Detección de sitios web de phishing aplicando algoritmos de clasificación

Fabian Farez
fabian.farez@ucuenca.edu.ec

Abstract:**El phishing es un ataque que afecta a la seguridad de un usuario en una página web. Este tipo de ataque busca obtener datos privados o confidenciales de los usuarios, haciéndose pasar por un sitio confiable. Este documento está centrado en evaluar distintas técnicas de machine learning para detectar este tipo de ataques, se realizan con distintos métodos de clasificación y se comprueba qué los algoritmos ensemble son los más efectivos en la detección de páginas web phishing. Además se realiza una práctica de clustering probando diferentes conceptos para encontrar un adecuado agrupamiento de páginas web.**

I. INTRODUCCIÓN

El phishing es una manera de ciberdelincuencia que se ha vuelto tendencia en la actualidad. Se trata de un tipo de estafa en la cual se hacen pasar por una empresa o una persona y de esta manera consiguen información confidencial del usuario como contraseñas o información de cuentas bancarias. Las técnicas que utilizan para el robo de información principalmente es la manipulación de correos electrónicos, mensajes directos o el envío de urls falsas. El phishing es difícil de detectar si no se tiene los recursos necesarios para poder distinguirlos. La mayoría de ataques phishing ocurren con la recepción de un correo electrónico que incluye enlaces a sitios web que imitan a una empresa legítima además se puede dar mediante el redireccionamiento de una página legítima a una página phishing, En este contexto los ataques de phishing afectan enormemente a la personas por lo que este trabajo se centra en examinar diferentes modelos de clasificación que nos permitan detectar si una página es sospechosa, legítima o se trata de una página fraudulenta.
El objetivo de este trabajo está centrado en encontrar qué técnica de clasificación propuesta es la más eficiente para detectar ataques de phishing. Se realizaron pruebas con modelos de aprendizaje automático y se realizó una comparativa en base a métricas de rendimiento que genera cada método. Este trabajo cuenta con secciones que se describen a continuación. La primera sección presenta la introducción, la tarea y el objetivo del proyecto.En la sección 2 se presenta el estado del arte que describen trabajos similares al presentado. En la sección 3 se presenta el dataset y sus características principales, En la sección 4 se presenta la metodología utilizada tanto para clasificación y clusterización. En la sección 5 se presentan los resultados más relevantes obtenidos después de realizar clasificación y clusterización, para finalizar en la sección 6 se presentan las conclusiones del trabajo.

II. ESTADO DEL ARTE

Hasta la actualidad con el constante evolución de métodos fraudulentos también se han creado útiles herramientas como las que se presentan los autores Hota, Srivas [1] que desarrollaron un modelo de identificación de paginas phishing a fin de obtener un clasificador robusto combinando dos técnicas de clasificación basadas en árboles de decisión, CART y C4.5, el primer clasificador construye un árbol de decisión binario en base a un atributo, mientras que C4.5 es un árbol capaz de manejar variables continuas y discretas mejorando la detección de este tipo de sitios fraudulentos.
Otro tipo de herramienta que desarrollaron los autores Mao et al. [2] al detectar similitudes entre diseños de soluciones para detección de phishing crearon una herramienta de detección de phishing evaluando Máquinas de soporte vectorial para diferentes gamas y comparando resultados para diferentes valores, además se evaluaron Árboles de Decisión configurado para distintos valores de profundidad del árbol.

III. DESCRIPCIÓN DEL DATASET

El dataset usado para realizar la experimentación se obtuvo de la página web Machine learning Repository en donde existe una gran colección de bases de datos que son utilizados generalmente en el ámbito del aprendizaje automático.
El dataset presenta las siguientes características relacionadas a sitios web legítimos y de phishing: Cuenta con 1353 instancias y se dividen en 10 atributos de tipo entero. Las instancias están divididas en 548 sitios legitimos, 703 de phishing y 103 instancias sospechosas. Además cuenta con valores categóricos que los diferencian: -1 ,0, 1 indican si un sitio phishing,sitio sospechoso y legítimo respectivamente.
Las variables de este dataset se presentan a continuación.

