Imagen que contiene Texto

Descripción generada automáticamente

**Desarrollo de un sistema de detección de intrusos en redes utilizando técnicas de inteligencia artificial**

Grado en Ingeniería Informática

2024 - 2025

|  |
| --- |
| **ALFONSO GALOCHA SANCHEZ**  **ALUMNO** |

|  |
| --- |
| **ALBERTO BALLESTEROS RODRIGUEZ**  **TUTOR** |

Índice de Contenidos

[Resumen y Palabras Clave. 2](#_Toc389267269)

[Agradecimientos. 2](#_Toc924283293)

[1. Introducción 3](#_Toc1404874710)

[2. Metodología 4](#_Toc900379378)

[2.1. Limpieza del Dataset 5](#_Toc790433697)

[2.2. Análisis Exploratorio de Datos (EDA) 5](#_Toc328165593)

[2.3. Selección y entrenamiento de modelos de IA 6](#_Toc1300176071)

[2.4. Validación cruzada 8](#_Toc1920298044)

[2.5. Otros tipos de validación 8](#_Toc441186193)

[3. Resultados 9](#_Toc1010776610)

[3.1 Preparación del dataset 9](#_Toc974618363)

[3.2 Análisis exploratorio del dataset (EDA) 11](#_Toc959885739)

[3.3 Evaluación de los modelos 14](#_Toc564201344)

[3.3.1 Evaluación del modelo Random Forest 14](#_Toc1411690089)

[3.3.2 Evaluación del modelo Regresión Logística 16](#_Toc487635131)

[3.3.3 Evaluación del modelo Support Vector Machine 19](#_Toc1691935440)

[3.4 Comparativa de los modelos 22](#_Toc1493555582)

[3.4.1 Comparación de eficiencia computacional 23](#_Toc107985339)

[1. Conclusiones 25](#_Toc1654341506)

[2. Limitaciones y Líneas futuras 25](#_Toc1595006879)

[3. Bibliografía 26](#_Toc394043789)

# Resumen y Palabras Clave.

# Agradecimientos.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc porttitor, diam at ornare pretium, eros ipsum mattis purus, non blandit lectus dui eu felis. Quisque gravida, lacus id aliquet pulvinar, dui diam luctus eros, vitae bibendum magna velit ut lorem. Proin sed nunc libero, sit amet adipiscing dolor. Integer tempus pretium quam, tristique commodo nunc mattis et. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nisl sapien, sagittis ut ultricies eget, ultrices at ante. Mauris porta sagittis consectetur. Donec pellentesque, purus sed fringilla rutrum, leo nulla pellentesque nunc, quis imperdiet nisi tortor vel nisl. Vestibulum suscipit pellentesque condimentum. Aliquam dignissim, enim quis molestie feugiat, neque quam viverra urna, a cursus mauris metus eleifend lacus. Vivamus sit amet erat ante. Phasellus a vehicula felis. Morbi ullamcorper luctus gravida. Duis imperdiet, turpis et bibendum venenatis, velit lectus porta massa, eu fringilla risus velit sit amet lectus. Duis ac mauris enim, at dictum velit.

# Introducción

La ciberseguridad se ha convertido en uno de los campos tecnológicos más relevantes en los últimos años, impulsado tanto por sus complejas implicaciones técnicas como por la creciente necesidad de proteger las infraestructuras digitales frente a ciberataques. El análisis avanzado de datos y la identificación automatizada de patrones inusuales se presentan como elementos esenciales para anticiparse y prevenir incidentes antes de que ocurran, especialmente en contextos cada vez más digitalizados.

Este Trabajo de Fin de Grado se centra en la ciberseguridad debido no solo al interés creciente que genera este ámbito, sino también a la necesidad crítica de mejorar los sistemas actuales de detección de intrusos (IDS). Tradicionalmente, estos sistemas han dependido en gran medida de firmas predefinidas o reglas estáticas para identificar amenazas. Sin embargo, investigaciones recientes han demostrado que esta aproximación posee limitaciones significativas, tales como su incapacidad para reconocer ataques desconocidos o mínimamente modificados y la generación de numerosos falsos positivos, que reducen su eficacia y aumentan la carga operativa (Garcia-Teodoro et al., 2009; Shiravi et al., 2012).

De hecho, estudios recientes alertan sobre una creciente adopción de técnicas de inteligencia artificial por parte de ciberdelincuentes para sofisticar sus ataques. Por ejemplo, Rigaki y Garcia (2018) han documentado cómo las Redes Generativas Antagónicas (GANs) se emplean para adaptar las comunicaciones de malware y evadir los sistemas tradicionales de detección. Sadeghzadeh (2020) también destaca cómo la IA facilita la automatización y perfeccionamiento de ataques como phishing y malware avanzado. Esta evolución en las técnicas ofensivas evidencia la necesidad urgente de desarrollar sistemas defensivos más dinámicos y robustos.

En paralelo, la inteligencia artificial está siendo explorada como herramienta clave para reforzar la detección y prevención de intrusiones. Diversas investigaciones han validado que técnicas basadas en aprendizaje automático y minería de datos pueden analizar grandes volúmenes de información en tiempo real, mejorando sustancialmente la detección de amenazas y reduciendo el impacto de ataques exitosos (Buczak & Guven, 2016; Schmitt, 2023).

