eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
VITORIA-GASTEIZKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE VITORIA-GASTEIZ

DETECCIÓN BASADA EN HTML DE PAGINAS WEB DE PHISHING MEDIANTE MACHINE LEARNING

Ciberseguridad

4.° Curso

Ibai Ruiz de Austri Lamas

Resumen

En este artículo se exploran distintos algoritmos de aprendizaje automático (Machine Learning) como herramientas de detección para paginas fraudulentas de phishing. Los modelos desarrollados con estos modelos se han entrenado mediante vectores de caracteristicas extraídos mediante BeautifulSoup de un dataset de páginas públicamente disponibles. Para el dataset de páginas legitimas se ha usado una sección de la base de datos proporcionada por Tranco – List mientras que para el dataset de páginas fraudulentas se ha hecho uso de un registro proporcionado por Phishtank. Tras el entrenamiento de los modelos se ha realizado una prueba en la que se han medido determinadas métricas de rendimiento tras la cual se ha hecho una comparación de los algoritmos empleados para determinar cuál es el más efectivo actualmente.

1. Introducción

El phising es actualmente una de las principales amenazas cibernéticas actuales, este se basa en la aplicación de métodos de ingeniería social para robar datos confidenciales tales como credenciales, información de tarjetas de crédito, datos personales o dinero, no solo es una amenaza para particulares sino también para empresas en forma de espionaje industrial.

Un ataque de phishing corriente comienza mediante el envío de un correo electrónico, SMS o algún otro tipo de mensaje a la víctima, usualmente con un enlace a un sitio web fraudulento. La victima entra al sitio web suponiendo que es legítimo e ingresa sus datos, ya sea mediante formularios o mediante cookies, lo que permite al atacante capturar la información registrada.

El phishing es especialmente peligroso debido a que es muy común, afecta a todos los usuarios de Internet, atacando a usuarios finales y no requiere conocimientos técnicos por parte de los atacantes al ser una amenaza basada en ingeniería social.

Las soluciones propuestas están típicamente basadas en concienciar al usuario de estas amenazas, pero esta solución no es infalible y requiere de constante monitorización, una de las otras soluciones que se plantea y que es la que se ha trabajado en este artículo es la de la detección y alerta automática de phishing.

1.1. Método de detección

El método de detección escogido es el aprendizaje automático o Machine Learning, este método consiste en generar un algoritmo que pueda clasificar una página web como legitima o como phishing al analizar las caracteristicas de la misma y determinar si estas son propias de una página legitima o no. Para esto se realiza un entrenamiento previo de este algoritmo analizando un conjunto de sitios web cuya legitimidad ya es conocida.

Hay múltiples métodos de analizar estas páginas web, entre ellas el análisis de URL, en el que únicamente se analiza la estructura y las propiedades de la URL para determinar la legitimidad de la página. El análisis de contenido, en el que se le da importancia al contenido interno de la página, especialmente al código HTML, las etiquetas, los formularios y otros elementos estructurales. El análisis de similitud visual, en los que se emplean técnicas de visión de computador para comparar el aspecto de la página sospechosa con la de sitios legítimos. También hay análisis híbridos en los que se realizan más de uno de los anteriores métodos para mayor exactitud.

En este artículo se ha empleado por un método de análisis basado en contenido, concretamente del código HTML. El criterio para esta elección ha sido la dificultad de falsificar los datos a analizar como las estructuras, los formularios, scripts maliciosos o patrones de ataque, en caso de usar un método basado en URL o en similitud visual es más fácil de falsificar. Otro criterio ha sido la accesibilidad de trabajar con HTML, esto nos permite utilizar bibliotecas para extraer y procesar las caracteristicas pertinentes y facilita la construcción del conjunto de datos empleado en el modelado del aprendizaje automático.

