Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Negocios y Administración Pública

CARRERA DE ESPECIALIZACIÓN EN MÉTODOS CUANTITATIVOS PARA LA GESTIÓN Y ANÁLISIS DE DATOS EN ORGANIZACIONES

TRABAJO FINAL INTEGRADOR

Análisis comparativo de técnicas de Machine Learning para la detección y clasificación de sitios web maliciosos

AUTOR: LUCIANO MARTÍN HAINZE

TUTOR: NÉLIDA MÓNICA CANTONI RABOLINI

[ABRIL 2024]

Resumen

El presente trabajo consiste en el planteo y desarrollo de un proyecto de Ciencia de Datos para la clasificación binaria y multiclase de sitios web benignos y maliciosos. La investigación consiste en la comparación de la performance de los modelos de aprendizaje automático supervisado Random Forest, Adaptive Boosting Decision Trees, Logistic Regression, Neural Net y Naive Bayes evaluando las métricas ROC-AUC, Precision, Recall y F1-Score. La clasificación es realizada primeramente entre sitios benignos y maliciosos, y en segundo lugar distinguiendo entre las clases de sitios maliciosos malware, spam, phishing y defacement. Se utilizan características del léxico empleado y datos provenientes de la base WHOIS para el entrenamiento de los modelos.

El modelo Random Forest alcanzó una mejor performance que los modelos restantes para la clasificación binaria y multiclase, alcanzando un ROC AUC de 0,998 y 0,999 respectivamente. El trabajo presenta una optimización del modelo seleccionado y una descripción de su funcionamiento. El trabajo finaliza con la postulación de una propuesta de arquitectura de datos para el despliegue del proyecto.

Palabras clave: ciberseguridad, ciencia de datos, machine learning.

Índice

Introducción

El adecuado desarrollo de las operaciones de las organizaciones depende del funcionamiento eficiente de sistemas informáticos y de red, que interconectan a los individuos involucrados en la consecución de sus objetivos. La ciberseguridad tiene como objetivo el diseño de prácticas, políticas y tecnologías para proteger a los sistemas informáticos y de red de riesgos y amenazas provenientes de la exposición a Internet y el uso de tecnologías digitales.

Entre las principales amenazas que enfrentan las organizaciones al conectarse a Internet, la más frecuente es la exposición a sitios webs maliciosos. Según el informe publicado por Kaspersky (2022), el 43,68% de los programas maliciosos utilizados para realizar ataques al usuario tuvieron su origen en la exposición del usuario a sitios webs maliciosos. Prevenir, detectar y responder adecuadamente a este tipo de ataques informáticos es de vital importancia, ya que la irrupción de software malicioso a la red a partir de una sola computadora puede llegar a contaminar su conjunto y como consecuencia ocasionar un daño en la integridad y privacidad de la información.

La Ciencia de Datos proporciona un conjunto de herramientas fundamentales para la detección temprana, de la exposición de los usuarios a sitios webs potencialmente maliciosos, permitiendo así la adopción de medidas para salvaguardar la integridad de las redes. Este proceso se lleva a cabo mediante la implementación de aplicaciones que incluyen en sus procesos modelos de machine learning para analizar las características de los sitios web visitados por los usuarios y clasificarlos entre benignos y maliciosos, identificando el tipo de riesgo que representan.

La problemática de la detección e identificación oportuna de sitios web maliciosos, a partir de técnicas de machine learning, ha tenido un importante desarrollo a través de diversas variantes en cuanto al tipo de solución planteada. A partir de la técnica empleada para la detección, las soluciones existentes pueden categorizarse en variantes estáticas, dinámicas e híbridas.

Por otro lado, un enfoque estático busca detectar sitios webs maliciosos analizando el contenido de su URL o sus contenidos. A partir del análisis del contenido de su URL, de las sentencias de HTML y contenidos web que posee, así como datos vinculados a fuentes como el WHOIS, este método anticipa y predice si el usuario intenta acceder a un sitio web potencialmente malicioso.

Un enfoque dinámico, en cambio, busca detectarlos analizando su comportamiento en tiempo real mediante Honeypots, sistemas informáticos o una aplicación que se configura de manera

intencional para parecer vulnerable o atractivo para los atacantes. Cuando un atacante interactúa con un Honeypot, este registra su actividad y permite conocer las técnicas utilizadas por los atacantes durante la navegación del usuario impidiendo que puedan ocasionar un daño a su dispositivo y a la red.

Así mismo, el enfoque híbrido integra ambos métodos fortaleciendo su detección de sitios webs maliciosos integrando estas herramientas.

Estos métodos empleados para la detección de sitios webs maliciosos enfrentan la necesidad de ser constantemente revisados. Para lograr sus propósitos, sitios webs maliciosos son creados continuamente y las técnicas para vulnerar al usuario son mejoradas para evitar su detección. Por lo tanto, en la búsqueda de una detección eficiente, es necesario poseer un conjunto de datos que incluya registros de sitios webs activos donde el análisis de los modelos se realice sobre sitios que un usuario pueda encontrarse al navegar por la web. Además, es necesario elaborar atributos que proporcionen información útil a los modelos y contribuyan a la identificación efectiva de los sitios webs maliciosos. Otro desafío es la selección adecuada del modelo de aprendizaje automático que satisfaga en mayor medida las necesidades del usuario.

Ante esta problemática, la propuesta del presente trabajo consiste en la comparación del rendimiento de los modelos de machine learning: Random Forest, Logistic Regression, Adaptive Boosting Decision Trees, Neural Net y Naive Bayes, buscando la optimización de la clasificación, primeramente, entre sitios web benignos y maliciosos, y en segundo lugar entre los sitios web maliciosos, la adecuada clasificación de los mismos entre las clases de ataque que representan, distinguiendo entre Phishing, Malware, Defacement y Spam, mismas clases utilizadas por Mamun et al. (2016).

El objetivo primario es la comparación de los resultados proporcionados por los modelos Random Forest, Logistic Regression, Adaptive Boosting Decision Trees, Neural Net y Naive Bayes para la distinción entre sitios web benignos y malignos, clasificando también estos últimos en la clase de ataque que representan.

Para la consecución de este propósito, el primer objetivo que es planteado será la elaboración de una propuesta metodológica basada en el estado del arte y los recursos disponibles. Con esta finalidad se describirán los conceptos mínimos necesarios para la comprensión de esta problemática, los antecedentes en su abordaje y los aportes más relevantes de soluciones

existentes planteadas, elaborando en base a estos una propuesta metodológica adecuada. En segundo lugar, se buscará desarrollar la propuesta metodológica, desde la obtención de la base de datos, creación de atributos, planteo de una estrategia de modelización, tuneo de hiperparámetros y generación de resultados siguiendo la propuesta metodológica planteada. Por último, se propone seleccionar un modelo basado en los resultados obtenidos y su optimización, explicar el funcionamiento del modelo a partir de la visualización de los resultados más relevantes, finalizando con una elaboración de una propuesta de despliegue del modelo escogido.

El presente trabajo consta de tres apartados. En el primero, se presentará el análisis del problema, un estado del arte de trabajos preliminares y la descripción de la propuesta metodológica. El segundo apartado constará de la obtención de los datos, creación de atributos, preprocesamiento de los datos, aplicación de modelos de machine learning y optimización de hiperparámetros, y la generación de resultados. El tercer apartado consiste en la selección del modelo y optimización del modelo, la explicación de su funcionamiento y la elaboración de una propuesta para el despliegue del modelo. El trabajo concluye con las contribuciones realizadas por el artículo, sus limitaciones y el trabajo futuro que puede realizarse a partir del mismo.

Contexto del problema y estado del arte

Es este primer apartado se abordará conceptualmente la problemática de los sitios webs maliciosos y su impacto en los usuarios. Seguidamente se desarrollarán los principales antecedentes en cuanto a la aplicación de técnicas de aprendizaje automático para la identificación de sitios webs maliciosos. Por último, finalizando esta sección, se planteará de manera teórica la metodología a implementar para la resolución del problema.

Métodos actuales de detección de sitios webs maliciosos.

Para comprender el problema, es necesario exponer conceptos que serán empleados durante el desarrollo de la presente investigación. En primer lugar, se identificará el problema desde la Ciencia de Datos. Luego, se expondrá qué es considerado un URL malicioso. Posteriormente, entendiendo que serán reunidos sitios web maliciosos de spam, defacement, malware y phishing, será analizado los conceptos de cada uno de ellos para obtener un entendimiento de las características de cada uno de estos ataques. Seguidamente se expondrán las diversas fuentes de las cuales se obtienen los atributos de los sitios web para realizar la clasificación de los URL.

Según la Universidad de Buenos Aires (2023), el Machine Learning no sólo buscar resolver un problema de negocio utilizando datos, sino que además busca que su performance mejore con la experiencia. Esto implica que la práctica continua, la repetición del método y la iteración permiten un incremento en la eficacia con la que se resuelve un problema. La formulación de un problema de Ciencia de Datos incluye la existencia de tres parámetros: una tarea, una experiencia y una performance. Una tarea consiste en el método a implementar, el modelo de aprendizaje automático a utilizar. La experiencia es el conjunto de datos objeto de análisis en el problema de negocio. La performance es la métrica de rendimiento a partir de la cual se evaluará el grado de eficacia de respuesta al problema planteado.

En primer lugar, se procederá a describir las características de la tarea en la Ciencia de Datos. Un proyecto de Ciencia de Datos es un ciclo de aprendizaje en torno a los datos. El inicio del proyecto viene dado por los objetivos de la organización. A continuación, se recogen, se preparan y gestionan los datos. Como siguiente paso, se desarrollan y evalúan algoritmos matemáticos sobre los datos, es decir, un modelo es construido, evaluado y criticado. Los resultados de estos modelos se presentan a los expertos en el dominio de la aplicación utilizando técnicas de visualización y presentación para su posterior integración dentro de la organización donde es desplegada la solución al problema.

La Ciencia de Datos distingue entre técnicas de aprendizaje supervisado y no supervisado. La diferencia entre ambas radica en la existencia o no de una variable o etiqueta a predecir. Cuando existe una etiqueta a predecir nos encontramos ante una técnica de aprendizaje supervisado; cuando no existe una variable o atributo a predecir, nos encontramos ante una técnica de aprendizaje no supervisado. Las técnicas de aprendizaje supervisado se dividen entre técnicas de clasificación o regresión según la variable a predecir sea cualitativa o cuantitativa respectivamente. Un modelo de clasificación consiste en una representación abstracta de una relación entre un conjunto de atributos y una etiqueta de clase (Pang-Ning et al., 2018). En contraste, un modelo de regresión es una herramienta estadística utilizada para representar la relación entre un conjunto de datos y una variable dependiente con el fin de predecir valores numéricos. Por otro lado, una técnica de aprendizaje no supervisado involucra no predecir una variable específica, sino buscar similitudes y diferencias entre los registros analizados, permitiendo la conformación de grupos (llamados clusters) o la identificación de anomalías. Dado que en este caso se busca clasificar los registros de sitios web en benignos y malignos y

luego en la clase de ataque que representa, se considera que el problema a resolver requerirá la implementación de técnicas de aprendizaje supervisado de clasificación.

La experiencia, el conjunto primario de los datos a evaluar consiste en un conjunto de URL. URL es la abreviatura de Uniform Resources Locator, en español, Localizador Uniforme de Recursos, y consiste en el nombre completo de un recurso en internet. Un URL contiene tres partes: el protocolo, el hostname y el path. El protocolo es la parte inicial de un URL que identifica el protocolo empleado para acceder al recurso deseado en Internet. Este establece reglas para que la comunicación entre el usuario y el servidor sea exitosa. Entre los protocolos más conocidos se encuentran el HTTP y el HTTPS. El hostname es una denominación que permite identificar de manera única al servidor que se desea acceder, así como la dirección de este servidor. El path, por último, es la sección de un URL que sigue al hostname e indica la ruta dentro de la estructura del servidor para acceder al contenido específico que el usuario busca.