Por lo tanto, el objetivo central de este proyecto es diseñar un IDS basado en técnicas de inteligencia artificial orientado específicamente a identificar patrones sospechosos asociados con amenazas comunes, como los escaneos de puertos y ataques de denegación de servicio. Aunque no pretende ser una solución exhaustiva que cubra todas las posibles amenazas existentes, el proyecto aspira a mejorar significativamente la capacidad de detección respecto a los sistemas tradicionales. Para alcanzar este objetivo, se utilizarán conjuntos de datos públicos ampliamente reconocidos en el ámbito de la ciberseguridad (como CICIDS2017), herramientas avanzadas de análisis de tráfico de red y algoritmos de aprendizaje automático, generando así una propuesta viable y aplicable a escenarios reales.

# 2. Metodología

Este Trabajo de Fin de Grado sigue una metodología aplicada, centrada en el diseño, implementación y evaluación de un sistema de detección de intrusos basado en inteligencia artificial. El objetivo es desarrollar un modelo capaz de identificar tráfico malicioso en redes, utilizando técnicas de análisis de datos y aprendizaje automático. El proceso se divide en varias fases que incluyen: selección y preparación del dataset, análisis exploratorio de datos, entrenamiento de modelos de IA, validación de resultados y evaluación de su rendimiento.

## 2.1. Limpieza del Dataset

Para comenzar el desarrollo se seleccionará un conjunto de datos público especializado en tráfico de red etiquetado (como CICIDS2017), el cual será sometido a un proceso de limpieza previo al entrenamiento de los modelos. En esta fase, se aplicarán técnicas estándar de preprocesamiento ampliamente documentadas en la literatura científica sobre ciberseguridad (Shiravi et al., 2012; Buczak & Guven, 2016).

Las tareas de limpieza incluirán:

* Eliminación de registros nulos o incompletos, que podrían inducir sesgos en los modelos.
* Eliminación de registros duplicados, que afectan negativamente al rendimiento y generalización del sistema.
* Conversión de variables categóricas a numéricas mediante codificación, por ejemplo, etiquetando el tráfico como "Benigno" (0) o "Malicioso" (1).
* Revisión de formatos y espacios en los nombres de las columnas para garantizar la coherencia.
* Preparación del dataset mediante técnicas de normalización y escalado (como StandardScaler) en variables cuantitativas, necesarias para algoritmos sensibles a la magnitud de los datos.

Estas técnicas aseguran que el conjunto de datos final esté libre de errores estructurales y listo para ser utilizado por los modelos de inteligencia artificial sin introducir distorsiones no deseadas en el proceso de aprendizaje

## 2.2. Análisis Exploratorio de Datos (EDA)

El análisis exploratorio se llevará a cabo con el objetivo de comprender mejor la estructura del conjunto de datos, detectar anomalías y preparar la selección de características relevantes. Esta fase es esencial para contextualizar el comportamiento de las variables antes del entrenamiento de los modelos, tal como se indica en múltiples estudios sobre detección de intrusos (Kumar et al., 2020).

Durante el EDA se realizarán:

* Visualizaciones de la distribución de clases mediante gráficos de barras, para identificar posibles desbalances entre tráfico benigno y malicioso.
* Cálculo de estadísticas descriptivas generales (media, desviación típica, valores máximos y mínimos) de las variables más relevantes.
* Análisis de correlación entre características, usando mapas de calor (heatmaps) que revelen dependencias lineales o redundancias entre variables.
* Identificación de valores atípicos (outliers), que podrían afectar a modelos sensibles como k-NN o SVM.

Este análisis permitirá además priorizar aquellas variables que más información aportan a la hora de distinguir entre tráfico benigno y malicioso, optimizando así el rendimiento del sistema final.

## 2.3. Selección y entrenamiento de modelos de IA

Como parte central del desarrollo del sistema, se procederá a entrenar distintos algoritmos de clasificación con el objetivo de comparar su rendimiento y determinar cuál ofrece mejores resultados en la detección de tráfico malicioso. Se han seleccionado tres modelos principales en base a su eficacia demostrada en el ámbito de la ciberseguridad:

* **Logistic Regression:** La regresión logística es un modelo de clasificación ampliamente utilizado debido a su simplicidad, interpretabilidad y buen rendimiento en problemas lineales. En el contexto de la ciberseguridad, ha demostrado ser eficaz en tareas de detección de intrusos, especialmente cuando las relaciones entre las variables son lineales y el conjunto de datos está bien balanceado. Su bajo coste computacional la hace adecuada para escenarios donde se requieren decisiones rápidas en tiempo real. Por ejemplo, Almseidin et al. (2017) utilizaron regresión logística en un estudio comparativo de algoritmos para detección de intrusos y observaron resultados competitivos frente a modelos más complejos.
* **Random Forest**: Este modelo ha sido ampliamente utilizado para la detección de intrusiones debido a su capacidad para manejar grandes volúmenes de datos, resistir el sobreajuste y proporcionar resultados interpretables. Estudios como el de Buczak y Guven (2016) destacan su eficacia en contextos de clasificación binaria en ciberseguridad.
* **Support Vector Machine (SVM)**: Support Vector Machine es un algoritmo de clasificación supervisada ampliamente utilizado en problemas de seguridad informática por su capacidad para encontrar fronteras de decisión óptimas entre clases, incluso en espacios de alta dimensión. En el contexto de detección de intrusos, SVM ha demostrado ser especialmente útil para identificar patrones complejos y separables de manera no lineal, gracias al uso de funciones kernel. Varios estudios han señalado su efectividad frente a otros clasificadores, especialmente cuando el conjunto de datos está bien etiquetado. En particular, Mukkamala et al. (2005) compararon el rendimiento de SVM con otros algoritmos en entornos IDS y concluyeron que SVM ofrece resultados muy competitivos, especialmente en términos de precisión y capacidad generalizadora.