Tabla.1. Características del Dataset

Características	Descripción
SFH	Verifica si la información ingresada en la página se transfiere a un servidor o si se transfiere a un servidor de un dominio diferente

popUpWindow	Verifica si la página genera ventanas emergentes
SSL final State	Verifica si el protocolo HTTPS de la página es de confianza o si es falso
Request_URI	Verifica si los objetos en la página web son cargados desde un dominio diferente al de la URL.
URL_of_Anchor	Verifica si los enlaces dentro de la página web apuntan a un dominio diferente al de la URL
web traffic	Verifica si el tráfico web de la página es el de una página legítima o no.
URL_Length	Verifica si la longitud de la URL
age of domain	Verifica si el tiempo de vida del dominio es de una página legítima
having_IP_Address	Verifica si la URL tiene la dirección IP
Result	Indica si la página es phishing, es sospechosa de phishing o si es legítima.

El dataset está conformado por valores enteros de 1,0,-1 que nos indican si las características pertenecen o no a una categoría de página web sospechosa, legítima o phishing.
El número de atributos asignados a cada clase nos indican un desbalance entre clases que por lo general afecta a los algoritmos en su proceso de generalización de información que perjudica a las clases minoritarias. La clase minoritaria para este ejemplo es la que corresponde a la clase de sitio web legítimos, por lo que se necesita aplicar técnicas que corrijan este desbalance y poder obtener resultados satisfactorios. Entre los métodos de balanceo de clase minoritaria para la experimentación se optó por aplicar un costo de penalización para la clase mayoritaria para tratar de corregir instancias mal clasificadas.

IV. METODOLOGÍA

La metodología usada en este trabajo se presenta a continuación.

A. **Modelos de clasificación para detección de phishing:** Primero se seleccionan los modelos de aprendizaje automático de clasificación que se usarán en el trabajo. Los modelos a probar son los siguientes: árboles de clasificación y support vector machine(SVM) que son unos de los principales y más conocidos algoritmos para clasificación. Además se utilizan algoritmos ensemble Randon Forest el Gradiente Boost y algoritmos XG BOOST que ayudan a mejorar el rendimiento de los modelos

B. **Implementación:** Escogido los modelos con los que se trabajarán se procede a cargar los datos y hacer un análisis de los mismos, su distribución y demás características importantes.

C. **Entrenamiento:** Para la fase de entrenamiento los datos obtenidos serán divididos en conjuntos para entrenamiento y prueba que serán utilizados para la implementación de cada clasificador. Para corregir el desbalance de clases en el algoritmo de los árboles de clasificación se utilizó una penalización para la clase mayoritaria. En cuanto al método SVM se realizará la prueba modificando el hyperparameter C que minimice los errores de la clase minoritaria.

D. **Análisis de resultados:** Una vez entrenados los modelos se realizará la verificación de los mismos con el conjunto de prueba. Se obtendrá valores de eficiencia del algoritmo con métricas como matriz de confusión y se realizará un análisis ROC para verificar que algoritmo presenta mejores resultados en la clasificación de los datos en este problema específico.

E. **Clusterización:** Se presentarán diferentes estrategias para encontrar un adecuado número de categorías de páginas web, y se probarán diferentes métodos de dimensionamiento para una mejor visualización de los datos. Entre los métodos de reducción de dimensionalidad propuestos están PCA y TSNE.

V. RESULTADOS y DISCUSIONES

En el análisis inicial de los datos se puede observar el desbalance de clases siendo la minoritaria la clase de sitios web sospechosos por lo que se requiere balancear las clases al momento de implementar cada método.
Para un análisis inicial de los datos y cómo afecta el desbalance de clases se presenta los resultados del algoritmo árboles de clasificación y sus métricas.