Un URL se considera malicioso o benigno según su contenido descargable de los servidores o los efectos que puedan ocurrir cuando un usuario accede al sitio web. El contenido malicioso de un sitio web puede ser una aplicación o un contenido web-based ejecutado en el sistema del usuario. En el primer caso, el URL es usado pasivamente, usualmente por otra aplicación maliciosa o mediante métodos de ingeniería social engañando al usuario para que ejecute la aplicación alojada. En el segundo caso, el contenido web-based es un programa que interactúa con el navegador. La interacción con este contenido malicioso infecta al sistema del usuario (Popescu et al., 2015).

La bibliografía actual clasifica los sitios web maliciosos mayormente en cuatro categorías, a saber, las siguientes:

- Defacement: conocida como desfiguración, implica el aprovechamiento de vulnerabilidades de los sitios web para alterar el contenido o su apariencia sin autorización, reemplazándolo por mensajes e imágenes indeseables u ofensivas.

- Spam: sitios web que reciben esta clasificación están relacionados a correos electrónicos no deseados o mensajes masivos no solicitados, pudiendo contener formularios de registro o suscripción falsos que recopilan direcciones de correo electrónico para su uso

posterior en campañas de correo no deseado o promover productos o servicios fraudulentos o engañosos.

- Malware: consiste en sitios web que se emplean para distribuir software maligno, que al interactuar con el usuario, permite la entrada de aplicaciones que infectan el dispositivo, causando daño y pudiendo replicarse en la red a la que pertenece el dispositivo infectado. (Martín de Diego & Fernández Isabel, 2020)

- Phishing: los sitios web maliciosos caracterizados como phishing tienen la finalidad de engañar a los usuarios y robar información confidencial, como contraseñas, cuentas bancarias, datos de tarjetas de crédito o información personal. Los atacantes suelen diseñar sitios web que simulan ser páginas conocidas por el usuario para que este interprete que ha ingresado a un sitio web seguro y comparta información confidencial que luego pueda ser empleada para suplantar la identidad del usuario y ocasionarle pérdidas financieras.

Habiendo identificado el problema desde la perspectiva de la Ciencia de Datos y conceptualizado las clasificaciones que los sitios web puede, es necesario conocer qué datos serán empleados para el aprendizaje de los modelos, en qué consisten los atributos a utilizar para llevar a cabo la predicción. Las variables para observar de los sitios web en el presente trabajo serán las siguientes:

- Atributos del léxico del URL: atributos vinculados a las propiedades del URL como ser su longitud total, la longitud del nombre del host, la cantidad de puntos, la presencia de caracteres especiales, dominios y símbolos sospechosos o redirecciones.

- Datos WhoIs: La fecha de registro, actualización y expiración del sitio web, el nombre del servidor y donde ha sido registrado, status, DNSSEC, entre otros.

La performance es el último elemento que resta por describir para culminar con la formulación del problema de Ciencia de Datos. Las métricas nos ofrecen la posibilidad de capturar una cualidad determinada del modelo. Las métricas serán las medidas a optimizar del modelo que indicarán el grado de eficacia para la resolución del problema. Las métricas de clasificación a emplear en la presente investigación son los siguientes:

- Accuracy: Relación entre el número de verdaderos positivos y el número total de predicciones. Mide la frecuencia con la que el clasificador hace la predicción correcta.

- Precision: Relación entre el número de verdaderos positivos y clasificados como positivos.

- Recall: Relación entre los verdaderos positivos y verdaderos positivos y falsos negativos.

- ROC-AUC: Métrica que mide cuántas clasificaciones positivas correctas se pueden obtener a medida que se admiten más falsos positivos.

- F1-Score: Consiste en la media harmónica entre las medidas de Precision y Recall.

Antecedentes en la aplicación de machine learning para la detección de sitios web maliciosos

El análisis de URLs y la identificación de sitios web maliciosos son temas abordados por la literatura actual. Se han desarrollado diversas técnicas y enfoques para mejorar la precisión y eficiencia en la detección de sitios web maliciosos y la optimización de su clasificación. A continuación, se presentarán trabajos relevantes en este campo que conforman un estado actual del conocimiento:

Una de las primeras contribuciones realizadas orientadas al problema de clasificación de contenidos web maliciosos es el artículo escrito por Abu-Nimeh et al. (2007) quienes llevaron a cabo un estudio comparativo de los modelos Regresión Logística, Árboles de Decisión y Clasificación, Árbol aditivo de decisión Bayesiano, Random Forest y Redes Neuronales clasificando un dataset conformado por 2889 mails a partir del cual fueron generados 43 atributos. El rendimiento de los modelos fue analizado a través de su precisión, la tasa de falsos positivos y falsos negativos.

Seguidamente, Ma et al. (2009) realizaron un estudio de detección de sitios web maliciosos a partir del análisis de datos obtenidos en tiempo real de la conformación del URL y de información del host. Fue generada una base de datos con 1,791,261 atributos de la conformación del URL y 1,117,901 atributos de información sobre el host, tratándose de uno de los trabajos más importantes en relación con esta problemática con una metodología estática

que presenta gran escalabilidad. Fueron empleados distintos modelos, los modelos de Perceptrón, Regresión Logística, Algoritmo Pasivo-Agresivo y el Algoritmo CW. Este último fue el modelo que llevó a la predicción a una precisión del 99%. Ma et al. (2009) llevan a cabo otro estudio de las aplicaciones del machine learning para la predicción de sitios webs maliciosos alcanzando una precisión entre los valores de 95% y 99%. El estudio fue realizado comparando el funcionamiento de los modelos SVM, Naive Bayes y Logistic Regression para una base conformada por atributos del contenido del URL y el host en 35,500 provenientes de las bases de datos de Yahoo, DMOZ, PhishTank y Spamscatter.

Y. Li, Z. Yang, X. Chen et al. (2018) realizaron un estudio sobre la implementación de un modelo de Stacking de múltiples capas que integraba los modelos GBDT, XGBoost y LightGBM en tres capas aplicado a una base de datos conformada por características obtenidas del URL y contenido en HTTP de los sitios web. La generación de características permitió elaborar una base de datos conformada por 238 atributos. La propuesta realizada alcanzó la tasa de Accuracy de 97,3%.

McGahagan, J., Bhansali, D., Pinto-Coelho, C., & Cukier, M. (2019) realizaron una evaluación de atributos de contenido de sitios web donde buscaron demostrar el potencial de estos atributos para detectar sitios webs maliciosos. Fueron generados 1,865 atributos que fueron reducidos habiendo identificado los más significativos a 26, generando 17 nuevos atributos. Estos atributos fueron evaluados a partir de técnicas de submuestro, sobremuestro y sin muestrear para verificar el efecto del desbalanceo en la base de datos. Así mismo, se realizaron transformaciones de atributos y reconstrucción de modelos comparando los resultados obtenidos. El modelo que obtuvo las mejores métricas fue el Random Forest de manera generalizada en la mayoría de las pruebas realizadas.

Gressel, G., Ashok, A., Poornachandran, P., Darling, M., & Heileman, G. (2015), por otro lado, generaron un enfoque léxico para la clasificación de sitios web maliciosos buscando desarrollar un programa que no sólo permita una clasificación eficaz en cuanto a su rendimiento en la métrica Accuracy sino buscando obtener un eficiente tiempo de respuesta. Fue alcanzado una accuracy del 99.1% y un tiempo de respuesta 0.627 segundos a partir del uso del modelo J48.

Sun, B., Akiyama, M., Yagi, T., Hatada, M., & Mori, T. (2016) realizaron un trabajo donde se enfocaron en un método dinámico para la detección de sitios web que contengan malware,

desarrollando un sistema llamado AutoBLG capaz de descubrir URLs maliciosos eficientemente.

El trabajo realizado por Xu, L., Zhan, Z., Xu, S., & Ye, K. en 2013 consistió en un análisis híbrido a una base de datos conformada por 124 atributos de clasificación que integraba las capas de aplicación y de red, donde fueron comparados los modelos SVM, J48, Naive Bayes y Logistic Regression. Fue realizado un análisis comparativo de los resultados obtenidos por la aplicación de los distintos modelos seleccionados y ante distintas transformaciones de la base de datos aplicando PCA, Subset o Information Gain. Los resultados fueron medidos utilizando las métricas Accuracy, FNR y FPR. La metodología implementada comparaba los resultados obtenidos empleando análisis de data-aggregation, OR-aggregation, AND-aggregation y XOR-aggregation. La aplicación del análisis XOR-aggregation, sin realizar transformaciones a la base de datos e implementando el modelo J48 demostró mejores rendimientos en cuanto a las métricas escogidas.

Mohaisen, A. (2015) lleva a cabo un método híbrido, que incluye atributos estáticos y dinámicos, para la clasificación de sitios web maliciosos en dos etapas clasificando en por un lado los sitios entre maliciosos y benignos y posteriormente llevando a cabo una clasificación en cuanto a la clase de vulnerabilidad presentada en el sitio web y sobre cada archivo transferido. Para la clasificación es empleado el modelo SVM. En cuanto a la precisión alcanzada, se alcanzó un 96% de precisión mientras que a nivel de archivo de transferencia fue obtenido un 91% de precisión.

Deng, W., Peng, Y., Yang, F., & Song, J llevan a cabo en 2019 un análisis híbrido de una base de datos conformada por 46 atributos clasificados en atributos de contenido web, de clase de script y de atributos del URL. Los autores realizan una selección de atributos a partir del método de Information Gain. Los modelos empleados fueron SVM, KNN, C4.5 y Naive Bayes. Las métricas para maximizar fueron Accuracy, AUC y TPR y FPR. Para las características de contenido web el modelo SVM demostró predecir con mayor eficacia. En cuanto a las características de clases de script, el modelo más apropiado fue KNN, mientras que en los atributos del URL el algoritmo C4.5 fue aquel que maximizó los valores de las métricas analizadas.

Planteo metodológico

Teniendo en cuenta los antecedentes mencionados que conforman el estado del arte y los conceptos anteriormente descriptos que conforman los elementos que constituyen la problemática a abordar, la presente investigación se propone brindar una solución que puede proporcionar la Ciencia de Datos a la detección e identificación de sitios web maliciosos. A fin de generar una respuesta será puesto en marcha la propuesta siguiendo con las etapas del ciclo de vida de un proyecto de Ciencia de Datos.

Habiendo descripto la problemática planteada, la etapa siguiente consiste en la ingesta de datos de URLs las cuales serán objeto de análisis y a partir de las cuales se producirá la creación de nuevos atributos. Los URLs serán obtenidos de los datasets disponibles en el sitio web de Kaggle. Seguidamente, se procederá a la creación de nuevos atributos. A partir del contenido del léxico empleado en los URL serán obtenidas características que incluirán la cantidad de caracteres especiales que posee, la longitud de los componentes del URL y relaciones que puedan obtenerse entre los caracteres del URL. La base de datos WHOIS será empleada para extraer de la base información entre las cuales se destaca Dominio, Registrar, Status, DNSSEC, nombre del Servidor, fecha de creación, última fecha de actualización y fecha de expiración de los URLs.

Tras la ingesta de datos se procede a realizar el preprocesamiento de ellos identificando aquellos registros con datos faltantes procediendo a eliminar dichos registros. Así mismo, los datos presentados de manera cualitativos son transformados en cuantitativos con la finalidad de que puedan ser procesados por los modelos de aprendizaje automático.

