Detección de sitios web de phishing usando técnicas de aprendizaje de máquina

Michael Steven Ruiz Palacio
Emmanuel Bustamante Valbuena
Jackson Leonardo Rivera Usuga
Universidad de Antioquia
Departamento de Ingeniería de Sistemas

***Resumen*—This proyecto busca desarrollar un sistema de predicción basado en técnicas de aprendizaje automático that permita detectar sitios web maliciosos (phishing) a partir de características extraidas de URLs and su contenido. Se explora el conjunto de datos de Kaggle titulado *Phishing Dataset for Machine Learning*, se realiza un análisis del problema, una revisión del estado del arte, y se entrenan varios modelos de clasificación.**

ÍNDICE

***Index Terms*—Phishing, Aprendizaje Automático, Clasificación Binaria, XGBoost, Dataset.**

I. Introducción

El **phishing** es una de las principales amenazas a la seguridad informática: páginas web fraudulentas que imitan portales legtimos para robar credenciales, datos personales o financieros. Dada la velocidad y creatividad con las que los atacantes disean nuevos sitios de phishing, las técnicas tradicionales basadas en reglas fijas (como listas negras de URLs o detección de palabras clave) se quedan cortas.

Contexto del problema

- **Volumen de datos**: Existen miles of URLs creadas cada hora, con variaciones en dominios, estructuras HTML y textos.
- **Evolución constante**: Los atacantes adaptan cadenas de texto, imágenes y patrones de enlace para evadir filtros.
- **Alcance global**: Un sitio de phishing puede apuntar a múltiples idiomas y regiones, lo que exige un modelo flexible que generalice.

II. Descripción del Dataset

II-A. Tamaño y Composición

El conjunto de datos contiene:

- **Instancias:** 10 000 URLs etiquetadas.
- **Atributos:** 50 características numéricas, binarias y categóricas extraídas de cada URL.

II-B. Datos Faltantes e Imputación

El dataset en cuestión no tiene datos faltantes por lo tanto no hizo falta realizar imputación de datos.

II-C. Codificación y Escalado

En el dataset se encontraron diferentes tipos de datos que fueron caracterizados de la siguiente forma.

- Variables binarias: 0/1.
- Discretas: one-hot encoding.
- continuas: Min–Max Scaling en [0,1].

II-D. Variables y Significado

A continuación se presenta la tabla completa de características del dataset:

Cuadro I
DESCRIPCIÓN DE LAS 50 CARACTERÍSTICAS DEL DATASET DE PHISHING

Características del Dataset			
#	**Característica**	**Tipo**	**Descripción**
1	NumDots	Discreto	Número de puntos en la URL
2	SubdomainLevel	Discreto	Nivel de subdominio en la URL
3	PathLevel	Discreto	Profundidad del path en la URL
4	UrlLength	Discreto	Longitud total de caracteres en la URL
5	NumDash	Discreto	Número de guiones (-) en la URL
6	NumDashInHostname	Discreto	Número de guiones (-) en el hostname
7	AtSymbol	Binario	Presencia de @ en la URL
8	TildeSymbol	Binario	Presencia de $\sim$ en la URL
9	NumUnderscore	Discreto	Número de guiones bajos (_)
10	NumPercent	Discreto	Número de símbolos de porcentaje ($\%$)
11	NumQueryComponents	Discreto	Número de parámetros en la query
12	NumAmpersand	Discreto	Número de símbolos &
13	NumHash	Discreto	Número de símbolos $\#$
14	NumNumericChars	Discreto	Cantidad de caracteres numéricos
15	NoHttps	Binario	Ausencia de HTTPS en la URL
16	RandomString	Binario	Presencia de cadenas aleatorias
17	IpAddress	Binario	Uso de dirección IP en el hostname
18	DomainInSubdomains	Binario	TLD/ccTLD usado en subdominio
19	DomainInPaths	Binario	TLD/ccTLD usado en el path
20	HttpsInHostname	Binario	HTTPS ofuscado en hostname
21	HostnameLength	Discreto	Longitud del hostname
22	PathLength	Discreto	Longitud del path
23	QueryLength	Discreto	Longitud de la query
24	DoubleSlashInPath	Binario	Presencia de // en el path
25	NumSensitiveWords	Discreto	Palabras sensibles (login, account, etc.)
26	EmbeddedBrandName	Binario	Marca incrustada en subdominio/path
27	PctExtHyperlinks	Continuo	Porcentaje de hipervínculos externos
28	PctExtResourceUrls	Continuo	Porcentaje de URLs externas en recursos
29	ExtFavicon	Binario	Favicon cargado desde dominio externo
30	InsecureForms	Binario	Formularios sin HTTPS
31	RelativeFormAction	Binario	Acción de formulario relativa
32	ExtFormAction	Binario	Acción de formulario en dominio externo
33	AbnormalFormAction	Categórico	Acciones anormales (#, about:blank, etc.)
34	PctNullSelfRedirectHyperlinks	Continuo	$\%$ de hipervínculos vacíos/redirigidos
35	FrequentDomainNameMismatch	Binario	Dominio principal $\neq$ dominio en HTML
36	FakeLinkInStatusBar	Binario	URL falsa en barra de estado
37	RightClickDisabled	Binario	Click derecho deshabilitado
38	PopUpWindow	Binario	Ventanas emergentes
39	SubmitInfoToEmail	Binario	Uso de mailto en formularios
40	IframeOrFrame	Binario	Uso de iframe/frame
41	MissingTitle	Binario	Título vacío
42	ImagesOnlyInForm	Binario	Formulario solo con imágenes
43	SubdomainLevelRT	Categórico	Nivel de subdominio (umbrales reforzados)
44	UrlLengthRT	Categórico	Longitud URL con reglas aplicadas
45	PctExtResourceUrlsRT	Categórico	$\%$ URLs externas (categorizado)
46	AbnormalExtFormActionR	Categórico	Acción de formulario externa anormal
47	ExtMetaScriptLinkRT	Categórico	$\%$ de tags externos (meta/script/link)
48	PctExtNullSelfRedirect-HyperlinksRT	Categórico	$\%$ hipervínculos externos/anómalos