1.2. Machine learning

El aprendizaje automático o Machine Learning es una rama de la Inteligencia Artificial que permite a los sistemas mejorar a partir de su funcionamiento sin ser esta mejora explícitamente programada. En lugar de seguir direcciones predefinidas estos algoritmos detectan patrones en la información ya procesada para predecir y tomar decisiones con respecto a la nueva información.

Existen tres tipos principales de aprendizaje automático, el aprendizaje supervisado, en el que el algoritmo aprende a partir de un conjunto de datos etiquetados, en este contexto un dato etiquetado representa una página web cuya legitimidad o ausencia de la misma es conocida y esto es a su vez interpretado por el modelo para poder usarlo como ejemplo antes de clasificar instancias de nueva información como legitimas o fraudulentas.

También está el aprendizaje no supervisado, en el que el modelo intenta buscar patrones, estructuras o agrupaciones ocultas en las caracteristicas del conjunto de datos sin conocer previamente la categorización. Por último, el aprendizaje por refuerzo,

en el que el modelo aprende mediante la interacción con un entorno, recibiendo recompensas o penalizaciones en función de la exactitud de sus predicciones, lo que lleva a la optimización conforme pasa el tiempo.

En este artículo se ha optado por el aprendizaje supervisado debido a que los ejemplos de sitios web etiquetados como legítimos o fraudulentos son de fácil acceso. Además, emplear sistemas de detección basados en Machine Learning permite adaptarse automáticamente a las nuevas técnicas de phishing que aparecen con el paso del tiempo y que métodos más tradicionales como son las listas negras no podrían tratar.

2. Trabajos relacionados

El phishing es una amenaza en la ciberseguridad tan extendida y que afecta a tantos usuarios que su detección y prevención se estudia y analiza con frecuencia, Rasha Zieni et al. Encontraron que un 13,6 % de todos los estudios en los que mencionaba el phishing, estaban dedicados a su detección y prevención, alrededor de 600 estudios en total **[1]**.

De entre los métodos más usados para determinar que una página sea fraudulenta están las listas. Estas son rápidas de crear y simples de mantener además de muy exactas en sus capacidades de detección, el problema principal es que son por naturaleza reactivas, por lo que fallan en la detección de un ataque la primera vez que este se realiza **[2]**.

Por otro lado, la similaridad de página, como método de detección, resulta bastante más complejo de implementar y usa un conjunto de técnicas, entre ellas Scale-Invariant Feature Trasform, Speeded-Up Robust Features o Contrast Context Histogram. Estas técnicas se basan en identificar puntos clave o firmas de las páginas fraudulentas y la distancia Euclidiana entre estos para poder dar su veredicto. Aunque también se han empleado para identificar el uso de logos legítimos en una página **[3]**. El principal problema de este método es que las páginas de phishing, por su naturaleza, ya intentan replicar de la forma más fiel posible sitios legítimos, por lo que estos ya emplean una variedad de técnicas de forma deliberada para dar la imagen de una página web fiable, lo cual influye negativamente en la efectividad de este método, que ya de por si es un método lento en comparación con otros.

3. Metodología

Se ha desarrollado una aplicación de detección de páginas web de phishing en Streamlit mediante el entrenamiento de un modelo aprendizaje automático, específicamente aprendizaje supervisado.

3.1. Dataset

El uso de aprendizaje supervisado en el proyecto implica la necesidad de set de datos etiquetados para el entrenamiento del modelo. Tanto de páginas web verificadas como legitimas como de páginas web cuya naturaleza maliciosa está confirmada. Esos datos son fácilmente accesibles y se han procurado de *Phishtank* y *Tranco List* para las URLs de páginas maliciosas y legitimas respectivamente **[4, 5]**.

La extracción del contenido HTML y su subsecuente adaptación en forma de vectores numéricos para el entrenamiento del modelo se ha realizado empleando la biblioteca de scraping BeautifulSoup.

3.2. Algoritmos

En el desarrollo de la aplicación se han entrenado y empleado los siguientes algoritmos para la determinación del modelo optimo.