Concluido el preprocesamiento de los datos, es elaborada la estrategia de modelado, definiendo los modelos a comparar, los hiperparámetros a optimizar para cada uno de los modelos y las métricas a optimizar con la que será evaluada la performance del modelo. Definidos estos elementos, son utilizados los modelos para el entrenamiento, optimización y prueba. Los modelos son implementados para clasificar los sitios webs en malignos y benignos y luego se procede a entrenar y probar la predicción de sitios web de cada una de las clases de sitio web malicioso analizado contra el resto de los sitios web maliciosos y aquellos que son benignos.

En base a las métricas seleccionadas para comparar el rendimiento de los modelos, se selecciona el modelo que ofrezca los mejores resultados según las métricas seleccionadas. Para contribuir

a la explicación del modelo, se desarrollan visualizaciones que generen insights sobre la performance del modelo y que describen su funcionamiento. Estas visualizaciones serán realizadas mediante el uso de la herramienta Power BI.

Por último, es llevado a cabo una propuesta para el despliegue del modelo elegido definiendo una arquitectura a partir de la cual se postulará la implementación del modelo utilizado.

Desarrollo de la propuesta metodológica

Obtención de datos y creación de atributos

Para desarrollar del modelo, se emplean dos conjuntos de datos que se encuentran disponibles en la plataforma Kaggle. Los conjuntos de datos utilizados son Malicious URLs dataset y Spam URLs Classification Dataset. Estos conjuntos se combinan para formar un nuevo conjunto de datos que incluye el URL y su tipo, categorizado como benign, malware, phishing, spam o defacement. La distribución de estos datos se presenta a continuación:

Tabla 1

Distribución inicial de sitios web por clase

Type	**Cantidad de URLs por Type URL**
Benign	472,067
Malware	52,985
Phishing	149,407
Spam	75,446
Defacement	211,954
Total de registros	961,870

La columna type, que representa la etiqueta a precedir, tiene una distribución equilibrada, entre Benign y Malicious, con Benign constituyendo el 49.08% del total. Sin embargo, las clases de

URL maliciosos tienen una participación desbalanceada con Malware, Phishing, Spam y Defacement representando el 5,5%, 15,53%, 7,84% y 22.03%, respectivamente.

Constituido el conjunto de datos con la clase a la que pertenece y el URL, se procede a obtener atributos relevantes. En primer lugar, se obtienen los datos del protocolo WHOIS. A partir de los URL, se consulta la siguiente información:

1. Nombre de Dominio: identificación alfanumérica del sitio web, utilizada para traducir una dirección IP.
2. Registrar: es la entidad acreditada que brinda servicios de registro de dominios.
3. Fecha de creación: fecha en que fue registrado por primera vez el sitio web.
4. Fecha de expiración: fecha en la que expira la registración actual del dominio.
5. Fecha de actualización: fecha que indica cuando se realizaron las últimas modificaciones en la información asociada con el dominio.
6. Nombre del servidor: servidor DNS que almacenan la información sobre la asociación entre nombre de dominio y direcciones IP.
7. Status: condición actual del dominio en el contexto del registro.
8. DNSSEC: El DNSSEC es una extensión de seguridad para el sistema DNS. Indica si el dominio está configurado para utilizar o no DNSSEC.

La búsqueda de esta información generó un error en una cantidad considerable de registros, de manera tal que, como tratamiento de los valores faltantes, a fin de posibilitar el funcionamiento correcto de los modelos de aprendizaje automático a comparar, se procede a la eliminación de los registros con valores faltantes.

Tabla 2

Distribución de valores faltantes por atributo de WHOIS

Atributo	**Valores faltantes**
Domain_Names	278,281
Registrars	329,251

Fecha de creación	323,934
Fecha de expiración	333,483
Fecha de actualización	319,853
Name Servers	305,457
Status	303,491
DNSSEC	384,990

Realizado este tratamiento de los valores faltantes, la distribución de URL según la variable type es modificada adquiriendo la siguiente participación relativa:

Tabla 3

Distribución posterior a eliminación de datos faltantes de sitios web por clase

Type	**Cantidad de URLs por Type URL**	**Participación relativa**
Benign	403,168	71,72%
Malware	14,563	2,59%
Phishing	33,738	6%
Spam	51,160	9,10%
Defacement	59,534	10,59%
Total de registros	562,163	100%

Los atributos Nombre del Dominio, Registrar, Nombre del servidor, Status y DNSSEC constituye constituyen variables categóricas. Siendo que las respuestas únicas de cada una de estas variables son significativas, ver cuadro debajo, se procede a la obtención de las variables dummies para las respuestas de mayor participación relativa eliminando las variables categóricas.

Tabla 4

Cantidad de respuestas únicas por atributo de WHOIS

Atributo	Cantidad de respuestas únicas
Domain_Names	51,736
Registrars	1,393
Name Servers	27,957

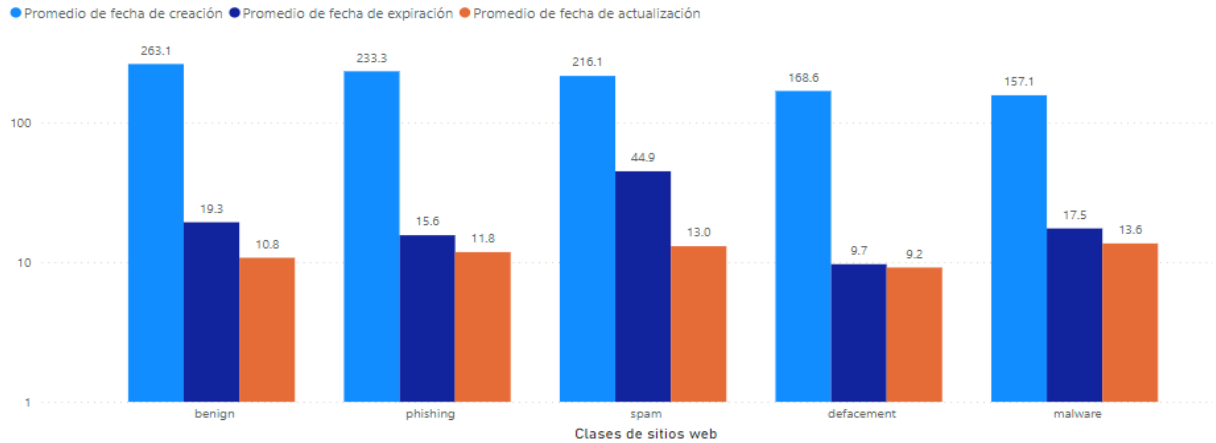
Status	410
DNSSEC	25

Para el tratamiento de las fechas, se calcula la diferencia entre la fecha de creación, de expiración y de última actualización y la fecha del 1 de febrero de 2024 expresada en meses.

En la distribución del conjunto de datos se observan diferencias en el promedio de la antigüedad de la última fecha de actualización y la fecha de creación, mientras que también existe una diferencia entre el período de tiempo en días que falta para que se cumpla la fecha de expiración del sitio web. Al analizar la variable de fecha de expiración, se puede observar cómo el promedio de la fecha de expiración de los sitios de spam, siendo el valor promedio en días 44,26, es visiblemente mayor que las fechas de expiración de las otras clases de ataque, siendo los valores promedios de días 17.05, 15.19 y 9.23 para malware, phishing y defacement. La fecha de la última actualización para los sitios de Defacement es en promedio mucho más cercana a la fecha que las otras clases. Las fechas de creación de sitios web de defacement y malware son más cercanas a la fecha de realización del trabajo (168.12 y 156.57) que los sitios de spam y phishing (215.52 y 232.8). En cuanto a los sitios benignos la antigüedad promedio de su última actualización es de 10.8 días, mientras que la antigüedad promedio de la fecha de creación es de 263.1 días y la diferencia promedio entre la fecha de expiración y la fecha de medición es de 19.3 días.

Ilustración 1

Promedio de fechas antigüedad de fechas de actualización, creación y diferencia entre expiración y fecha de medición.



Luego de tratar con los atributos obtenidos de la consulta a la base del WHOIS se procede a la obtención de los atributos provenientes del léxico de los URL. Aquí se obtendrán características basadas en aspectos de la estructura, contenido, longitud, presencia de ciertos caracteres, utilización de palabras claves, entre otros. A continuación, se presentan los atributos léxicos más relevantes:

1. Longitud del URL: la cantidad de caracteres que posee una dirección de sitio web es un atributo que puede incidir en la determinación si un URL es malicioso o no. Se considera que una longitud inusual de un URL, donde la longitud es extensa, puede asociarse a un sitio web malicioso, mientras que también un sitio cuya dirección es inusualmente corto buscará engañar a los usuarios. En el conjunto de datos analizados se observa que la media del atributo Tamano_URL para los registros benign es de 53.78 mientras que la media para registros malign 62.62. Entre los registros categorizados como malign, defacement 81.18, malware 72.27, phishing 43.88 y spam 50.63.

2. Contiene caracteres especiales: la utilización de caracteres especiales es un factor importante en el que se diferencian los sitios web maliciosos de los benignos. La cantidad media de caracteres especiales para los registros benign es de 8.01, mientras que los datos malign tienen un promedio de 10.57, siendo el promedio para os registros defacement 13.87, para malware 13.18, para phishing 6.47 y para spam 8.69.

3. Contiene subdominio: la presencia de subdominios incide en la posibilidad de que un sitio pueda ser considerado malicioso. La cantidad de subdominios puede relacionarse con los intentos de atacantes de engañar a usuarios utilizando subdominios conocidos a fin de provocar el descuido en su visita. La cantidad promedio de subdominos de los registros considerados benign fue de 0.079, mientras que los malign fue de 0.835.

4. Contiene barra doble: el uso de barras doble puede ser un intento de atacantes de redirección fraudulenta, puede ser un intento de confundir a los usuarios para ocultar la verdadera dirección a la que se encuentran accediendo. Para el conjunto de datos benign sólo el 7.91% contiene doble barra, mientras que los datos con la etiqueta malign poseen en un 83.49% doble barra.

5. Cantidad de “dos puntos”: En una URL, los dos puntos a menudo se utilizan para separar la parte principal de la URL de los parámetros y consultas. Un aumento inusual en la

cantidad de dos puntos podría indicar intentos de manipular o inyectar parámetros de manera maliciosa. La media de cantidad del carácter ":" es para el conjunto de datos benign 0.089, mientras que para los registros malign 0.923. Los sitios web maliciosos poseen un promedio de ":" de 1.183 para defacement, de 0.959 para malware, de 0.284 para phishing, de 1.031 para los registros spam.

6. Cantidad "puntos": La cantidad de puntos en una URL puede indicar el número de niveles de subdominios. Un aumento inusual en la cantidad de puntos puede sugerir una estructura de subdominios compleja, y algunos subdominios maliciosos podrían estar diseñados para imitar la estructura de dominios legítimos. Los datos con la etiqueta benign tienen un promedio de 1.667 puntos, mientras que los malign poseen una media de 2.335, siendo el promedio de defacement 2.522, malware 2.519, phishing 2.381 y spam 2.036.

7. Cantidad de "guiones": El promedio de guiones presentes en los URL por clase de ataques varia. En la clase de phishing existe un promedio de guiones de 0.38, mientras que el promedio de guiones en las clases de malware es mayor, su valor consiste en 0.79. En cuanto a las clases de defacement y spam el promedio de guiones es mayor que las anteriores donde su valor asciende a 1.61 y 1.53. Los sitios benignos tienen un promedio de guiones de 1.9.

