Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencias de la Computación

Yongbum Park (20117)

Santiago Taracena Puga (20017)

Laboratorio 01

Detección de phishing

Security Data Science

Catedrático: Jorge Yass

Introducción

En el ámbito actual de la ciberseguridad, la detección y prevención de ataques de phishing se han convertido en prioridades esenciales para salvaguardar la integridad de la información y la privacidad de los usuarios en línea. El phishing, basado en la astuta manipulación psicológica de las víctimas, se ha erigido como una de las tácticas más eficaces utilizadas por ciberdelincuentes para obtener información confidencial, como credenciales de acceso a cuentas bancarias, datos personales y contraseñas.

El presente informe aborda la implementación de un modelo de Machine Learning (ML) destinado a clasificar URLs como legítimas o phishing, fundamentándose en el análisis de características intrínsecas de las direcciones web. Este enfoque busca contrarrestar los sofisticados métodos empleados por los atacantes, quienes, a través de ingeniería social, logran crear réplicas convincentes de sitios web legítimos para engañar a los usuarios.

El laboratorio se estructura en dos partes esenciales: la primera se centra en la ingeniería de características, exploración de datos, derivación de características y preprocesamiento, mientras que la segunda se enfoca en la implementación de modelos de ML y la evaluación de su desempeño. Todo el proceso se desarrollará en el lenguaje de programación Python, aprovechando herramientas y librerías especializadas.

La detección de phishing no solo constituye un desafío técnico, sino también un imperativo para mitigar riesgos financieros y salvaguardar la confidencialidad de la información sensible. Este laboratorio proporcionará una perspectiva práctica sobre cómo abordar esta problemática mediante la aplicación de técnicas de ML, ofreciendo a los participantes una valiosa experiencia en la intersección de la seguridad informática y la ciencia de datos.

Parte 1 - Ingeniería de Características

En la primera fase de este laboratorio, nos enfocaremos en la Ingeniería de Características, un paso crucial para dotar al modelo de Machine Learning de la capacidad de discernir entre URLs legítimas y potenciales intentos de phishing. Comenzaremos por cargar el conjunto de datos proporcionado en un dataframe de pandas, permitiendo una visualización inicial de cinco observaciones para comprender la estructura del mismo. A continuación, evaluaremos la distribución de las etiquetas en la columna 'status', identificando la proporción de URLs etiquetadas como 'legit' y 'phishing', con el objetivo de determinar si el conjunto de datos presenta un equilibrio adecuado.

Posteriormente, nos sumergiremos en la Derivación de Características, donde exploraremos las ventajas del análisis de una URL en comparación con otros datos, como el tiempo de vida del dominio o las características de la página web. Inspirándonos en artículos especializados en la clasificación de phishing, identificaremos al menos quince funciones basadas en estas investigaciones para enriquecer nuestro conjunto de datos con características relevantes.

El proceso de preprocesamiento será esencial para convertir la variable categórica 'status' en una variable binaria, eliminar la columna del dominio y realizar ajustes necesarios para garantizar la coherencia y eficacia del análisis. Finalmente, emplearemos la herramienta pandas_profiling para generar un reporte de perfil que nos proporcionará insights cruciales sobre las características del conjunto de datos y nos guiará en la selección de las características más significativas para la detección de phishing.

Después de generar el reporte de perfil con pandas_profiling, se llevaron a cabo varios ajustes en base a las observaciones obtenidas. Se filtraron algunas características que mostraron una baja variabilidad o que no aportaban información relevante para la detección de phishing, como aquellas con una correlación muy baja con la variable objetivo 'status'. Además, se realizaron modificaciones adicionales en el preprocesamiento de los datos para abordar posibles inconsistencias detectadas en el reporte de perfil, asegurando así la calidad y coherencia del análisis. Estos cambios fueron fundamentales para optimizar la selección de características y mejorar el desempeño del modelo de Machine Learning en la detección de posibles intentos de phishing.

Exploración de datos

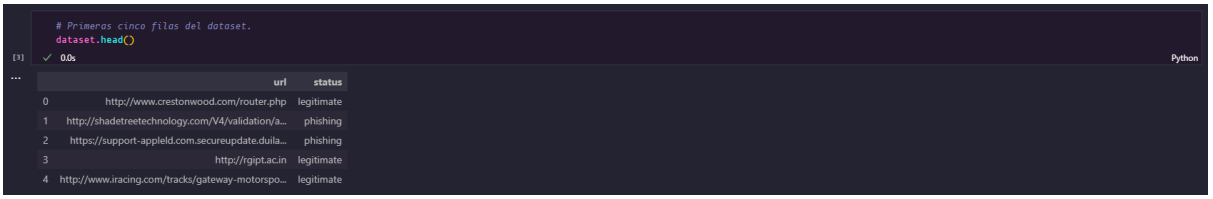
Existen algunas preguntas importantes antes de comenzar el procedimiento de la implementación de los dos modelos.

