataset CICDDoS-2019, que si bien incluye varios tipos de ataque DDoS este no incluye datos sobre ataques DDoS más recientes, como los de tipo low rate o de múltiples vectores. Otras limitaciones en el conjunto de datos es que no está balanceado lo que puede generar una tasa alta de falsos positivos y que su falta de diversidad de ataques DDoS puede provocar sobreajuste en los modelos. Adicionalmente, es importante mencionar que se utilizaron solamente tres modelos basados en ML lo cual representa solo una muestra representativa de los modelos existentes basados en ML. En el contexto de ciberseguridad se ha identificado que un área de oportunidad para los mecanismos de seguridad que utilizan modelos basados en ML es lograr que sus soluciones sean escalables para el almacenamiento de los conjuntos de datos que utilizan y para mejorar el tiempo procesamiento y validación. Por lo anterior, se plantea que el trabajo realizado podría ser utilizado como plataforma base para otras líneas de investigación en ciberseguridad como la detección de malware, para la generación y almacenamiento de conjuntos de datos a partir de redes generativas antagónicas (GAN's) la cual es un área muy relevante debido a la falta de conjuntos de datos grandes y diversos disponibles para el entrenamiento de los modelos y para la detección de amenazas en ecosistemas específicos como Internet de las Cosas, Redes Definidas por Software o Cómputo en la Nube. El presente trabajo puede evolucionar hacia las líneas de investigación mencionadas previamente a través del uso de conjuntos de datos especializados en esas áreas y mediante la integración al escenario del clúster de una arquitectura basada en microservicios que permita realizar el despliegue rápido y escalable de múltiples modelos basados en ML para la detección de diversos ataques cibernéticos de forma simultánea.

## VII. Conclusión

En este trabajo se presenta la implementación de un clúster para la fase de entrenamiento de modelos basados en ML para la detección de ataques DDoS, con capacidades de almacenamiento y procesamiento escalables mediante cómputo distribuido. Durante el desarrollo del trabajo fue posible evaluar el desempeño de los modelos LR, RF y GBT para la detección de ataques DDoS de tipo Reflectivo (DNS y NTP) y por Inundación (TCP SYN Flood y UDP Flood). El entrenamiento de los modelos fue realizado a partir de las características más relevantes de cada uno de los ataques DDoS seleccionados. Los resultados de la experimentación demostraron que la fase de entrenamiento de los modelos utilizados se realizó en el orden de 57 a 86 segundos y que el modelo que obtuvo mejores resultados con respecto al balance entre su efectividad y tiempo de ejecución fue Random Forest, mientras que el modelo Gradient Boosted Tree obtuvo los mejores resultados en cuanto a precisión. El escenario para cómputo distribuido tipo clúster implementado demostró que puede ser una alternativa para proveer de forma transparente paralelización de tareas, tolerancia a fallos, redundancia y balanceo de carga para mecanismos de detección de ataques DDoS que utilizan modelos basados en ML, por lo cual puede

ser una plataforma que sirva de base para el despliegue de estos mecanismos un ambiente de producción.

## VIII. Trabajo futuro

La implementación del clúster si bien provee requerimientos no funcionales de escalabilidad y tolerancia a fallos también presenta áreas de mejora. Un área de oportunidad identificada es la necesidad de complementar la arquitectura del clúster con un módulo para captura de tráfico y un componente que permita utilizar los modelos basados en ML que ya fueron entrenados para realizar detección de ataques DDoS en tiempo real, así como incluir un componente para monitoreo y actualización del entrenamiento de los modelos para detectar ataques DDoS más recientes. Una línea de investigación que también se considera como trabajo futuro es utilizar el clúster como componente de procesamiento para realizar experimentación con modelos de Deep Learning para la detección de ataques DDoS de día cero. Para complementar la evaluación del clúster en la detección de ataques DDoS en un escenario real, se considera utilizar métricas que permitan evaluar su desempeño considerando las variables del entorno como: uso de CPU, uso de memoria RAM, ancho de banda, así como métricas específicas de ciberseguridad como: Mean Time To Detect (MTTD) y Mean Time To Respond (MTTR).