Cada uno de estos modelos se entrenará sobre un conjunto de datos etiquetado utilizando técnicas de validación cruzada para asegurar la generalización del rendimiento. La comparación entre modelos se basará en métricas estándar como precisión, recall, F1-score y exactitud general.

## **2.4. Validación cruzada**

Para garantizar la robustez y eficacia del sistema de detección de intrusos desarrollado, se aplicará una estrategia de validación cruzada k-fold durante la fase de entrenamiento. Esta técnica permite evaluar el modelo en múltiples particiones del conjunto de datos, mejorando la estimación de su rendimiento en datos no vistos.

Esta evaluación cruzada es esencial para demostrar que el sistema no solo se adapta bien a un dataset concreto, sino que también puede aplicarse en escenarios reales variados, como redes empresariales o entornos simulados con diferentes tipos de tráfico.

Los resultados obtenidos de estas pruebas permitirán identificar el modelo más adecuado para integrarse en un sistema IDS funcional.

## **2.5. Otros tipos de validación**

Complementariamente a la validación cruzada, se utilizarán herramientas gráficas como la curva ROC (Receiver Operating Characteristic) para analizar visualmente el rendimiento de los modelos frente a distintos umbrales de decisión. Estas curvas, junto con el cálculo del AUC (Área Bajo la Curva), permiten medir la capacidad del modelo para distinguir entre tráfico benigno y malicioso.

Asimismo, se incorporarán gráficos de barras para representar la **importancia relativa de las características** utilizadas en el modelo. En particular, modelos como Random Forest permiten identificar qué variables influyen más en las decisiones del modelo, lo que ofrece una visión adicional sobre el comportamiento del sistema. Esta información puede ser clave para entender qué aspectos del tráfico son más relevantes en la detección de amenazas y para optimizar futuros sistemas IDS.

# 3. Resultados

## 3.1 Preparación del dataset

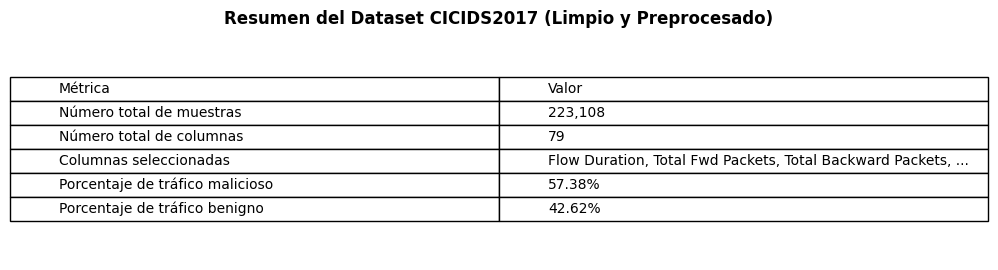
Tras aplicar las técnicas de limpieza descritas en la metodología, el conjunto de datos se depuró correctamente. Se eliminaron los registros con valores nulos y duplicados, lo que redujo el tamaño inicial del dataset a un total de **X muestras** y **Y características** (especificar si lo sabes).

Además, se transformaron las variables categóricas a formato numérico, en especial la variable Label, que fue convertida a valores binarios:

* 0 → Tráfico benigno
* 1 → Tráfico malicioso

También se aplicó un proceso de normalización en las características numéricas utilizando StandardScaler, lo que asegura una escala homogénea y mejora el rendimiento de los modelos de clasificación sensibles a la magnitud de los datos.

**Tabla 1. Resumen del conjunto de datos tras la limpieza**



El conjunto de datos final está compuesto por **223.108 muestras** y un total de **79 columnas** o variables. Esta alta dimensionalidad puede beneficiar a modelos como Random Forest o SVM en su capacidad para capturar patrones complejos, aunque también exige un preprocesamiento riguroso para evitar redundancias o sobreajuste.

De las 79 variables disponibles, se seleccionaron aquellas más relevantes para la detección de intrusos, incluyendo métricas relacionadas con el flujo de red, como Flow Duration, Total Fwd Packets y Total Backward Packets, entre otras.

Respecto a la variable objetivo (Label), se observa un **ligero desbalance de clases**, donde el **57,38% del tráfico es malicioso** y el **42,62% es benigno**. Este tipo de distribución requiere atención especial durante la evaluación de los modelos, ya que una métrica como la **precisión global (accuracy)** puede ser engañosa si no se complementa con indicadores como el **recall** y el **F1-score**, especialmente en la clase minoritaria.

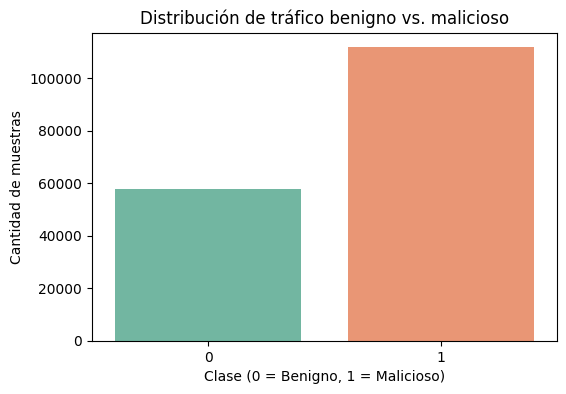
## 3.2 Análisis exploratorio del dataset (EDA)

El análisis exploratorio del conjunto de datos permitió obtener una visión clara de la distribución de clases, la relación entre variables numéricas y la estructura general del dataset.