1. Cargue el dataset en un dataframe de pandas, muestre un ejemplo de cinco observaciones.

Lo primero que debemos realizar, como en todo procedimiento de Ingeniería de Características y procedimientos relacionados a Ingeniería de Datos antes de comenzar con el entrenamiento de un nuevo modelo de aprendizaje de máquina, es familiarizarnos con los datos que nos han sido proporcionados. Antes de cualquier otra cosa, necesitamos hacer uso de la librería de pandas para poder leer el dataset que nos ha sido proporcionado.

Con pandas a nuestra disposición, podemos proceder a leer el dataset que tenemos qué utilizar para el laboratorio. Esto lo podemos realizar utilizando la función `read_csv`, que nos permite leer un archivo con extensión .csv y convertirlo a un DataFrame de pandas, mucho más fácil de manipular.

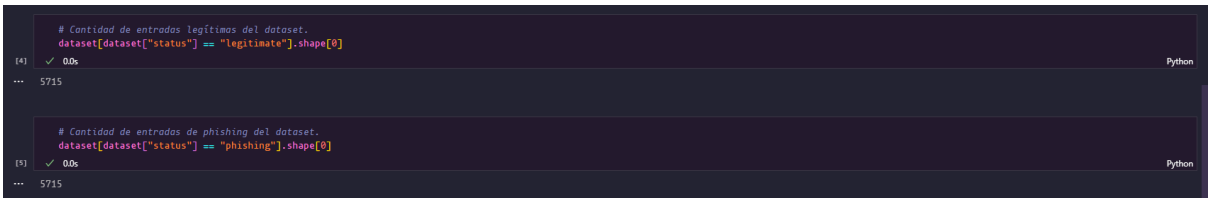
Algo que se solicita en las instrucciones del laboratorio es poder realizar una observación rápida de un ejemplo de cinco entradas del dataset. Esto lo podemos realizar utilizando la función `head`, que nos retorna las primeras cinco entradas de un DataFrame de pandas.



2. Muestre la cantidad de observaciones etiquetadas en la columna status como “legit” y como “phishing”. ¿Está balanceado el dataset?

Posteriormente necesitamos verificar si el dataset se encuentra balanceado, ya que esta característica es particularmente útil al momento de entrenar un modelo que sea capaz de clasificar entre dos, valga la redundancia, clases diferentes. Esto lo podemos observar, en primer lugar, averiguando cuántas filas tiene nuestro dataset. Esto se puede realizar observando la propiedad `shape` presente en todos los DataFrames de pandas.

Sabiendo que contamos con 11,430 filas en el dataset, podemos observar cuántas de estas filas corresponden a entradas legítimas o entradas de phishing. A continuación se realiza el query para observar cuántas filas hay de cada una de las clases.



Podemos notar cómo el dataset se encuentra perfectamente balanceado, ya que tenemos 5,715 entradas de cada clase para hacer de cada una el 50% del dataset. Al tener esta característica, nos evitamos la gran mayoría de procedimientos manuales para balancear el dataset antes de entrenar al modelo.

Derivación de características

Posteriormente a la lectura de los artículos proporcionados, se respondieron las siguientes preguntas.

- ¿Qué ventajas tiene el análisis de una URL contra el análisis de otros datos, cómo el tiempo de vida del dominio, o las características de la página Web?

El análisis de una URL es útil por varias razones. Primero, te da información directa sobre a dónde te llevará el enlace, lo cual es esencial para verificar si es legítimo o si podría ser un intento de phishing. Además, la estructura de la URL puede darte pistas sobre el contenido de la página, lo que ayuda a saber si es relevante para ciertos temas o palabras clave.

Desde el punto de vista de la seguridad, el análisis de URL es importante para identificar posibles riesgos, como patrones asociados con sitios web maliciosos o si la página está en listas negras de seguridad. También ayuda a detectar redirecciones y rutas, lo que es relevante tanto para la seguridad como para entender cómo está organizado el sitio web.

La validación de formato es otra ventaja del análisis de URL, asegurando que esté bien escrita y siga las reglas estándar, evitando problemas al interpretarla o al navegar hacia ella. Aunque también es crucial analizar datos como el tiempo de vida del dominio o las características de la página web, el análisis de URL destaca por brindar información inmediata y directa sobre la naturaleza del enlace. Para una evaluación completa, aún es importante combinar diversas fuentes de información.

- ¿Qué características de una URL son más prometedoras para la detección de phishing?

Según la investigación realizada, las características a utilizar y a implementar son las siguientes.

1. Largo de la URL
2. Largo del pathname
3. Largo del domain name
4. Largo del filename
5. Número de vocales del domain
6. Número de puntos en el domain name
7. Número de slashes de la URL
8. Número de puntos de la URL
9. Número de dashes de la URL
10. Entropía de la URL
11. Entropía del dominio
12. Número de puntos y comas de la URL
13. Número de porcentajes de la URL
14. Número de hashtags en la URL
15. Número de guiones bajos de la URL

Cada una de las funciones se encuentran implementadas en el código para posteriormente ser utilizadas en el mismo.