Empezando por *Gaussian Naive Bayes* (GNB), este modelo asume que se cumple una distribución normal de los datos analizados y mediante el Teorema de Bayes determina la legitimidad del sitio web por probabilidad. *Decision Tree* (DT), este modelo funciona como un diagrama de flujo con preguntas sucesivas. *Random Forest* (RF) es un modelo basado en múltiples arboles de decisión que mejora la generalización mediante agregación. *Support Vector Machine* (SVM), este modelo busca la mejor división entre sitios legítimos y fraudulentos mediante una "línea divisora" teórica, optimizando así el margen de seguridad entre las categorías. *AdaBoost*, combina los clasificadores simples y da más peso a los ejemplos más difíciles de clasificar, aprende de sus errores previos tras cada iteración. *K-Nearest Neighbors* (KNN), clasifica las nuevas paginas comparándolas con las 'K' número de sitios con las caracteristicas más parecidas, si la mayoria son fraudulentas las clasifica de la misma manera. *Logistic Regression* (LR), calcula la probabilidad de que un sitio sea phishing mediante una combinación ponderada de sus caracteristicas.

3.3 Métricas de rendimiento

Para determinar la efectividad de los modelos a la hora de clasificar nuevos sitios web es importante definir ciertos parámetros y conceptos. Uno de estos es la matriz de confusión, al analizar una página web cuya legitimidad es desconocida existen cuatro casos que pueden darse, tal y como se ve en la **Tabla 1**.

	Veredicto – Legitimo	**Veredicto - Phishing**
Realidad - Legitimo	Verdadero negativo (TN)	Falso positivo (FP)
Realidad - Phishing	Falso negativo (FN)	Verdadero positivo (TP)

Tabla 1. Matriz de confusión.

En esta tabla se representan los posibles casos que pueden darse cuando el modelo da un veredicto con respecto a la legitimidad de una página sin categorizar, los casos verdaderos son aquellos en los que coincide el veredicto del modelo con la realidad y son aquellos que se busca maximizar, pero en los casos falsos ocurre algo que merece la pena recalcar. Mientras que un falso positivo puede suponer una molestia para el usuario final, es un exceso de seguridad que no causa daños reales, pero un falso negativo supone poner en peligro al usuario y comprometer sus datos. En consecuencia, es muy preferible tener falsos positivos que falsos negativos y al aplicar estos métodos de detección en la industria se debería buscar como máxima prioridad reducir los falsos negativos incluso a costa de generar falsos positivos.

Para poder evaluar y comparar los algoritmos entre sí se van a definir las siguientes métricas.

Exactitud (Accuracy): Porcentaje total de aciertos, tal que. $\frac{N.º\ Total\ Aciertos}{N.º\ Total\ \mathrm{Predicciones}}$

Precisión (Precision): El porcentaje de acierto solamente cuando el modelo predice una página como phishing, tal que. $\frac{TP}{TP + FP}$

Sensibilidad (Recall): El número de casos de phishing que son correctamente detectados, tal que. $\frac{TP}{TP + FN}$

La sensibilidad es una métrica de especial interés, pero para determinar el mejor algoritmo se buscará un equilibrio entre las tres métricas.

4. Resultados experimentales

Los resultados del entrenamiento de los modelos y el posterior testeo de su efectividad se han representado gráficamente tal y como se muestra en la **Figura 1**.