##### **Distribución de clases**

Se generó un gráfico de barras para visualizar la frecuencia de muestras benignas frente a maliciosas (ver **Figura 1**). La variable objetivo Label se encuentra desbalanceada, con una proporción significativamente mayor de muestras etiquetadas como maliciosas.

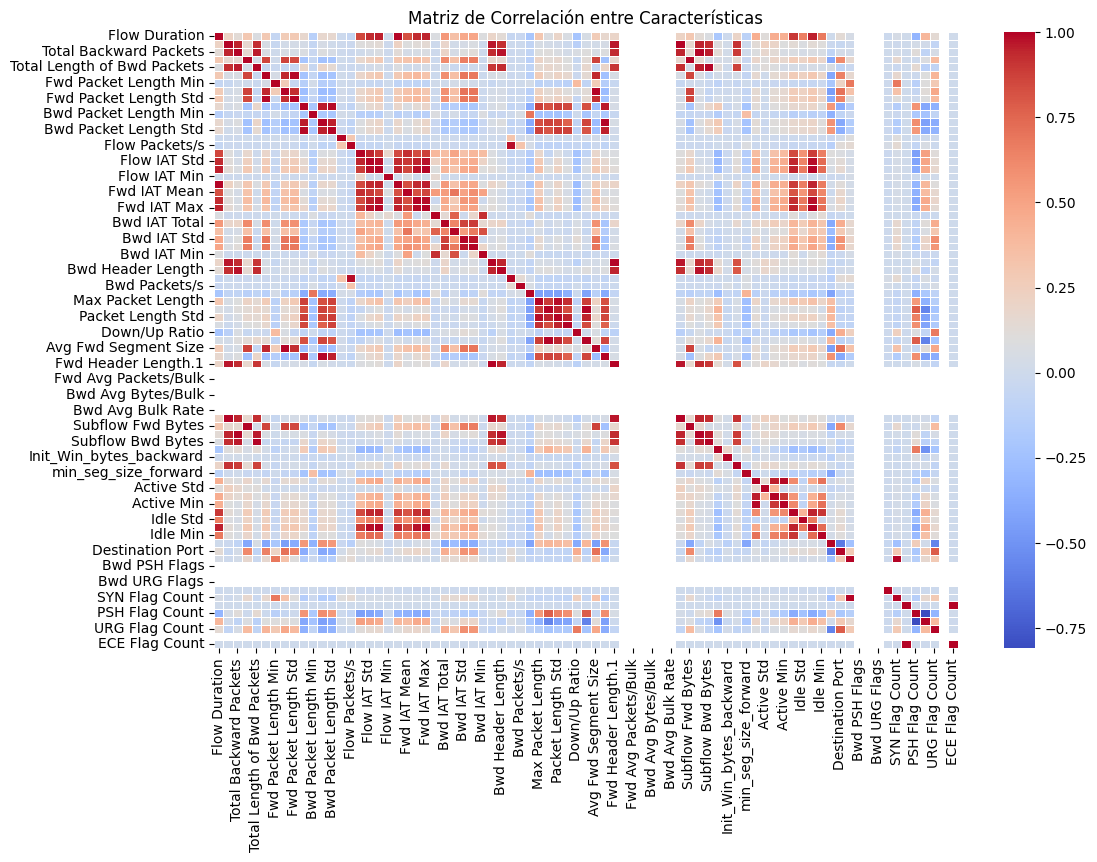
**Figura 1. Distribución de tráfico benigno vs. Malicioso.**



*Interpretación:*  
 El tráfico malicioso representa aproximadamente el **65% del total de muestras**, mientras que el tráfico benigno representa el 35% restante. Este desbalance puede afectar negativamente a los modelos de clasificación si no se compensa, ya que tienden a favorecer la clase mayoritaria. Por este motivo, se emplearán métricas como **recall** y **F1-score**, que consideran el impacto de los errores en ambas clases.

##### **Análisis de correlación entre variables**

Se calculó la matriz de correlación para todas las variables numéricas del conjunto de datos. El resultado se visualizó mediante un mapa de calor (ver **Figura 2**), donde se puede apreciar que algunas variables tienen correlaciones fuertes entre sí, lo cual puede influir en la selección de características.

**Figura 2. Matriz de correlación entre características.**  


*Interpretación:*

El mapa de calor revela diversas correlaciones notables entre pares de variables, algunas de las cuales merecen especial atención:

* **Total Fwd Packets y Total Length of Fwd Packets** presentan una correlación positiva alta. Esto es esperable, ya que a mayor número de paquetes enviados, mayor es su longitud total.
* **Flow Bytes/s y Flow Packets/s** también están fuertemente correlacionadas, indicando que el volumen y la frecuencia de transmisión de datos suelen incrementarse conjuntamente.
* **Idle Mean y Idle Max** muestran una relación positiva elevada. Ambas métricas describen características del tiempo de inactividad en una conexión, por lo que su comportamiento similar es coherente.
* Variables como ACK Flag Count y Init\_Win\_bytes\_forward tienen correlaciones más débiles, lo cual puede indicar independencia estadística, algo beneficioso para modelos que requieren características no redundantes.