8. Cantidad de símbolos de "porcentaje": El promedio del símbolo de porcentaje para los ataques de malware donde su valor para esta clase es 1.04. En contraste la clase de defacement, spam y phishing tiene un promedio del símbolo de porcentaje de 0.08, 0.09 y 0.09 respectivamente. El promedio para los sitios benignos es de 0.52 para la presencia del símbolo de porcentaje.

9. Cantidad de barras: El promedio de la cantidad de barras para phishing es inferior al resto de las clases, existiendo un promedio de 2.65. El promedio de las clases de defacement y spam del símbolo de barras es de 3.85 y 3.76 respectivamente. La cantidad promedio de barras es superior al resto de las clases, alcanzando un valor de 4.76. El promedio de barras para los sitios benignos es de 2.64.

10. Cantidad de signos de interrogación: La cantidad de signos de interrogación para la clase de spam es 0 en promedio, mientras que el promedio de la cantidad de signos de

interrogación para phishing, malware y defacement es 0.12, 0.37 y 0.53 respectivamente. En cambio, los sitios web benignos tienen un promedio de 0.16 de signos de interrogación.

Ilustración 2

Promedio del tamaño del URL por Clase de sitio web

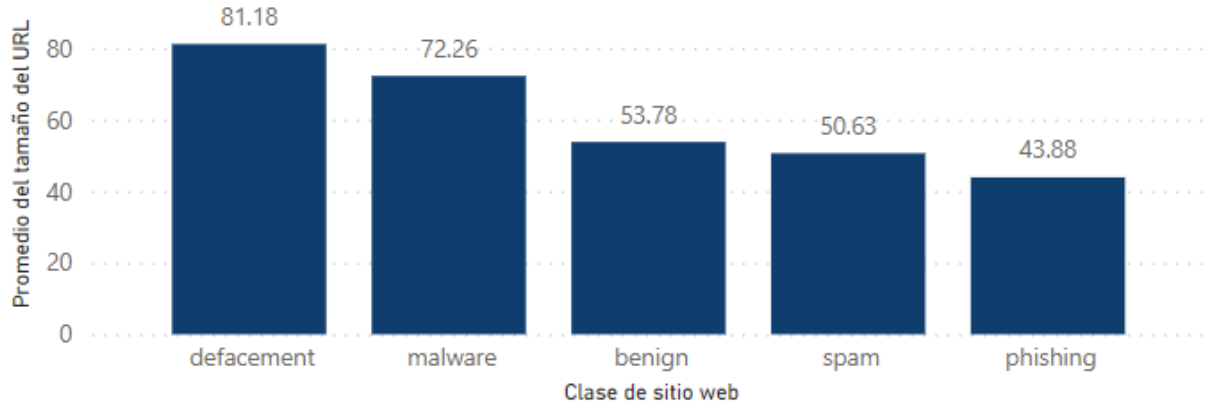
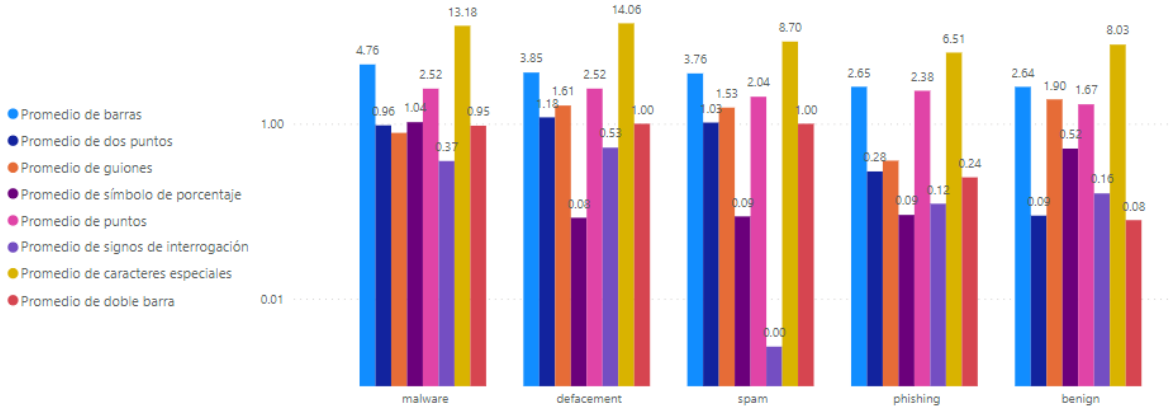


Ilustración 3

Promedio de presencia de símbolos del léxico por clase de URL



Habiendo culminado la creación de atributos y su preprocesamiento, el conjunto de datos se encuentra preparado para ser utilizado durante la etapa del modelado en el ciclo de vida del proyecto de ciencia de datos, encontrándose este conformado por una columna polinómica llamada type que para el modelo será la variable label a predecir, junto a 373 variables numéricas y 562,163 para abordar el problema de clasificación.

Estrategia de modelado

Una vez completado el preprocesamiento de los datos y la creación de los atributos, se precede a definir la estrategia para la realización del modelado. El problema se divide en dos órdenes,

como el trabajo realizado por Choi et al. (2011): en primer lugar, la tarea de clasificar un sitio web en malicioso o benigno, y, en segundo lugar, la clasificación de un sitio web malicioso según la clase de ataque que representa para la organización al ser detectado. Para abordar estas tareas, se propone entrenar dos modelos distintos para abordar estas tareas, a partir de dos conjuntos de datos, uno general, que incluye sitios benignos y maliciosos clasificados en estas dos categorías directamente, y otro conteniendo únicamente sitios maliciosos, que excluye sitios benignos y busca distinguir entre las clases de ataque. El conjunto general se someterá a un problema de clasificación binaria, mientras que al conjunto de datos Malicioso se le implementará la técnica de clasificación multiclase.

Para ambos conjuntos se determinará cómo se dividirán para su entrenamiento, validación y prueba, los modelos con los cuales se experimentará, los hiperparámetros a ser optimizados y la métrica a partir de la cual será evaluada su rendimiento. La división del conjunto de datos para ambos casos se realizará dividiendo los registros utilizando una semilla fija en un conjunto de entrenamiento compuesto por el 70% de los datos y un 30% destinado a la prueba. Los conjuntos de datos de entrenamiento son divididos para el entrenamiento y validación de hiperparámetros en cinco subconjuntos a partir de los cuales será entrenado el modelo usando técnicas de validación cruzada y optimizando los hiperparámetros en el 20% de este subconjunto donde es ajustada la predicción del modelo para después evaluarla en el 30% inicial destinado a la prueba.

La tarea de predicción es evaluada a partir de los modelos seleccionados, Random Forest, Naive-Bayes, Linear Regression, Redes Neuronales y Adaptative Boosting Decision Trees. A continuación, se presenta brevemente las características de cada uno de ellos y de los hiperparámetros a optimizar.

El modelo de Random Forest consiste en un modelo ensamblado de árboles de decisión. Son creados arboles de decisión a partir de subconjuntos de datos que reúnen una muestra de los registros y variables del conjunto de datos de entrenamiento. Cada árbol ofrece una respuesta a la variable a predecir y la respuesta con más votos es la respuesta que finalmente el modelo asigna por registro evaluado.

Los hiperparámetros a evaluar en este modelo serán: n_estimators, que consiste en el número de árboles de decisión empleado, donde serán evaluados 50, 100 y 200 árboles, max_features, es decir, el número máximo de características que se incluirán al dividir un nodo en un árbol,

donde se analizará su rendimiento con su configuración “auto”, “sqrt” y “log2”, max_depth que es la profundidad máxima de un árbol, analizando la profundidad 0 o None, 10 y 20, min_sample_split, siendo el número mínimo de respuestas requeridas para dividir un nodo interno, analizando el rendimiento estableciendo las opciones de 2, 5 y 10 respuestas mínimas y min_sample_leaf que establece la cantidad de registros mínimos que se requiere para que exista una hoja, tomando los valores 1, 2 y 4.

El clasificador de Naive_Bayes es un modelo sencillo basado en el teorema de Bayes. Es un modelo que suele utilizarse para procesamiento de lenguaje natural, modelos con muchos valores posibles y de clasificación de texto. El teorema de Bayes relaciona con la probabilidad condicional de dos sucesos A y B. Su formulación básica es la siguiente:

$$P(C_k|X) = \frac{P(X|C_k) * P(C_k)}{P(X)}$$

Para el modelo de Naive Bayes $P(C_k \mid X)$ consiste en la probabilidad de pertenecer a la clase A dado un conjunto de características X. $P(X \mid C_k)$ es la probabilidad de observar las características B cuando un registro pertenece a la clase A. $P(C_k)$ sería la probabilidad a priori de pertenecer a la clase C_k y P(X) es la probabilidad de observar el conjunto de características X.

Este modelo supone que las variables son independientes entre sí, por eso se considera como suposición ingenua o Naive. Esta suposición permite descomponer la fórmula interpretando que la probabilidad de $P(Ck \mid X)$ está determinado por el producto de las probabilidades de las características individuales.

El hiperparámetro a optimizar en la utilización del modelo Naive Bayes será var_smoothing. Este hiperparámetro consiste en la modificación de la varianza de las características, es utilizado para agregar una mayor variabilidad en las estimaciones y evitar así el sobreajuste. Será introducido en el var_smoothing desde el valor 1 al 10^{-9}.

Regresión Logística es un modelo de aprendizaje supervisado con una gran facilidad de interpretación. Permite modelar la relación entre una variable de respuesta binaria y un conjunto de variables cuantitativas o cualitativas. Para los casos donde la variable a predecir puede tomar más de dos valores, existe una versión de este modelo conocido como Regresión Politómica o Multinomial.

La función logística, conocida como función sigmoide, es fundamental para el modelo de Regresión Logística. Esta función transforma cualquier número real a un valor en un rango de 0 a 1 permitiendo así su utilización para determinar probabilidades. La función sigmoide se define como:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Siendo σ(z) la salida de la función sigmoide y z la combinación lineal de los atributos evaluados del modelo.

Matemáticamente, el modelo de regresión lineal puede ser expresado de la siguiente forma:

$$P(Y = 1|X|) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

Siendo P(Y=1|X|) la probabilidad de que la variable dependiente Y sea 1 dado el conjunto de características X, β_0 la intersección o sesgo del modelo, y β_1, β_2, …, β_n los coeficientes asociados con las características del modelo por registro.

Los hiperparámetros para optimizar para el modelo de Regresión Logística serán “Penalty” y “C”. “Penalty”, es decir, penalización, es un hiperparámetro que controla la regularización del modelo. Esta, es una técnica utilizada para los modelos sobreajusten penalizando los coeficientes grandes. Los valores principales que pueden tomar son L1 y L2, que penalizarán de distinta manera a los coeficientes del modelo. El hiperparámetro C, por otro lado, controla la fuerza de regularización, cuanto mayor sea C, menor será la regularización del modelo y permitirá un mayor sobreajuste de las respuestas a los datos del subconjunto de entrenamiento. Serán evaluados los valores de C de -4,4 y 20.

Los modelos de redes neuronales consisten en modelos computacionales que realizan cálculos a través de un sistema de aprendizaje, mediante un conjunto de unidades de entrada y salida conectadas en las que cada conexión posee un peso asociado. En la fase de aprendizaje la red aprende ajustando los pesos para predecir la variable objetivo. La red neuronal más simple es el perceptrón, que es una red neuronal de una sola capa con tantas neuronas como salidas se requieren. Así mismo, cada neurona tiene entradas como variables explicativas tiene el problema. El incremento de cantidad de capas en una red neuronal permite resolver problemas más complejos. Al incorporar más capas, la red tendrá una capa de entrada donde se toman los

valores de las características y una capa de salida donde se obtienen las respuestas resultantes, permaneciendo ocultas las capas intermedias.