## IX. Agradecimientos

El presente trabajo fue realizado en el marco del curso de Sistemas Distribuidos del posgrado en Ciencias y Tecnologías de Seguridad, por lo cual se agradece especialmente a la profesora del curso por su asesoría y retroalimentación. Se extiende el agradecimiento a la SECIHTI por el apoyo brindado a través de la beca para el estudio del posgrado, al INAOE por las facilidades otorgadas para realizar estudios de posgrado en un entorno de excelencia académica, a los profesores del núcleo académico por sus enseñanzas, así como a los autores de las publicaciones y de los recursos utilizados.

## Referencias

[1] Abdallah, Amira Mahamat, Aysha Saif Rashed Obaid Alkaabi, Ghaya Bark Nasser Douman Alameri, Saida Hafsa Rafique, Nura Shifa Musa y Thangavel Murugan: *Cloud Network Anomaly Detection Using Machine and Deep Learning Techniques— Recent Research Advancements*. IEEE Access, 12:56749–56773, 2024.

[2] Coulouris, George, Jean Dollimore, Tim Kindberg y Gordon Blair: *Distributed Systems: Concepts and Design*. Addison-Wesley Publishing Company, USA, 5th edición, 2011, ISBN 0132143011.

[3] Dai, Fei, Md Akbar Hossain y Yi Wang: *State of the Art in Parallel and Distributed Systems: Emerging Trends and Challenges*. Electronics, 14(4), 2025, ISSN 2079-9292. https://www.mdpi.com/2079-9292/14/4/677.

[4] Groenewald, Elma Sibonghanoy, Sarath Babu Dodda, Coenrad Adolph Groenewald, Amol Dhumane, Aditi Sharma y Ketan Kotecha: *Parallelized Distributed Computing for Large-Scale Support Vector Machine Training*. En Goar, Vishal, Aditi Sharma, Jungpil Shin y M. Firoz Mridha (editores): *Deep Learning and Visual Artificial Intelligence*, páginas 445–460, Singapore, 2024. Springer Nature Singapore, ISBN 978-981-97-4533-3.

[5] Hameed, Sufian y Usman Ali: *Efficacy of Live DDoS Detection with Hadoop*. En *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, páginas 488–494, 2016.

[6] Ivanova, Petya y Todor Tagarev: *Challenges and Opportunities for Network Intrusion Detection in a Big Data Environment*. En Tagarev, Todor y Nikolai Stoianov (editores): *Digital Transformation, Cyber Security and Resilience*, páginas 93–106, Cham, 2024. Springer Nature Switzerland, ISBN 978-3-031-44440-1.

[7] Jawad, Wasnaa y Abbas Al-Bakry: *Performance evaluation of Hadoop and spark distributed frameworks by using them to build an intelligent intrusion detection system*. En *American Institute of Physics Conference Series*, volumen 3211 de *American Institute of Physics Conference Series*, página 030005, Mayo 2025.

[8] Ouhssini, Mohamed, Karim Afdel, Mohamed Akouhar, Elhafed Agherrabi y Abdallah Abarda: *Advancements in detecting, preventing, and mitigating DDoS attacks in cloud environments: A comprehensive systematic review of state-of-the-art approaches*. Egyptian Informatics Journal, 27:100517, 2024, ISSN 1110-8665. https://www.sciencedirect.com/science/article/pii/S111086652400080X.

[9] Raparthi, Mohan, Monika Soni, Vipin Tiwari, Amol Dhumane y Rahul Sharma: *Scalable Implementation of Random Forests for Big Data Classification on Cloud Infrastructure*. En Goar, Vishal, Aditi Sharma, Jungpil Shin y M. Firoz Mridha (editores): *Deep Learning and Visual Artificial Intelligence*, páginas 493–512, Singapore, 2024. Springer Nature Singapore, ISBN 978-981-97-4533-3.

[10] Sharafaldin, Iman, Arash Habibi Lashkari, Saqib Hakak y Ali A. Ghorbani: *Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy*. En *2019 International Carnahan Conference on Security Technology (ICCST)*, páginas 1–8, 2019.

[11] Steen, Maarten y Andrew S. Tanenbaum: *A brief introduction to distributed systems*. Computing, 98(10):967–1009, Octubre 2016, ISSN 0010-485X. https://doi.org/10.1007/s00607-016-0508-7.

[12] Vashishtha, Lalit Kumar y Kakali Chatterjee: *Strengthening cybersecurity: TestCloudIDS dataset and SparkShield algorithm for robust threat detection*. Computers & Security, 151:104308, 2025, ISSN 0167-4048. https://www.sciencedirect.com/science/article/pii/S016740482400614X.