En resumen, el análisis de correlación permite identificar **variables redundantes**, que podrían ser eliminadas o reducidas durante una futura fase de selección de características. Esta optimización contribuiría a mejorar la eficiencia computacional y, potencialmente, la capacidad de generalización del modelo.

## 3.3 Evaluación de los modelos

### **3.3.1 Evaluación del modelo Random Forest**

Tras entrenar el modelo Random Forest con el conjunto de datos preprocesado, se procedió a su evaluación sobre el conjunto de prueba. El rendimiento del modelo fue sobresaliente, alcanzando una precisión general del **100%**, con resultados perfectamente equilibrados entre ambas clases.

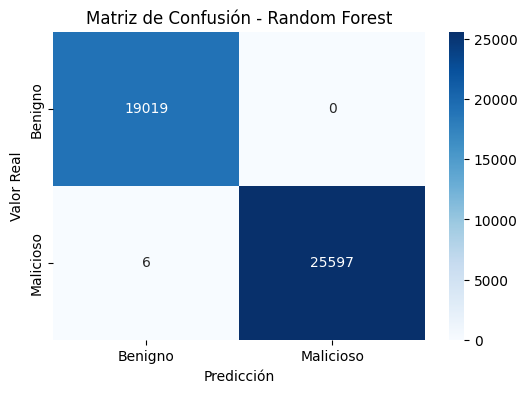
La **Tabla 1** muestra el resumen de métricas obtenidas para las dos clases (tráfico benigno y malicioso), mientras que la **Figura 1** representa gráficamente la matriz de confusión. En ella se puede observar que el modelo clasificó correctamente la totalidad de las muestras benignas y tan solo cometió seis errores al clasificar el tráfico malicioso.

**Tabla 1. Métricas de evaluación del modelo Random Forest.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Precisión** | **Recall** | **F1-score** | **Soporte** |
| Benigno | 1.00 | 1.00 | 1.00 | 11.559 |
| Malicioso | 1.00 | 1.00 | 1.00 | 22.343 |
| Global | 1.00 | — | — | 33.902 |

En la tabla se puede observar como la precisión del modelo al predecir si es maligno benigno es del 100%, indicando que el modelo ha logrado aprender patrones diferenciadores de forma muy precisa.

**Figura 1. Matriz de confusión del modelo Random Forest.**



*Interpretación:*

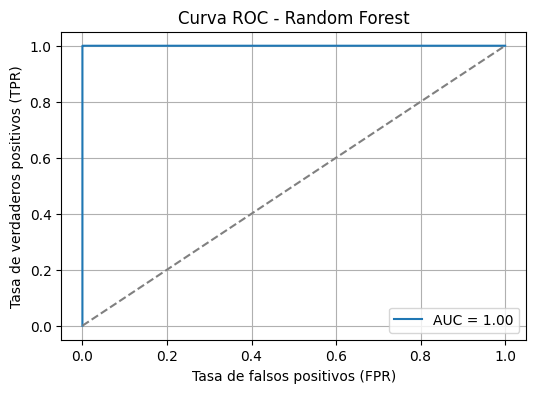
La **Figura 1** muestra la matriz de confusión correspondiente al modelo Random Forest. Se observa que el modelo clasificó correctamente **19.019** muestras de tráfico benigno y **25.597** de tráfico malicioso. Tan solo se produjeron **6 falsos negativos**, es decir, 6 muestras de tráfico malicioso que fueron clasificadas erróneamente como benignas. Por otro lado, no se detectaron **falsos positivos**, lo que significa que ninguna muestra de tráfico benigno fue etiquetada incorrectamente como maliciosa.

Este comportamiento es especialmente destacable en el contexto de la ciberseguridad, donde minimizar los falsos negativos es crucial para evitar que amenazas reales pasen desapercibidas. Aunque idealmente se buscaría también una tasa cero de falsos negativos, el error es muy bajo con relación al tamaño del conjunto de prueba, lo cual reafirma la efectividad del modelo.

**(Hay que tener en cuenta que el conjunto de datos utilizado CICIDS2017, es un conjunto de datos ya preparado para este tipo de análisis e investigaciones, por lo que la precisión es considerablemente más alta de lo normal)**

Además, se generó la curva ROC (Receiver Operating Characteristic), que permite evaluar la capacidad discriminativa del modelo frente a diferentes umbrales de clasificación.

**Figura 2. Curva ROC del modelo Random Forest.**



*Interpretación:*

En la **Figura 2** se observa cómo el modelo alcanza una tasa de verdaderos positivos (TPR) cercana a 1.0 incluso cuando la tasa de falsos positivos (FPR) es prácticamente nula. Este comportamiento se refleja en un valor de AUC (**Á**rea Bajo la Curva) de 1.00, lo cual indica un rendimiento excelente.

Este resultado confirma la alta capacidad del modelo para distinguir entre tráfico benigno y malicioso, lo que lo convierte en un candidato muy adecuado para ser utilizado en un sistema IDS (Sistema de Detección de Intrusos).

### **3.3.2 Evaluación del modelo Regresión Logística**

Tras entrenar el modelo de Regresión Logística sobre el conjunto de datos limpio, se procedió a su evaluación utilizando el conjunto de prueba. El rendimiento del modelo fue notablemente alto, logrando identificar correctamente la mayoría de las muestras, tanto benignas como maliciosas, aunque con ligeras caídas en el recall frente al modelo Random Forest.