Los hiperparámetros a optimizar serán: “hidden_layers_size”, que refiere a la arquitectura de las redes neuronales, es decir, el número de neuronas en cada capa oculta y la cantidad de capas intermedia. En este hiperparámetro serán evaluadas las configuraciones de capas ocultas: 64, 32, 64-32 y 32-64. En segundo lugar, el hiperparámetro “activation” que define la función de activación utilizada en las capas ocultas de la red. Se evaluará la función tangente hiperbólica y la función ReLU, Rectified Linear Unit. El Alpha es un término de regularización, controla la magnitud de los pesos de la red, serán probados los valores de 0.0001, 0.001 y 0.01.

El modelo Adaptive Boosting Decision Trees consiste en un modelo ensamblado conformado por una cantidad determinada de árboles de decisiones. Un algoritmo de árbol de decisión consiste en un proceso iterativo de particionamiento del conjunto de las características generando subconjuntos de datos, denominados ramas por cada uno de los posibles valores que pueden tomar las variables. El modelo Adaptive Boosting Decision Trees es un modelo de ensamble porque su funcionamiento consiste en la secuenciación de modelos débiles, en este caso de Árboles de Decisión que toman algunas características del modelo, tomando uno como base un modelo inicial, alterando los pesos tras cada predicción a fin de darle una mayor importancia a aquellos registros donde los modelos previos clasificaron de manera incorrecta. Por lo tanto, cada modelo se encarga de corregir los errores del anterior. Cada modelo contribuye a la predicción final con un peso proporcional a su rendimiento en el conjunto de entrenamiento.

Los hiperparámetros a optimizar para este modelo serán n_estimators, siendo estos los números de árboles de decisión que serán empleados de manera secuencial, escogiendo a evaluar las cantidades de 50, 100, 150 y 200 estimadores. También será evaluado el hiperparámetro de tasa de aprendizaje, siendo este un elemento que afecta a la contribución que generará cada estimador al modelo. Cuanto menor sea la tasa de aprendizaje, mayor será el número de estimadores que serán necesarios para generar un rendimiento apropiado del modelo. Por otro lado, una mayor tasa de aprendizaje significará un mayor sobreajuste en el conjunto de entrenamiento. Las tasas de aprendizaje evaluadas son: 0.01, 0.1 y 1. El último hiperparámetro empleado en la optimización será “algorith”, este consiste en la variable del algoritmo empleado para el modelo AdaBoosting, pudiendo ser SAMME o SAMME.R, siendo utilizado el primero

mayormente en problemas de clasificación binaria, mientras que en el segundo caso suele emplearse con mayor habitualidad en problemas de clasificación multiclase.

Modelado y optimización de hiperparámetros

Habiendo diseñado la estrategia de modelado, a continuación, se exponen los resultados por modelo que generan las distintas combinaciones de hiperparámetros seleccionada para los problemas de clasificación entre sitios benignos y maliciosos con el conjunto de datos General y el problema de clasificación multi-clase de sitios web maliciosos con el conjunto de datos Maliciosos.

Para la aplicación del modelo de Random Forest se dispuso la siguiente combinación de hiperparámetros:

- N-estimators: 50, 100, 200.
- Max-features: auto, sqrt, log2.
- Max-depth: None, 10, 20.
- Min-sample-split: 2, 5, 10.
- Min-sample-leaf: 1,2,4.

La optimización de hiperparámetros arrojó que la mejor combinación de los mismos fue el tunéo con N-estimators: 200, max-features: “auto”, max-depth: “none”, min-sample-split: 2 y min-sample-leaf: 1.

El valor obtenido para la métrica a maximizar ROC-AUC en el conjunto de prueba fue de 0.99881. Entre las otras medidas obtenidas, se alcanzó una precision weighted average de 0.99, recall weighted average de 0.99 y f1-score weighted average de 0.99.

La matriz de confusión del conjunto de prueba obtenido fue la siguiente:

Tabla 5

Matriz de confusión conjunto de prueba para Random Forest clasificación binaria

Concepto	Predicción – 0	Predicción - 1
Etiqueta verdadera – 0	120,734	275

Etiqueta verdadera – 1	1,334	46,306

En cuanto al problema de la clasificación multiclase se obtuvieron los siguientes resultados:

La optimización arrojó la misma configuración que la mejor combinación de hiperparámetros que la combinación empleada para la clasificación entre benignos y maliciosos, con N-estimators: 200, max-features: “auto”, max-depth: “none”, min-sample-split: 2 y min-sample-leaf: 1.

El valor de la métrica ROC-AUC alcanzada por el modelo fue de 0.999 para la predicción de cada una de las clases contra el resto, alcanzando un Recall de 0,967 para Phishing, 0.959 para Malware, 0.998 para Spam y 0.999 para Defacement.

La matriz de confusión del conjunto de prueba para la clasificación entre clases de ataque malicioso fue la siguiente:

Tabla 6

Matriz de confusión conjunto de prueba clasificación multiclase para Random Forest

Concepto	Predicted – Phishing	Predicted – Malware	Predicted – Spam	Predicted - Defacement
True – Phishing	17,896	0	17	1
True – Malware	14	4,165	163	1
True – Spam	118	9	9,993	27
True - Defacement	7	0	27	15,261

Para observar el rendimiento para las otras métricas evaluadas se obtuvo los siguientes resultados:

Tabla 7

Informe de clasificación conjunto de prueba clasificación multiclae para Random Forest

Concepto	Precision	Recall	F1-Score	Support
Defacement	0.99	1.00	1.00	17,914
Malware	1.00	0.96	0.98	4,343

Phishing	0.98	0.98	0.98	10,147
Spam	1.00	1.00	1.00	15,295
Accuracy			0.99	47,699
Macro Avg	0.99	0.99	0.99	47,699
Weighted Avg	0.99	0.99	0.99	47,699

La aplicación del modelo Naive-Bayes fue realizada a partir de la combinación del hiperparámetro var-smoothing adoptando los valores de 10^{0} a 10^{-9}. A continuación se presentan los resultados tanto para la clasificación del conjunto de datos General como para el problema de clasificación de clase de sitios maliciosos.

Para el problema de clasificación entre sitios benignos y maliciosos, el valor de var-smoothing que arrojó los resultados con una mayor performance en las métricas evaluadas fue de 10^{-5}. El valor de la métrica ROC AUC para este caso fue de 0.9331. En las otras medidas obtenidas, se alcanzó una precision weighted average de 0.90, recall weighted average de 0.9 y f1-score weighted average de 0.9.

La matriz de confusión del conjunto de prueba obtenido fue la siguiente:

Tabla 8

Matriz de confusión conjunto de prueba clasificación binaria para Naive-Bayes

Concepto	Predicción – 0	Predicción - 1
Etiqueta verdadera – 0	112,171	8,838
Etiqueta verdadera – 1	8,008	39,632

En cuanto al rendimiento del modelo Naive-Bayes para el problema de clasificación multiclase, el hiperparámetro evaluado var-smoothing tuvo su mejor rendimiento cuando su valor fue 10^{-6}. se obtuvo una accuracy multiclass de 0.8152, la métrica ROC-AUC para Defacement 0.93, para Spam 0.97, para Malware 0.86 y para Phishing 0.96.

La matriz de confusión del conjunto de prueba para la clasificación utilizando el modelo Naive-Bayes entre clases de ataques malicioso fue la siguiente:

Tabla 9

Matriz de confusión conjunto de prueba clasificación multiclase para Naive-Bayes

Concepto	Predicted – Phishing	Predicted – Malware	Predicted – Spam	Predicted - Defacement
True – Phishing	15,665	1,706	173	370
True – Malware	1,085	2,740	358	160
True – Spam	1,205	355	8,368	219
True - Defacement	2,343	708	132	12,112

Así mismo, para conocer el rendimiento de las otras métricas de rendimiento se expone el siguiente informe de clasificación:

Tabla 10

Informe de clasificación conjunto de prueba clasificación multiclase para Naive-Bayes

Concepto	Precision	Recall	F1-Score	Support
Defacement	0.77	0.87	0.82	17,914
Malware	0.5	0.63	0.56	4,343
Phishing	0.93	0.82	0.87	10,147
Spam	0.94	0.79	0.86	15,295
Accuracy			0.82	47,699
Macro Avg	0.78	0.78	0.78	47,699
Weighted Avg	0.83	0.82	0.82	47,699

La aplicación del modelo de Regresión Logística obtuvo como resultado en la selección de hiperparámetros a un Coste de 0.033598 y una penalidad de L2 para el problema de clasificación binaria. En cuanto a las métricas evaluadas, el ROC AUC obtenido fue de 96.67, mientras que las métricas precision weighted average, recall weighted average y f1-score weighted average alcanzaron un valor de 0.93 para cada una de ellas.

La matriz de confusión arrojada por este modelo es la siguiente:

Tabla 11

Matriz de confusión conjunto de prueba clasificación binaria para Regresión Logística

Concepto	Predicción – 0	Predicción - 1
Etiqueta verdadera – 0	78,864	1,831
Etiqueta verdadera – 1	5,964	25,774

En cuanto al problema multiclase y a los resultados obtenidos a partir de su evaluación sobre el conjunto de datos de prueba, los hiperparámetros seleccionados fueron un Coste de: 0.0001 y una penalidad de L2. Los valores alcanzados de las métricas ROC-AUC para cada una de las clases fueron 0.896 para Defacement, 0.82 para Malware, 0.913 para Phishing y 0.933 para Spam.

La performance del modelo para las otras métricas analizadas fueron las siguientes:

Tabla 12

Matriz de confusión conjunto de prueba clasificación multiclase para Regresión Logística

Concepto	Precision	Recall	F1-Score	Support
Defacement	0.69	0.83	0.75	17,914
Malware	0.33	0.11	0.17	4,343
Phishing	0.86	0.76	0.8	10,147
Spam	0.79	0.8	0.79	15,295
Accuracy			0.74	47,699
Macro Avg	0.66	0.63	0.63	47,699
Weighted Avg	0.72	0.74	0.72	47,699

Así mismo, la matriz de confusión obtenida que expone la distribución de las clasificaciones realizadas por el modelo comparadas a sus clases verdaderas fue la siguiente:

Tabla 13

Informe de clasificación conjunto de prueba clasificación multiclase para Regresión Logística

Concepto	Predicted – Phishing	Predicted – Malware	Predicted – Spam	Predicted - Defacement
True – Phishing	14,798	331	462	2,323
True – Malware	2,892	499	338	614

True – Spam	1,627	406	7,691	423
True - Defacement	2,231	281	481	12,302

El rendimiento de los modelos de Redes Neuronales para el problema de clasificación binaria entre sitios benignos y malignos obtuvo como mejores hiperparámetros una función de activación tangente hiperbólica: “tanh”, un Alpha de 0.01 como parámetro de regularización del ajuste y una combinación de una capa oculta de 64 neuronas y otra de 32.

La performance obtenida para la clasificación binaria con esta clasificación de hiperparámetros fue de un ROC-AUC de 0.99, obteniendo en las métricas de Recall, Precision y F1-Score valores de 0.98 para estas métricas con sus pesos equilibrados.

La matriz de confusión que presenta la predicción realizada para la clasificación binaria empleando este modelo fue la siguiente:

Tabla 14

Matriz de confusión conjunto de prueba clasificación binaria Redes Neuronales

Concepto	Predicción – 0	Predicción - 1
Etiqueta verdadera – 0	80,050	645
Etiqueta verdadera – 1	1,209	30,529

La performance obtenida para la clasificación multiclase con redes neuronales tuvo como una mejor combinación de hiperparámetros los mismos que para la clasificación binaria, función de activación: “tanh”, Alpha de 0.01 y dos capas ocultas de 64 y 32 neuronas.