Tabla.2. Métricas

Desbalanceo	**Accuracy**	**Recall**	**F1-score**
-1	0.87	0.87	0.87
0	0.90	0.90	0.90
1	0.84	0.84	0.84
Balanceo	**Accuracy**	**Recall**	**F1-score**
-1	0.90	0.86	0.88
0	0.88	1	0.96
1	0.85	0.88	0.86

Para el análisis de los resultados obtenidos y elegir con qué métrica se obtendrá los mejores resultados, se presentan las matrices de confusión obtenidas en cada modelo. Al existir 3 clases se procede a calcular los valores TP,FP,TN,FN para la clase legítima la misma que nos servirá para realizar análisis posteriores. Se escoge realizar la matriz de confusión sobre la categoría legítima principalmente para evaluar los falsos negativos que podrían presentarse al momento de evaluar las

páginas web. En el caso de la detección de phishing, si una página se predice como pishing pero en realidad no, entonces puede causar problemas a los usuarios. En este caso, debemos centrarnos en reducir el valor de FP es decir tratar de reducir los casos cuando una página legítima se clasifica como no legítima pero en realidad sí lo es.

Tabla.3. Matriz de confusión clase Desbalanceada

Desbalanceo	1	-1
1	TP=95	FN=18
-1	FP=18	TN=140

Tabla.4.Matriz de confusión clase balanceadas

Balanceo	1	-1
1	TP=99	FN=14
-1	FP=17	TN=141

En las matrices de confusión presentadas con respecto a clasificación con árboles se observa una mejoría en la clasificación correcta de páginas legítimas. De igual manera se intenta mejorar los otros métodos de clasificación propuestos variando sus hyperparametros.

En cuanto a la evaluación de los métodos se propone la métrica de sensibilidad que es bastante útil cuando se tiene datos desbalanceados.

En base a experimentar diferentes modelos se obtuvieron los siguientes resultados. La figura 1 representa la curva ROC obtenida en los algoritmos Árbol de clasificación y SVC , también se obtiene el área bajo la curva que nos indica que tan eficiente es un algoritmo.

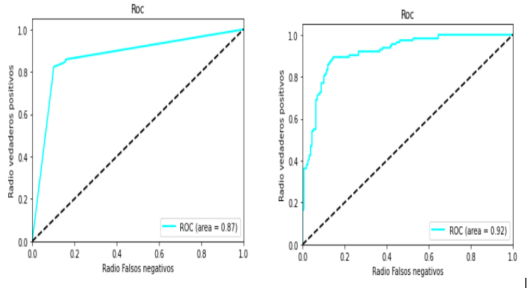

Fig.1. Curva ROC Árbol de clasificación y SVC

De la misma manera se obtiene la curva ROC para los métodos ensemble propuestos.

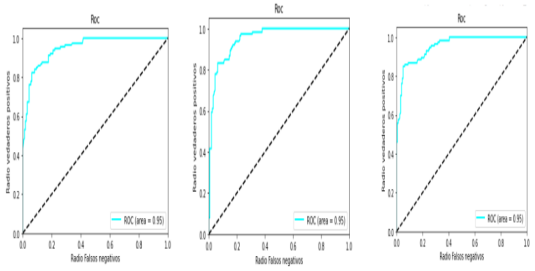

Fig.2. Curva RandomForest y Gradient Boosting XGBoost

En los gráficos se puede comprobar que como las clases comparten atributos la clasificación no puede ser perfecta siempre se tendrán errores. Así al realizar una análisis con ROC que es una de las técnicas más importantes para evaluar el rendimiento de cualquier clasificador y el área bajo la curva AUC nos da una idea de que tan bien funciona un modelo. La mejor prueba de clasificación representa la que tenga una mejor área. La forma de evaluar la curva ROC para decir que algoritmo es el mejor en base a AUC se presenta a continuación buscando predecir una página phishing .