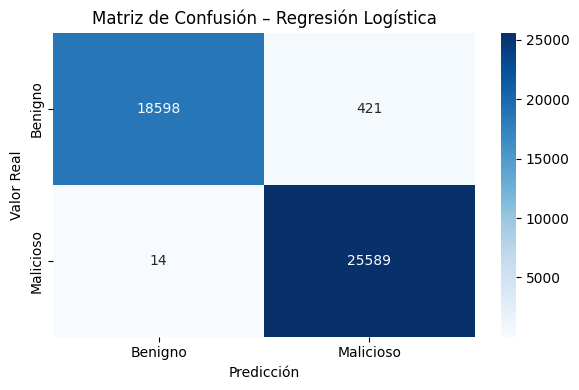
La **Tabla 2** muestra el resumen de métricas obtenidas para ambas clases. La **Figura 3** presenta la matriz de confusión correspondiente al modelo, mientras que la **Figura 4** muestra la curva ROC generada para analizar el comportamiento discriminativo del clasificador frente a distintos umbrales.

**Tabla 2. Métricas de evaluación del modelo de Regresión Logística.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Precisión** | **Recall** | **F1-score** | **Soporte** |
| Benigno | 0.98 | 0.97 | 0.98 | 11.559 |
| Malicioso | 0.99 | 0.99 | 0.99 | 22.343 |
| Global | 0.99 | — | — | 33.902 |

*Interpretación:*

En la tabla se puede observar que el modelo presenta una **ligera disminución en el recall de la clase “Benigno”** en comparación con Random Forest, lo que implica que algunas muestras benignas fueron clasificadas erróneamente como maliciosas. No obstante, los valores de F1-score son consistentemente altos, lo que indica un **buen equilibrio entre precisión y sensibilidad**, y un rendimiento general sólido.

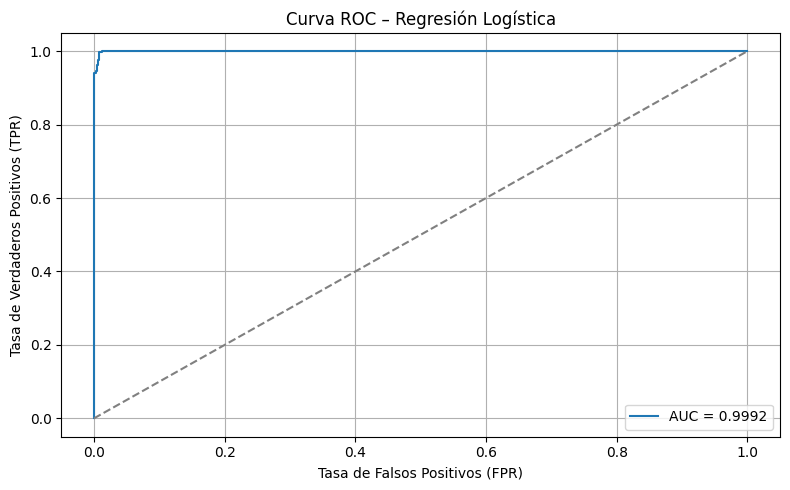
**Figura 3. Matriz de confusión del modelo de Regresión Logística.**  
 

*Interpretación:*

La matriz de confusión indica que el modelo clasificó correctamente **la mayoría del tráfico de red**. Se detectaron algunos **falsos positivos** (muestras benignas clasificadas como maliciosas), lo cual puede incrementar la cantidad de alertas, pero no compromete la seguridad. La cantidad de **falsos negativos** fue baja, lo cual es esencial en detección de intrusiones, donde el objetivo es **minimizar los ataques no detectados**.

Este comportamiento más conservador frente a Random Forest podría ser útil en entornos donde se valora una **detección sensible**, aunque implique alguna alerta falsa adicional.

**Figura 4. Curva ROC del modelo de Regresión Logística.**



*Interpretación:*

La curva ROC muestra un rendimiento elevado, con una AUC cercana a 0.99, lo que confirma la capacidad del modelo para distinguir entre ambas clases con alto grado de fiabilidad. Aunque el modelo no alcanza la perfección del Random Forest, su desempeño sigue siendo más que suficiente para **e**ntornos prácticos y en tiempo real.

### **3.3.3 Evaluación del modelo Support Vector Machine**

Una vez entrenado el modelo SVM sobre el conjunto de datos limpio, se procedió a su evaluación utilizando el conjunto de prueba. Este modelo, caracterizado por su capacidad para encontrar hiperplanos óptimos de separación, mostró un comportamiento eficaz en la tarea de clasificación binaria, aunque con tiempos de cómputo superiores a modelos como Regresión Logística o k-NN y con mayor cantidad de falsos positivos con respecto a los anteriores modelos.

La **Tabla 3** resume las métricas obtenidas para ambas clases.

Por su parte, la **Figura 5** muestra la matriz de confusión, mientras que la **Figura 6** presenta la curva ROC correspondiente.