En cuanto a la métrica ROC-AUC, la performance obtenida para la clase Defacement fue de 0.999, para Malware 0.996, para Phishing 0.997 y para Spam 0.9998. En cuanto a las restantes métricas obtenidas, se obtuvieron los siguientes valores:

Tabla 15

Matriz de confusión conjunto de prueba clasificación multiclase para Redes Neuronales

Concepto	Precision	Recall	F1-Score	Support
Defacement	0.69	0.83	0.75	17,914
Malware	0.33	0.11	0.17	4,343

Phishing	0.86	0.76	0.8	10,147
Spam	0.79	0.8	0.79	15,295
Accuracy			0.74	47,699
Macro Avg	0.66	0.63	0.63	47,699
Weighted Avg	0.72	0.74	0.72	47,699

Así mismo, la distribución de respuestas predichas en función de su valor verdadero se demuestra de la siguiente manera a partir de la matriz de confusión obtenida:

Tabla 16

Informe de clasificación conjunto de prueba clasificación multiclase para Redes Neuronales

Concepto	Predicted – Phishing	Predicted – Malware	Predicted – Spam	Predicted - Defacement
True – Phishing	17,806	9	36	9
True – Malware	28	4,187	143	11
True – Spam	91	76	9,885	70
True - Defacement	13	1	17	15,317

Los resultados obtenidos del modelo Adaptative Boosting Decision Tree en cuanto a los hiperparámetros que proporcionaron un mayor valor en las métricas evaluadas fueron el algoritmo SAMME.R, la tasa de aprendizaje de 1.0 y 200 estimadores empleados.

Las métricas de rendimiento obtenidas para este modelo, en la primera instancia de clasificación binaria se obtuvo un ROC-AUC de 0.99, y un recall, f1-score y precision de 0.99 para cada una de las métricas mencionadas.

La matriz de confusión para la clasificación binaria de este modelo fue la siguiente:

Tabla 17

Matriz de confusión conjunto de prueba clasificación binaria para Adaptative Boosting Decision Tree

Concepto	Predicción – 0	Predicción - 1
Etiqueta verdadera – 0	120,606	403

Etiqueta verdadera – 1	1,328	46,312

Para la clasificación multiclase la combinación de hiperparámetros óptima fue también el algoritmo SAMME.R, la tasa de aprendizaje de 1.0 y 200 estimadores empleados.

Así mismo, las métricas obtenidas en la predicción de este modelo fueron un ROC AUC de 0.99 para cada una de las clases, y un Recall, Precision y F1-score 0.99.

La matriz de confusión obtenida para la predicción multiclase fue la siguiente:

Tabla 18

Matriz de confusión conjunto de prueba clasificación multiclase para Adaptative Boosting Decision Tree

Concepto	Predicted – Phishing	Predicted – Malware	Predicted – Spam	Predicted - Defacement
True – Phishing	17,872	14	18	10
True – Malware	24	4227	85	7
True – Spam	128	84	9,889	46
True - Defacement	4	1	7	15,283

Selección y Operacionalización

Selección y optimización del modelo

Habiendo obtenido los resultados de cada uno de los modelos, se procede a la selección del modelo que ofreció la mejor performance para la resolución de ambos problemas, la clasificación binaria entre sitios benignos y maliciosos y la clasificación multiclase entre sitios web de Defacement, Malware, Phishing y Spam, a continuación, se presentan los resultados obtenidos para su comparación:

Tabla 19

Resultados de métricas evaluadas por modelo para clasificación binaria

Concepto	ROC-AUC	Recall	Precision	F1-Score
Random Forest	0.998	0.99	0.99	0.99

Naive-Bayes	0.9331	0.9	0.9	0.9
Logistic Regression	0.9667	0.93	0.93	0.93
Redes Neuronales	0.99	0.98	0.98	0.98
Adaptive Boosting Decision Trees	0.99	0.97	0.97	0.97

Por otro lado, los resultados obtenidos según las métricas seleccionadas para la clasificación multiclase fueron los siguientes:

Tabla 20

Resultados de métricas evaluadas por modelo para clasificación multiclase

Concepto	Random Forest	Naive-Bayes	Logistic Regression	Redes Neuronales	Adaptive Boosting Decision Trees
ROC-AUC Defacement	0.999	0.934	0.8963	0.9994	0.999
ROC-AUC Malware	0.999	0.867	0.8204	0.9961	0.999
ROC-AUC Phishing	0.999	0.96	0.9131	0.997	0.999
ROC-AUC Spam	0.999	0.972	0.9331	0.9998	0.999
Recall	0.99	0.82	0.72	0.99	0.99
Precision	0.99	0.83	0.74	0.99	0.99
F1-Score	0.99	0.82	0.72	0.99	0.99

En función de los valores obtenidos en las métricas seleccionadas, se observa que los modelo que se desempeñan de manera más eficaz en base de las medidas de performance escogidas son los modelos de Random Forest y Adaptive Boosting Decision Trees. Comparando la cantidad de predicciones incorrectas observadas en la matriz de confusión de cada uno de los modelos, se observa que los aciertos del algoritmo de Random Forest fueron mayores que en el modelo Adaptive Boosting Decision Trees tanto para la clasificación binaria como para la clasificación multiclase. Por lo tanto, el algoritmo de Random Forest es el modelo escogido para la implementación del modelo y la propuesta de operacionalización. Para su operacionalización

se debe conocer como el modelo se desempeña y buscar optimizar su rendimiento en búsqueda de una implementación adecuada.

Al haber escogido el modelo de Random Forest para la implementación, a fin de reducir el volver la predicción eficiente, se propone la búsqueda de la reducción de la dimensionalidad del conjunto de datos de manera tal que sea posible la reducción del tiempo de demora que el modelo requiera para generar la predicción de un registro desde la carga del URL, la obtención de los atributos para el registro y la elaboración de su predicción, sin que esto signifique una disminución en su performance.

Para la disminución en la cantidad de atributos se emplea el informe de importancia de características. Partiendo en cada caso de un conjunto de datos compuesto por 373 atributos se procede a la obtención de la importancia de cada una de las características, eligiendo conservar únicamente los atributos cuya importancia supere el 0.001. A partir de esta restricción, en el conjunto de datos General se disminuye la dimensionalidad del conjunto de datos a un dataset de 68 atributos y la columna type a predecir. Por otro lado, para el conjunto de datos Malicioso, siendo que la utilización de un modelo de clasificación multiclase utiliza la estrategia de One vs. Other, fueron obtenidos cuatro informes de importancias de características. Se procedió a la obtención del promedio de las importancias y a la eliminación de los atributos siguiendo el mismo umbral de 0.001 utilizado para el conjunto de datos General. La eliminación de estas características de importancia inferior al 0.001 permitió disminuir el conjunto de datos Malicioso a un dataset de 79 atributos y la columna type a predecir.

Tras la reducción de la dimensionalidad del conjunto de datos se procede a la comprobación de la performance del modelo donde se observa que su rendimiento no sufrió alteraciones. Los resultados de esta comprobación serán presentados en el siguiente apartado donde se describe y visualiza el funcionamiento del modelo y los errores que comete.

Visualización de performance y explicación del modelo

Tras la selección del modelo y la optimización realizada a partir de la disminución de la dimensionalidad de los conjuntos de datos en base a la importancia de las características de los modelos utilizados, fueron obtenidos los resultados definitivos del modelo. Este apartado tendrá la finalidad de exponer la manera en la que se desempeña el modelo y la eficacia que ha alcanzado para llevar a cabo sus predicciones sobre el conjunto de datos.

En primer lugar, se analizará el rendimiento de la métrica elegida para la evaluación de su performance. El ROC-AUC fue elegida la métrica a evaluar durante la elaboración de la estrategia del modelado. Para tanto la clasificación binaria entre sitios web maliciosos y benignos como para la clasificación multiclase entre sitios web que representan ataque de Defacement, Spam, Malware o Phishing, el valor alcanzado en esta métrica fue de 0.999, lo que implica una capacidad alta de distinción entre una clase y la otra.

Ilustración 4

Curva ROC-AUC en conjunto de prueba para clasificación binaria

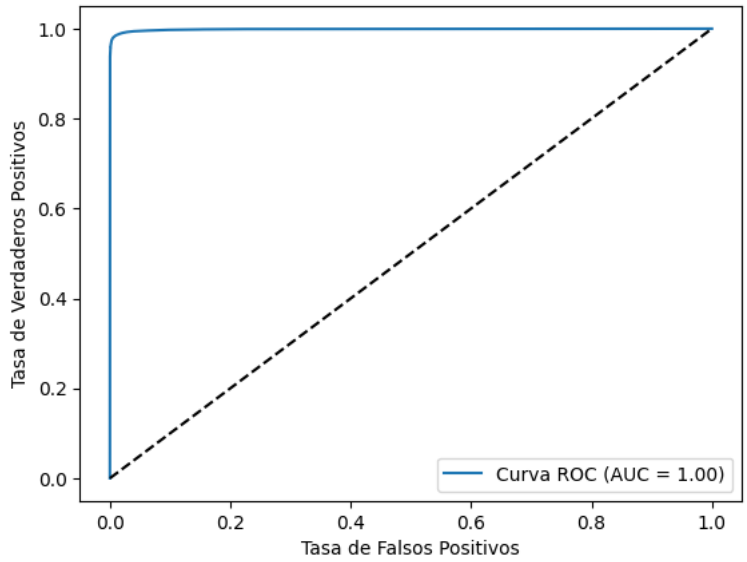
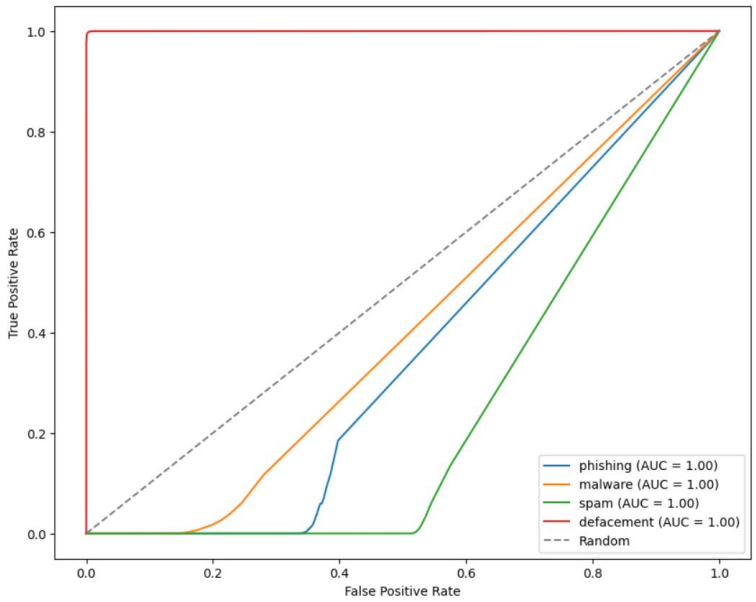


Ilustración 5

Curva ROC-AUC en conjunto de prueba para clasificación multiclase



La matriz de confusión para cada uno de los modelos, según se observa a continuación, permitirá observar el funcionamiento de ambos modelos y cómo se encuentra distribuida la clasificación permitiendo conocer cómo se encuentra distribuido el error.