La gráfica se presenta en dos dimensiones el eje y representa los verdaderos positivos que nos indican si la página es pishing, el eje x representa a los falsos positivos o las páginas que se clasificaron mal como phishing.A medida que el valor los falsos positivos disminuye los verdaderos positivos aumentan , de la misma manera mientras el valor de los verdaderos positivos disminuyen los falsos positivos aumentan lo que provocaría una clasificación inadecuada. Así mientras la curva se acerque más al punto máximo de verdaderos positivos significan que el modelo predice adecuadamente la división entre las clases, sin embargo si existe una curva ROC con un área menor y que se acerque más al punto medio nos indicaría que los modelos no predicen correctamente ya que no habría una diferencia clara de cómo deben clasificarse.

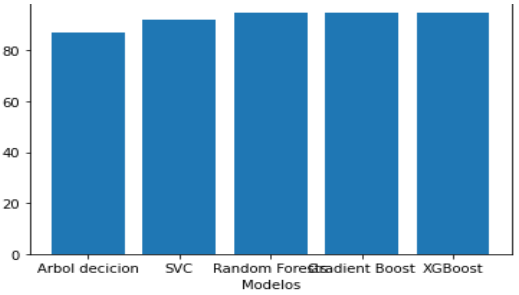

Fig.3. Mejor clasificador ROC y AUC

En base a los resultados se puede concluir que el uso de clasificadores ensemble nos dan un mejor resultado en el área bajo la curva de 95, el menor resultado para la clasificación nos dan los árboles de clasificación con un área de 87. Sin embargo el modelo XG Boost es muy sensible a cambios en sus hyperparameters y para mejorar el rendimiento del algoritmo se probó con otra configuración de hyper paramétricos, se agregó un criterio de parada máximo en 100 árboles y una profundidad de 5 para cada árbol lo que aumentó su AUC.

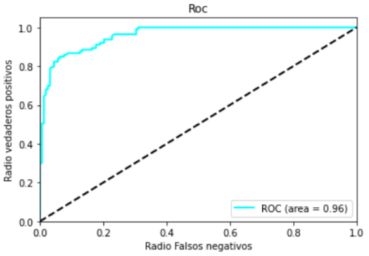

Fig.4. Mejor clasificador XGBoost

Con una ligera modificación de los hiperparametros se llega a un mejor resultado mejor en la AUC para el algoritmo XGBoost.

- **Clusterización**

Para la práctica de clusterización se escogió métodos de Clustering jerárquico aglomerativo, K Means y DBSCAN, junto con técnicas de reducción de dimensionalidad PCA y TSNE.

Primero se realizó una reducción de dimensionalidad para el dataset aplicando TSNE, este tipo de reducción de dimensionalidad hace que el cluster sea sea más preciso porque convierte los datos en un espacio de 2 dimensiones donde los puntos están en forma circular.En la figura 4 se presenta la ejecución de método para los datos sin reducción de dimensionalidad, y aplicando reducción TSNE y PCA.

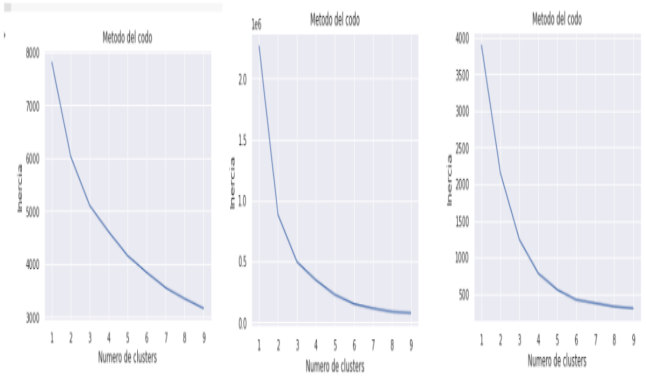

Fig.5. Número óptimo de clusters usando TSNE y PCA

En base a los resultados obtenidos de método del codo usando una reducción de dimensiones el codo se presenta en k=3 por lo que se realizaron tres pruebas con un tamaño de clusters igual a 2, 3 para ver qué tanto afecta reducir una clase o si esque existe una categoría diferente a las presentadas.