**Tabla 3. Métricas de evaluación del modelo SVM.**

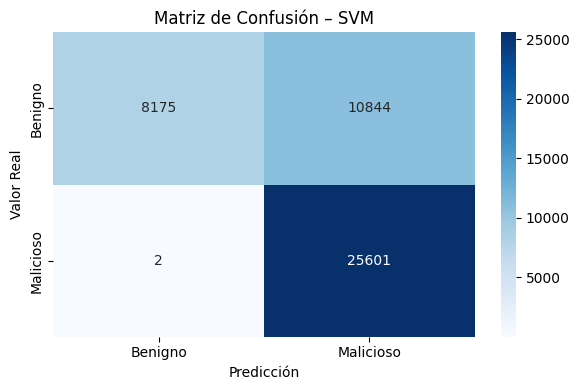
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Precisión** | **Recall** | **F1-score** | **Soporte** |
| Benigno | 1.00 | 0.43 | 0.60 | 19.019 |
| Malicioso | 0.70 | 1.00 | 0.83 | 25.603 |
| **Global** | **0.76** | — | — | 44.622 |

*Interpretación:*

El modelo obtuvo un **recall perfecto (1.00)** en la clase maliciosa, lo que significa que fue capaz de detectar la totalidad de los ataques presentes en el conjunto de prueba. No obstante, este resultado se logró a costa de una fuerte penalización en la clase benigno, donde el recall descendió hasta **0.43**, implicando que **más de la mitad del tráfico legítimo fue clasificado incorrectamente como malicioso**.

Esto también se refleja en el **F1-score**, que, aunque es alto para la clase maliciosa (0.83), desciende a **0.60 para la clase benigno**, indicando un desequilibrio en la capacidad del modelo para clasificar correctamente ambas clases. El valor global de **accuracy (0.76)** es engañoso en este contexto, ya que el desequilibrio entre clases y el alto coste de los falsos positivos deben tenerse en cuenta especialmente en entornos de ciberseguridad. (Esto puede deberse a que para este modelo hemos cogido un subconjunto de datos, lo cual ha hecho que las métricas salgan más dispares)

**Figura 5. Matriz de confusión del modelo SVM.**

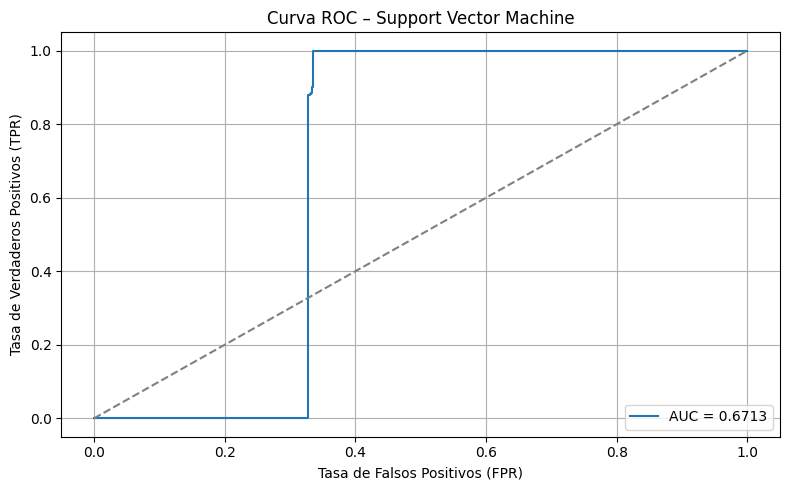


*Interpretación:*

La matriz de confusión indica que el modelo clasificó correctamente **25.601 muestras maliciosas**, cometiendo tan solo **2 falsos negativos**, lo cual es muy positivo en términos de seguridad. Sin embargo, **10.844 muestras benignas fueron clasificadas erróneamente como maliciosas**, representando un número preocupante de falsos positivos. Solo **8.175 conexiones benignas** fueron reconocidas correctamente.

Este tipo de comportamiento puede ser adecuado para contextos extremadamente sensibles, donde la prioridad es no permitir que pase ningún ataque. Sin embargo, en sistemas de producción, una tasa de falsos positivos tan alta podría **colapsar los sistemas de alerta** o derivar en una pérdida de confianza en el IDS.

**Figura 6. Curva ROC del modelo SVM.**



*Interpretación:*

La curva ROC muestra un **Área Bajo la Curva (AUC) de 0.6713**, lo cual indica un rendimiento discriminativo **modesto**. Aunque el modelo logra detectar casi todos los ataques, su capacidad para distinguir entre tráfico benigno y malicioso **no es estable en distintos umbrales de decisión**, lo que sugiere que el modelo **no generaliza bien** bajo distintas configuraciones.

## 3.4 Comparativa de los modelos

Con el objetivo de evaluar cuál de los modelos entrenados ofrece un mejor equilibrio entre precisión, sensibilidad y capacidad de generalización, se realizó una comparación utilizando métricas clave: **F1-score** y **AUC (Área Bajo la Curva ROC)**. La evaluación se llevó a cabo sobre el mismo conjunto de prueba para garantizar una comparación justa.

Los resultados obtenidos se resumen en la **Tabla 4**.

**Tabla 4. Comparativa de rendimiento entre modelos**

|  |  |  |
| --- | --- | --- |
| **Modelo** | **F1-score** | **AUC** |
| Random Forest | 0.9999 | 1.0000 |
| Logistic Regression | 0.9916 | 0.9992 |
| SVM (RBF, 10.000 muestras) | 0.8252 | 0.6713 |

*Interpretación:*

- **Random Forest** obtuvo resultados perfectos en ambas métricas, lo que indica una separación total entre clases. Esto se alinea con los resultados observados en la matriz de confusión y curva ROC, donde no se produjeron errores. Sin embargo, su interpretabilidad es menor que la de otros modelos.