Ilustración 6

Matriz de Confusión en conjunto de prueba para clasificación binaria

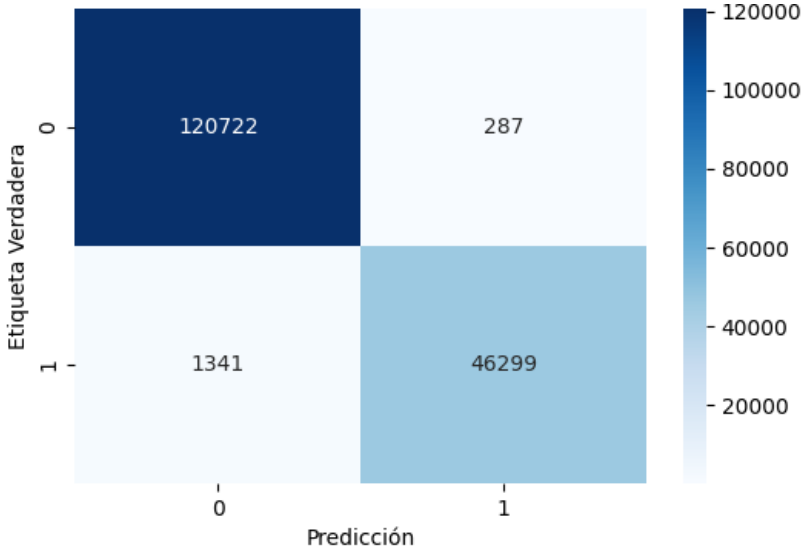
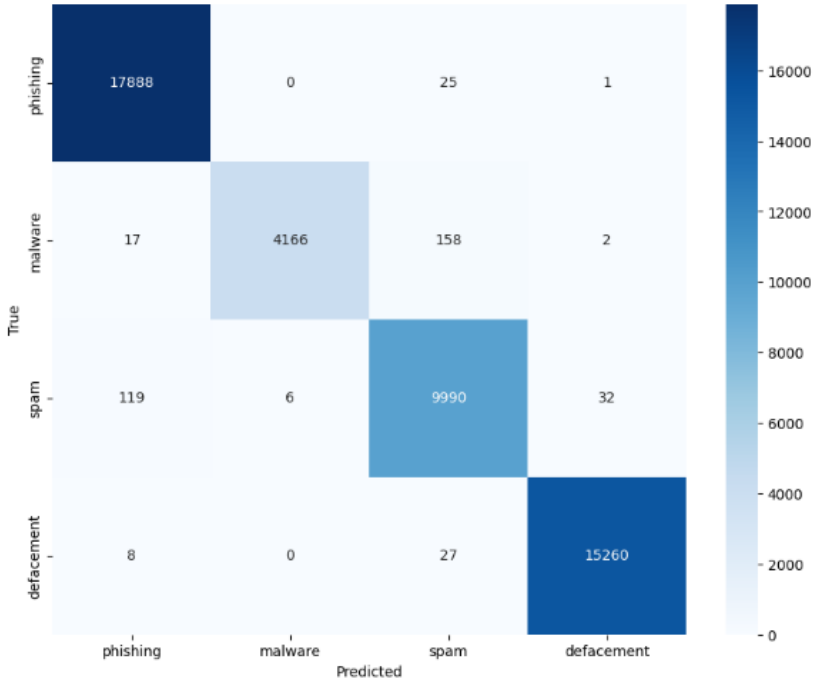


Ilustración 7

Matriz de confusión conjunto de prueba para clasificación multiclase

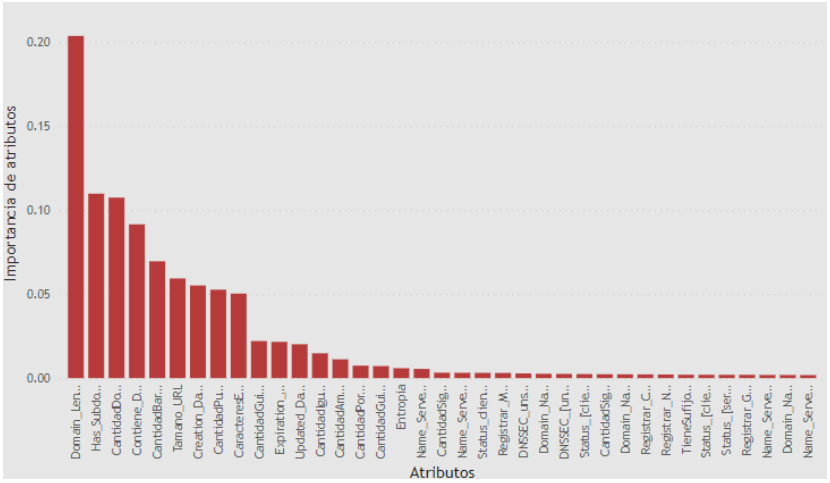


En ambas matrices se observa una gran capacidad del modelo para llevar a cabo sus predicciones. En el modelo de clasificación binaria se observa que existe una mayor presencia de falsos negativos que positivos, es decir que el modelo se equivoca en su predicción en mayor medida clasificando sitios web como benignos tratándose de sitios maliciosos. En el modelo de clasificación multiclase se observa que los mayores errores son cometidos en el modelo para la clasificación de sitios web de Malware.

A continuación, a fin de describir cómo es el funcionamiento del modelo, se exponen los gráficos de importancia de las características. Los modelos de árboles son de fácil interpretabilidad ya que ofrecen información sobre la contribución de las características a la predicción del algoritmo. Las importancias de las características se calculan en función de la frecuencia del uso que realizan los árboles estimadores el atributo para dividir los nodos y la ganancia de información que conlleva su utilización. Los atributos de mayor importancia para la clasificación binaria se encuentran a continuación:

Ilustración 8

Importancia de los atributos en clasificación binaria

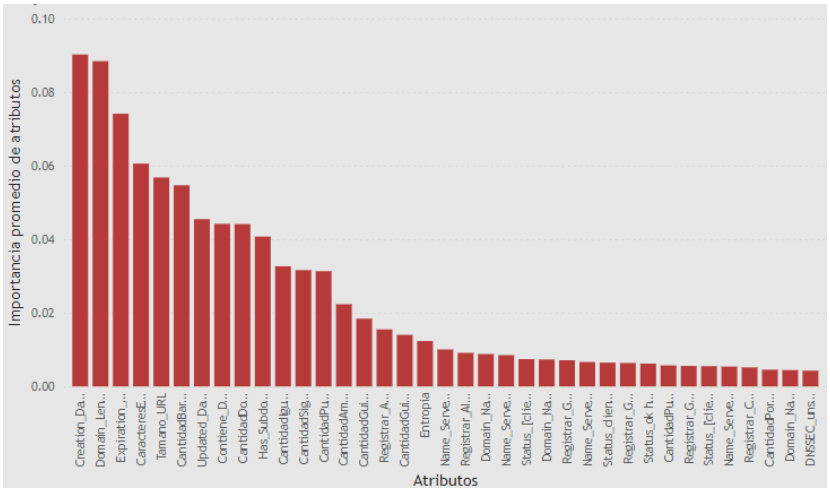


Los atributos que más se destacaron en la clasificación binaria fueron el tamaño del dominio, la presencia de un subdominio, cantidad de “dos puntos”, si el URL contiene “doble barra”, la longitud del URL, la cantidad de barras que posee, la fecha de creación, entre otros.

Seguidamente se expone la importancia promedio de los atributos para el modelo de Random Forest en la clasificación multiclase:

Ilustración 9

Importancia de los atributos en clasificación multiclase

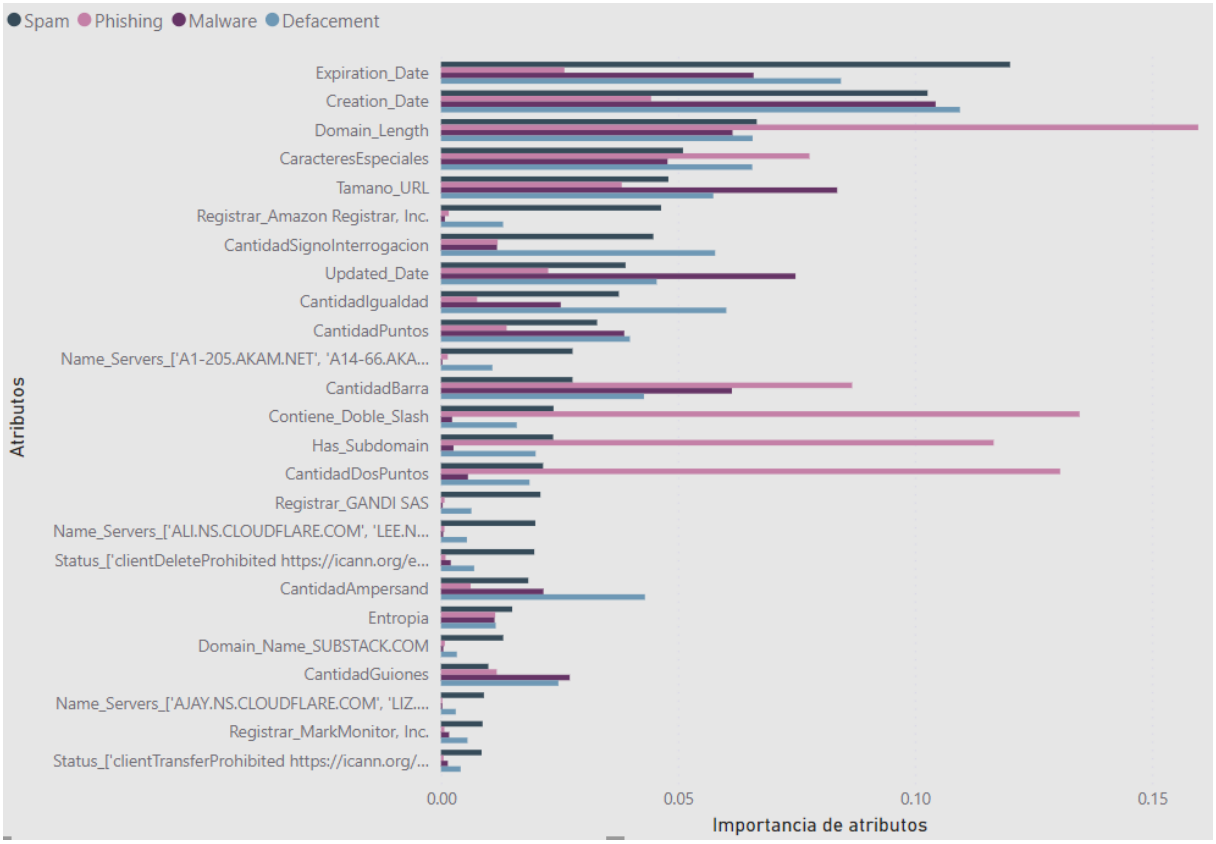


Como puede observarse en el gráfico anterior, la importancia de los atributos se encuentra mayormente en la fecha de la creación del sitio web, seguida por el tamaño del dominio, la fecha de expiración, la cantidad de caracteres especiales utilizados en el URL, el tamaño del URL, la cantidad de barras, la fecha de expiración, entre otros.

Por último, se observa los valores obtenidos para los atributos más relevantes en cuanto a su importancia en el modelo multiclase, este permite observar como varía la importancia de las variables conforme a la clase de sitio web malicioso que se trate.

Ilustración 10

Importancia de variables según clase de sitio web malicioso



A partir de lo que se puede observar, las variables más significativas para los modelos de Spam resultan ser la fecha de expiración, creación y el tamaño del dominio. En el caso de Phishing, el tamaño del dominio, si el URL contiene doble barra, tiene subdominio y la cantidad de puntos fueron variables fundamentales en la predicción. Para el caso de Malware, los atributos más importantes fueron la fecha de creación, el tamaño del URL, la fecha de actualización y el tamaño del dominio. Por último, en el caso de Defacement, los atributos con mayor relevancia fueron la fecha de expiración, fecha de creación, el tamaño del dominio, la cantidad de signos igual, cantidad de signos de interrogación, entre otros.