Discusión

1. ¿Cuál es el impacto de clasificar un sitio legítimo como phishing?

Clasificar un sitio legítimo como phishing puede tener un impacto significativo en la confianza de los usuarios y la reputación de la empresa o servicio asociado con ese sitio. Los usuarios pueden perder la fe en la seguridad del servicio y optar por evitarlo por completo, lo que podría resultar en la pérdida de clientes y dañar la imagen de la empresa. Además, si se proporcionan datos personales o confidenciales en el sitio legítimo, existe el riesgo de que esa información se vea comprometida, lo que podría tener consecuencias graves para los usuarios afectados y la empresa responsable del sitio.

2. ¿Cuál es el impacto de clasificar un sitio de phishing como legítimo?

Por otro lado, clasificar un sitio de phishing como legítimo también puede tener consecuencias negativas. Los usuarios podrían ser engañados para proporcionar información confidencial, como contraseñas o información financiera, lo que podría conducir a robo de identidad, fraude financiero u otros tipos de ataques cibernéticos. Esto puede resultar en pérdidas financieras para los usuarios afectados y dañar la reputación y la confianza en la empresa o servicio que fue falsamente identificado como legítimo.

3. En base a las respuestas anteriores, ¿Qué métrica elegiría para comparar modelos similares de clasificación de phishing?

Dada la importancia de minimizar tanto los falsos positivos (sitios legítimos clasificados como phishing) como los falsos negativos (sitios de phishing clasificados como legítimos), la métrica más relevante para comparar modelos de clasificación de phishing sería el F1-score. El F1-score es una medida que combina precisión y recall, lo que lo hace adecuado para evaluar el equilibrio entre la capacidad de un modelo para identificar correctamente tanto los sitios de phishing como los sitios legítimos.

4. ¿Qué modelo funcionó mejor para la clasificación de phishing? ¿Por qué?

Para determinar cuál de los modelos funcionó mejor en la clasificación de phishing, necesitaríamos comparar sus métricas de rendimiento, incluido el F1-score, accuracy, precision, recall y AUC. Basándonos en las métricas proporcionadas para ambos modelos (Random Forest y Red Neuronal), podemos observar que el modelo de Random Forest parece haber obtenido un F1-score más alto, lo que indica un mejor equilibrio entre precisión y recall. Sin embargo, sería prudente analizar más a fondo las características específicas de cada modelo y considerar otros factores, como la facilidad de interpretación y el costo computacional, antes de tomar una decisión final.

5. Una empresa desea utilizar su mejor modelo, debido a que sus empleados sufren constantes ataques de phishing mediante e-mail. La empresa estima que, de un total de 50,000 emails, un 15% son phishing. ¿Qué cantidad de alarmas generaría su modelo? ¿Cuántas positivas y cuantas negativas? ¿Funciona el modelo para el BR propuesto? En caso negativo, ¿qué se podría hacer para reducir la cantidad de falsas alarmas?

Si la empresa decide utilizar su mejor modelo para detectar correos electrónicos de phishing, podemos calcular la cantidad de alarmas que generaría en un conjunto de 50,000 correos electrónicos, asumiendo que el 15% son phishing. Multiplicando el total de correos electrónicos por el porcentaje de phishing, obtenemos 7,500 correos electrónicos de phishing. Si el modelo tiene un recall del 64.61% (como en el caso de la Red Neuronal), esto significaría que detectaría correctamente aproximadamente el 64.61% de los correos electrónicos de phishing, es decir, alrededor de 4,846 correos electrónicos de phishing. Sin embargo, también generaría falsas alarmas al clasificar incorrectamente algunos correos electrónicos legítimos como phishing. Para evaluar si el modelo funciona para el Business Requirement (BR) propuesto, necesitaríamos comparar la cantidad de falsas alarmas con el umbral de tolerancia de la empresa para determinar si es aceptable o si se necesitan ajustes en el modelo para reducir la cantidad de falsas alarmas. Esto podría incluir ajustar el umbral de decisión del modelo o implementar técnicas de post-procesamiento para mejorar la precisión y reducir las falsas alarmas.

Repositorio de Github

El repositorio con todo el código fuente es https://github.com/SantiagoTaracena/sds-lab-01, en el mismo se puede observar todo el código implementado. Muchas preguntas sobre el rendimiento de los modelos también se encuentran allí.

Bibliografía

- Al-Riyami, S. S., Al-Jaroodi, J., & Benkhelifa, E. (2020). Phishing detection techniques: A survey. IEEE Access, 8, 26793-26816.
- Bhowmick, A., & Bhattacharyya, S. (2020). A comprehensive survey on phishing attack detection techniques. IEEE Communications Surveys & Tutorials, 23(2), 1119-1156.
- Jagtap, S., & Gavhane, A. (2021). A survey on phishing detection techniques using machine learning and deep learning. International Journal of Recent Technology and Engineering (IJRTE), 10(2), 114-120.