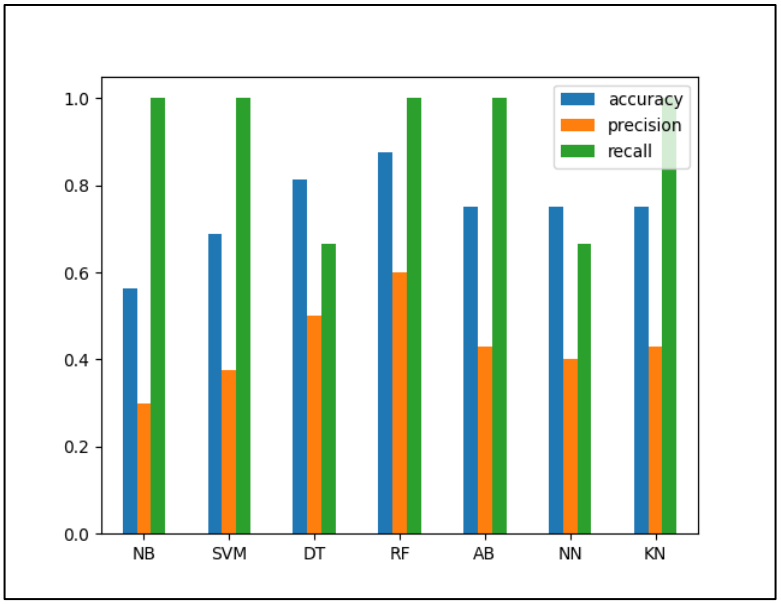

***Figura 1**. Representación gráfica de las métricas de rendimiento de los modelos.*

Como se puede apreciar la mayoría de los modelos cuentan con una alta sensibilidad, pero una baja precisión, lo que indica que tienden a dar falsos positivos con frecuencia. De entre los modelos entrenados se ha determinado que el algoritmo RF es el más efectivo para este contexto debido a que tiene una muy alta sensibilidad y aún mantiene un buen equilibrio de precisión y exactitud a diferencia de los otros modelos. Con 0,9; 0,6 y 1 para la exactitud, precisión y sensibilidad respectivamente, este modelo supera a todos los demás en las métricas de rendimiento impuestas.

Aunque estos hayan sido los resultados obtenidos cabe recalcar nuevamente que las técnicas de phishing están en constante cambio y las páginas identificadas como phishing empleadas para entrenar estos modelos tienden a ser eliminadas por lo que es posible que estos resultados no sean replicables en el futuro cercano y otro modelo resulte más efectivo para identificar el phishing del futuro.

5. Conclusiones

Tras en entrenamiento y el testeo se ha determinado que el modelo Random Forest es el más efectivo y exacto a la hora de predecir si una URL determinada es legítima o es phishing. Emplear Machine Learning para crear modelos predictivos que determinen la legitimidad de una URL soluciona en parte el problema de las listas negras, las cuales serían las soluciones tradicionales, pero al ser estáticas quedan rápidamente obsoletas. A pesar de esto, aunque el modelo Random Forest haya mostrado el mayor

rendimiento para el dataset con el que se ha entrenado las técnicas de phishing están constantemente cambiando y es posible que en el futuro cercano otro modelo sea más efectivo en detectar esas nuevas técnicas.

6. Bibliografía

[1] R. Zieni, L. Massari and M. Carla Calzarossa, 2023, ”Phishing or Not Phishing? A survey on the Detection of Phishing Websites”, *IEEE Access*, Feb. 2023, pp. 18499-18519.

[2] S. Bell and P. Komisarczuk, “An analysis of phishing blacklists: Google Safe Browsing, OpenPhish, and PhishTank,” in *Proc. Australas. Comput. Sci. Week Multiconf.*, Feb. 2020, pp. 1–11.

[3] S. Afroz and R. Greenstadt, “PhishZoo: Detecting phishing websites by looking at them,” in *Proc. IEEE 5th Int. Conf. Semantic Comput.*, Sep. 2011, pp. 368–375.

[4] Cisco Talos Intelligence Group, ”PhishTank”, Web: https://phishtank.org/index.php (Accedido el 20 de diciembre de 2025).

[5] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński and Wouter Joosen, "Tranco-List", Dec. 2025 Web: https://tranco-list.eu/ (Accedido el 20 de diciembre de 2025).