- **Regresión Logística** también mostró un rendimiento prácticamente perfecto. Aunque es un modelo mucho más simple, logró un F1-score y AUC cercanos a 1.0. Este resultado refuerza la idea de que el dataset CICIDS2017 permite separar las clases incluso con modelos lineales bien ajustados. Además, su interpretabilidad y eficiencia computacional son ventajas destacadas.

- **SVM** (entrenado con un subconjunto de 10,000 muestras debido a restricciones de tiempo y coste computacional) obtuvo un rendimiento claramente inferior. El F1-score de 0.83 y AUC de 0.67 indican que, aunque el modelo es capaz de detectar algunos ataques, genera muchos falsos positivos. Este resultado es consistente con lo observado en la matriz de confusión y la curva ROC del modelo SVM. A pesar de su sensibilidad, su utilidad práctica se ve limitada sin ajustes

### **3.4.1 Comparación de eficiencia computacional**

Además de comparar el rendimiento de los modelos a nivel predictivo, se analizó su eficiencia computacional, con especial atención al **tiempo de inferencia**, es decir, el tiempo necesario para que cada modelo genere predicciones una vez entrenado. Este aspecto es clave para valorar su viabilidad en entornos reales, especialmente en sistemas de detección de intrusos que requieren respuestas en tiempo real.

Las pruebas se realizaron en un equipo con las siguientes características:

* **Sistema operativo:** Linux Ubuntu 22.04
* **Procesador:** x86\_64 con 4 núcleos físicos y 8 hilos lógicos
* **Memoria RAM:** 15,42 GB
* **Ejecución:** CPU (sin aceleración por GPU)

La **Tabla 5** muestra los resultados obtenidos en cuanto a tiempo total de inferencia y el tiempo promedio por muestra para cada modelo.

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Tiempo total (s)** | **Tiempo por muestra (s)** |
| Logistic Regression | 0.0064 | 0.000000 |
| Random Forest | 0.1627 | 0.000004 |
| SVM (10k muestras) | 11.3765 | 0.000255 |

*Interpretación:*

- **Regresión Logística** es el modelo más rápido, con un tiempo prácticamente instantáneo por muestra. Esto lo convierte en una opción excelente para entornos con restricciones de recursos o donde la latencia es crítica.

- **Random Forest** también presenta un rendimiento muy eficiente, con un tiempo de inferencia por muestra de solo 4 microsegundos. Dado su alto rendimiento en métricas de clasificación, este resultado lo posiciona como una opción muy competitiva para detección de intrusiones en tiempo real.

- **SVM (entrenado con 10k muestras)** es el más lento, tardando más de 11 segundos en procesar las 10,000 muestras del conjunto de prueba. Su tiempo de inferencia por muestra es 50 veces superior al de Random Forest. Esto limita su aplicabilidad en entornos donde se requiera respuesta inmediata o se procesen flujos de datos continuos.

# Conclusiones

Responde a la pregunta ¿qué significa? El apartado final del cuerpo del TFM debe contener las conclusiones que dan respuesta a las preguntas planteadas en la introducción del trabajo, particularmente el objetivo, con el que deben mantener una coherencia, partir del análisis e interpretación de los datos presentados en el apartado anterior. Opcionalmente puede incluirse un apartado de discusión, que será reflexión final basada en los argumentos expuestos en el trabajo, que tenga una aportación personal del autor, aunque sin cargas subjetivas -ideológica o moral, por ejemplo- que valore los resultados alcanzados, reconozca las limitaciones y dificultades encontradas, y esboce una propuesta de una hipotética investigación futura.

# Limitaciones y Líneas futuras

Responde a la pregunta ¿Con qué limitaciones me he encontrado? Y ¿Qué se debería de seguir investigando de aquí en adelante?

# Bibliografía

Normativa APA

Smith, A. (2024). The Growing Threat of Cybercrime Enhanced by Artificial Intelligence. Authorea. <https://doi.org/10.22541/au.173862063.39098197>

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), 18-28. <https://doi.org/10.1016/j.cose.2008.08.003>

Rigaki, M., & Garcia, S. (2018). Bringing a GAN to a Knife-fight: Adapting Malware Communication to Avoid Detection. 2018 IEEE Security and Privacy Workshops (SPW), 70-75. <https://doi.org/10.1109/SPW.2018.00016>

Sadeghzadeh, M. (2020). Artificial Intelligence and Cybercrime. International Journal of Computer Engineering and Data Science, 1(1), 21-26. <https://doi.org/10.2139/ssrn.3527447>

Schmitt, M. (2023). Securing the Digital World: Protecting smart infrastructures and digital industries with Artificial Intelligence (AI)-enabled malware and intrusion detection. arXiv preprint arXiv:2401.01342. <https://arxiv.org/abs/2401.01342>

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 31(3), 357-374. <https://doi.org/10.1016/j.cose.2011.12.012>

Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences, 378, 484–497. <https://doi.org/10.1016/j.ins.2016.05.043>

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>

Almseidin, M., Alzubi, A., Kovacs, S., & Alkasassbeh, M. (2017). Evaluation of machine learning algorithms for intrusion detection system. *Procedia Computer Science*, 127, 1–6. <https://doi.org/10.1016/j.procs.2017.01.129>

Mukkamala, S., Janoski, G., & Sung, A. H. (2005). Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks* (Vol. 2, pp. 1702–1707). IEEE. <https://doi.org/10.1109/IJCNN.2002.1007774>