Propuesta de operacionalización

La implementación del modelo de Machine Learning evaluado durante el presente documento requiere la selección de elementos que conforman la arquitectura de datos donde será ejecutado el modelo y puesto en funcionamiento en el contexto de una organización. Por lo tanto, a continuación, se postulará una arquitectura de datos donde el modelo tratado puede ser implementado.

Los componentes tecnológicos para definir en el despliegue del modelo serán aquellos que llevarán a cabo el ciclo de vida del dato. El dato en la organización debe ser obtenido de fuentes, incorporado a través de la ingesta de datos, almacenado, procesado y comunicado. A fin de que los procesos de ingesta, almacenado, procesamiento y comunicación de los datos sea realizado de manera adecuada, deben ser escogidas herramientas que permitan llevar a cabo estos procesos. Así mismo, deben ser definidos las fuentes de datos a partir de las cuales será obtenida la información a ser utilizada por la organización en la implementación de este modelo. A continuación, se presenta una propuesta de arquitectura empresarial que permita el despliegue de los modelos de clasificación tratados.

En primer lugar, en cuanto a las fuentes de los datos, los datos de los cuales el modelo empleado se alimenta son los URL que utilizan los empleados en su navegación y los datos de la base de datos WHOIS. Por lo tanto, las fuentes de datos para esta información a adquirir será el uso de servicios de consulta WHOIS y la utilización de logs de navegación para la obtención de los URL.

Tras determinar las fuentes de datos, es necesario establecer qué componente tecnológico será empleado para la ingesta de estos. Para esta tarea se propone la utilización de Azure Event Hubs. Se considera que esta herramienta puede ser eficaz en la ingesta de datos en tiempo real y la obtención de grandes volúmenes de datos con la posibilidad de escalar en su funcionamiento.

En cuanto al almacenamiento de datos, un Data Lake será empleado a fin de posibilitar el almacenamiento de datos tanto estructurados como no estructurados. La posibilidad de emplear datos no estructurados a partir de un Data Lake permitirá a la organización que implemente el modelo incorporar datos no estructurados si desea para la predicción y fortalecer, de esta manera, la predicción. Se propone la utilización de la tecnología Azure Data Lake Storage.

Para el procesamiento de datos, limpieza, obtención de los atributos para la preparación de los datos se debe definir una tecnología que permita el entrenamiento del modelo. Este proceso será realizado a partir de la tecnología Azure Databricks.

En cuanto al despliegue y puesta en producción del modelo, se propone la utilización de la tecnología Azure Machine Learning. Esta herramienta permitirá la experimentación e implementación de los modelos en producción ya que posee una interfaz gráfica que permite a los usuarios la posibilidad de realizar su tarea sin la necesidad de una programación intensiva.

Los datos serán monitoreados a partir de la utilización de Azure Purview como herramienta de gobernanza. Esta tecnología será utilizada para proteger la privacidad de los datos, la asignación de responsabilidades sobre los datos, gestión de accesos, entre otros.

La visualización de los datos será realizada a través de Power BI. Esta aplicación será utilizada para brindar información al usuario sobre las amenazas a las que se encuentra expuesta su organización en base a los sitios web que visitan los empleados que forman parte de la red de la organización. La elaboración de tableros permitirá al usuario conocer que tipos de ataques puede recibir la entidad y realizar las alertas necesarias para evitar la pérdida de información o un daño en su privacidad.

Conclusión

El presente trabajo permitió realizar un análisis comparativo de la performance de los modelos Random Forest, Neural Net, Logistic Regresion, Naive Bayes y Adaptive Boosting Decision Trees donde se verificó que el modelo Random Forest alcanzó valores superiores en las métricas seleccionadas ROC-AUC, Precision, Recall y F1-Score con un valor de 0.999 para cada una de las métricas.

En primer lugar, se realizó un planteo metodológico para la comparación de la performance de modelos de aprendizaje automático supervisado para la clasificación binaria y multiclase de sitios web benignos y maliciosos. El planteo metodológico fue realizado basándose en los antecedentes relevantes de investigaciones anteriores.

Habiendo definido el problema, se desarrolló la metodología planteada siguiendo el ciclo de vida de un proyecto de ciencia de datos, realizando la adquisición de los datos, su preparación, la ingeniería de predictores y el entrenamiento de los modelos para posibilitar la selección.

Por último, fue realizada la selección del modelo, escogiendo a Random Forest por presentar una mejor performance en las métricas seleccionadas. Se optimizó el funcionamiento del modelo reduciendo la dimensionalidad del conjunto de datos a partir de la eliminación de atributos, basándose en el informe de importancia de los atributos. Este informe, junto con las curvas ROC-AUC y las matrices de confusión permitieron la explicación del funcionamiento del modelo. Se propuso, finalmente, una arquitectura de datos para el despliegue del modelo.

El presente trabajo puede ser de utilidad para la gestión y protección de los datos en organizaciones que deseen implementar métodos para monitorear la navegación de los usuarios de la red en la que realizan sus funciones.

Se identifica como líneas de trabajo futuro la posibilidad de incorporar mayores conjuntos de datos de URL maliciosos a fin de entrenar modelos más eficientes para detectar y clasificar sitios web que amenazan a los datos pertenecientes a la entidad. Así mismo, la variedad de atributos puede incrementarse, incorporando características del servidor, la utilización de su contenido HTML, entre otros.

Referencias bibliográficas

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). *A Comparison of Machine Learning Techniques for Phishing Detection.* Dallas,TX 75275: Southern Methodist University.

Choi, H., Zhu, B., & Lee, H. (2011). *Detecting Malicious Web Links and Identifying Their Attack Types.*

Deng, W., Peng, Y., Yang, F., & Song, J. (2019). *Feature optimization and hybrid classification for malicious.* Wuhan, P. R. China: School of Information and Communication Engineering, Hubei University of Economics.

Gressel, G., Ashok, A., Poornachandran, P., Darling, M., & Heileman, G. (2015). *A Lexical Approach for Classifying Malicious URLs.* Conference: 2015 International Conference on High Performance Computing & Simulation (HPCS).

Kaggle. (2023). Obtenido de Malicious URLs dataset: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

Kaggle. (2023). *Spam URLs Classification Dataset*. Obtenido de https://www.kaggle.com/datasets/shivamb/spam-url-prediction

Kaspersky. (2023). *Kaspersky security bulletin 2022, overall statistics for 2022.*

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2018). *A stacking model using URL and HTML features for phishing webpage.* Future Generation Computer Systems.

Ma, J., Saul, L., Savage, S., & Voelker, G. (2009). *Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs.* Department of Computer Science and Engineering University of California, San Diego.

Ma, J., Saul, L., Savage, S., & Voelker, G. (2009). *Identifying Suspicious URLs: An Application of Large-Scale Online Learning.* Department of Computer Science & Engineering, UC San Diego.

Mamun, M. S., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016). *Detecting Malicious URLs Using Lexical Analysis.* University of New Brunswick, Fredericton, NB, Canada.

Martín de Diego, I., & Fernández Isabel, A. (2020). *Ciencia de Datos para la Ciberseguridad.* Ra-Ma.

McGahagan, J., Bhansali, D., Pinto-Coelho, C., & Cukier, M. (2019). *A Comprehensive Evaluation of Webpage Content Features for Detecting Malicious Websites.* Natal, Brasil: Latin-American Symposium on Dependable Computing (LADC).

Mohaisen, A. (2015). *Towards Automatic and Lightweight Detection and Classification of Malicious Web Contents.* 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies.

Pang-Ning, T., Steinbach, M., Karpatne, A., & Kumar, V. (2018). *Introduction to Data Mining (Second Edition).* Pearson.

Popescu, A. S., Gavrilut, D. T., & Prelipcean, D. B. (2015). *A Study on Techniques for Proactively Identifying Malicious URLs.* 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing.

Scikit-Learn. (2023). Obtenido de https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

Scikit-Learn. (2023). Obtenido de https://scikit-learn.org/stable/modules/linear_model.html#bayesian-regression

Scikit-Learn. (2023). Obtenido de https://scikit-learn.org/stable/modules/ensemble.html#random-forests-and-other-randomized-tree-ensembles

Scikit-Learn. (2023). Obtenido de https://scikit-learn.org/stable/modules/ensemble.html#adaboost

Scikit-Learn. (2023). Obtenido de https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification

Sun, B., Akiyama, M., Yagi, T., Hatada, M., & Mori, T. (2016). *AutoBLG: Automatic URL Blacklist Generator Using Search Space Expansion and Filters.* 20th IEEE Symposium on Computers and Communication (ISCC).

Universidad de Buenos Aires. (2023). *Implementación de modelos de aprendizaje automático*. Obtenido de https://e72102.readthedocs.io/es/latest/

Xu, L., Zhan, Z., Xu, S., & Ye, K. (February de 2013). Cross-layer detection of malicious websites. *In Proceedings of the third ACM conference on Data and application security and privacy*, págs. 141-152.

Apéndices

Enlace a carpeta de Google Drive:

https://drive.google.com/drive/folders/1CeOhNrNzH5r0UDbGyF_Yl7OF8JDwsvZ0?usp=drive_link

Autor del trabajo: Luciano Martín Hainze

Anexo – Reporte del Mentor

Este trabajo consiste en el planteo y desarrollo de un proyecto de Ciencia de Datos para la clasificación binaria y multiclase de sitios web benignos y maliciosos.

Se presenta el problema de comparación de modelos de aprendizaje automático supervisado. Se utilizan modelos analíticos predictivos y metodologías avanzadas de aprendizaje automático. Se aplican métricas para evaluar los diferentes modelos.

En cuanto al planteo del problema, el mismo se encuentra correctamente definido. En la introducción se realiza una descripción de la problemática, se presentan antecedentes en la aplicación de modelos de aprendizaje automático para la detección de sitios web maliciosos.

El objetivo general del trabajo es comparar los diferentes modelos de aprendizaje automático midiendo la performance de cada uno de ellos y decidir cuál funciona mejor. El mismo es coherente con los objetivos que se pretenden alcanzar.

El planteo del problema y los objetivos se encuentran articulados, presentan coherencia interna y corresponden con los contenidos de la especialización que está realizando.

El tema elegido resulta interesante y novedoso para aplicar como trabajo final de la especialización.

El trabajo se realiza utilizando una base de datos abierta.

En el desarrollo del trabajo se observa la fundamentación del problema, el procesamiento de datos, la aplicación de las diferentes metodologías, los resultados obtenidos, la conclusión articulada con el objetivo del trabajo y la bibliografía actualizada.

Se presentan los resultados en forma de tabla y con gráficos que favorecen la interpretación.

De aprendizaje automático utilizó Random Forest, Naive-Bayes, Regresión lineal, Redes Neuronales y Adaptative Boosting Decision Trees.

Para la realización de este trabajo se realizó una etapa de preprocesamiento de datos que incluye el proceso de limpieza, selección y transformación de atributos para poder aplicar los métodos de aprendizaje automático. Se presenta un detalle de todos los procedimientos realizados en el trabajo que son consistentes con los contenidos académicos desarrollados en la especialización. Asimismo, se presenta la aplicación de las metodologías abordadas en las diferentes asignaturas utilizando herramientas informáticas adecuadas a cada tema.

Se considera que se han alcanzado los objetivos propuestos y se presentan los resultados correspondientes acompañados de una fundamentación adecuada y de la bibliografía correspondiente.

Mentora: Nélida Mónica Cantoni Rabolini