II-E. Paradigma de Aprendizaje Supervisado

Dado que contamos con etiquetas binarias para cada URL, se adoptó el aprendizaje supervisado. Los modelos evaluados incluyen:

- Random Forest: robusto ante ruido.
- Gradient Boosting (XGBoost): rendimiento superior en clasificación de phishing.
- Red Neuronal Ligera: captura relaciones complejas.

III. Estado del arte

Diversos estudios han abordado la detección de sitios de phishing usando machine learning. Algunos de ellos se detallan a continuación:

- **Modelo de Conjunto de Apilamiento**: La metodología consta de tres fases principales: las fases de entrenamiento, clasificación y prueba. En la fase de entrenamiento, randomForest, AdaBoost, XGBoost, Bagging, GradientBoost y LightGBM fueron entrenados sin optimización. Los clasificadores mencionados anteriormente se optimizaron utilizando el algoritmo genético, el cual simula la evolución natural mediante los siguientes pasos: inicialización de una población de soluciones candidatas, evaluación mediante una función de aptitud, selección de los mejores individuos, cruces y mutaciones para generar nuevas soluciones, y repetición del proceso hasta converger a un óptimo o alcanzar un número máximo de iteraciones.
  Para evaluar el rendimiento del modelo de conjunto propuesto, se utilizaron las siguientes medidas de rendimiento: precisión de clasificación, precisión, recuperación (la tasa de detección), puntuación F1, tasa de falsos positivos (FPR) y tasa de falsos negativos (FNR); todos los experimentos realizados, incluidos los clasificadores optimizados y no optimizados, se validaron utilizando una validación cruzada de 10 veces. La precisión obtenida alcanzó el 97,16 %.

- **Detección de sitios web de phishing**: El estudio empleó técnicas de aprendizaje supervisado utilizando las siguientes arquitecturas de redes neuronales: LSTM (Long Short-Term Memory), Redes Neuronales Profundas Totalmente Conectadas (FCnet), Redes Neuronales Convolucionales (CNN).

  Las técnicas de optimización utilizadas fueron:
  - Búsqueda en cuadrícula (Grid Search): Exploración exhaustiva de combinaciones de hiperparámetros predefinidos.
  - Algoritmo Genético (GA): Método de optimización inspirado en la evolución natural para encontrar combinaciones óptimas de hiperparámetros.
  - Se utilizaron las siguientes métricas para evaluar el rendimiento de los modelos: precisión REacall F!-score, FPR, curvas ROC.

  El modelo LSTM utilizando características combinadas (LSTM-all) logró la mayor precisión con un 97.37 % y un F1-Score de 0.974 en el Tan-dataset además de un tiempo de entrenamiento de 1 minuto.

- **Aprendizaje automático de Ensamble**: Los autores utilizaron un enfoque de conjunto basado en votación (Voting), y lo compararon con seis algoritmos individuales de machine learning: Decision Tree, Random Forest, Gradient Boosting, XGBoost, AdaBoost y Multi-Layer Perceptron. Además, aplicaron una técnica de normalización a los datos antes del entrenamiento de los modelos. Para evaluar el rendimiento del sistema, se utilizaron dos conjuntos de datos relacionados con phishing. Los autores no detallan la metodología exacta de validación (como la división de los datos o el uso de validación cruzada), sí menciona que se emplearon cuatro métricas para medir el desempeño: Accuracy, Precision, Recall y F1-score.
  En los resultados obtenidos con el primer conjunto de datos, el modelo basado en Voting fue el que mostró el mejor rendimiento. Por otro lado, en el segundo conjunto, todos los algoritmos evaluados obtuvieron resultados idénticos en todas las métricas.

IV. Conclusiones y Trabajo Futuro

Los próximos pasos incluyen:

1. Implementación del preprocesado automático.
2. Búsqueda de hiperparámetros con GridSearchCV.
3. Evaluación final y comparación de modelos.

Referencias