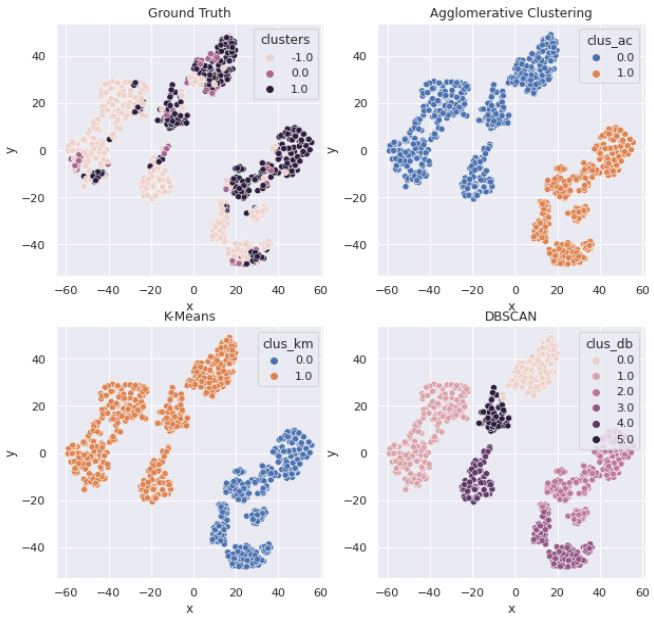

Fig.6. Clustering con k=2 y TSNE

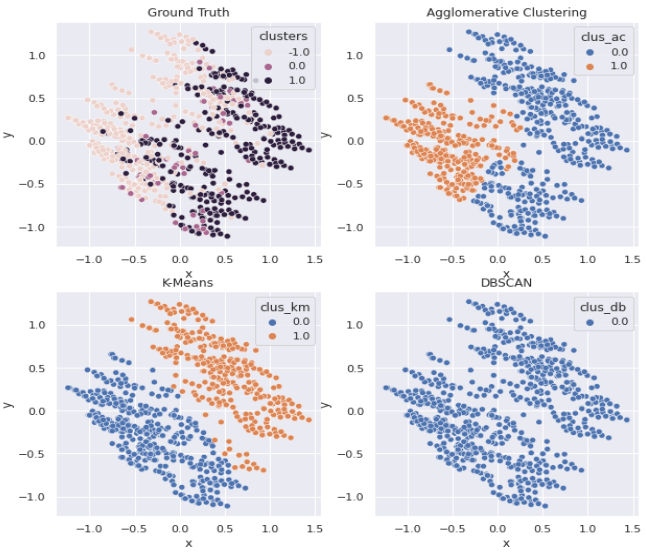

Fig.7. Clustering K=2 y PCA

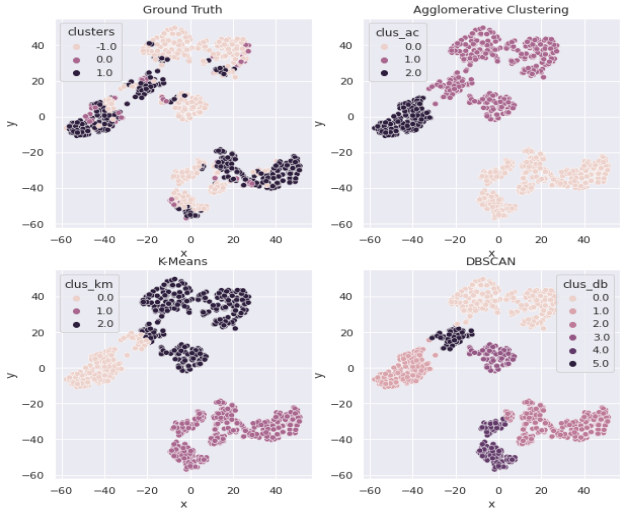

Fig.8. Clustering K=3 y TSNE

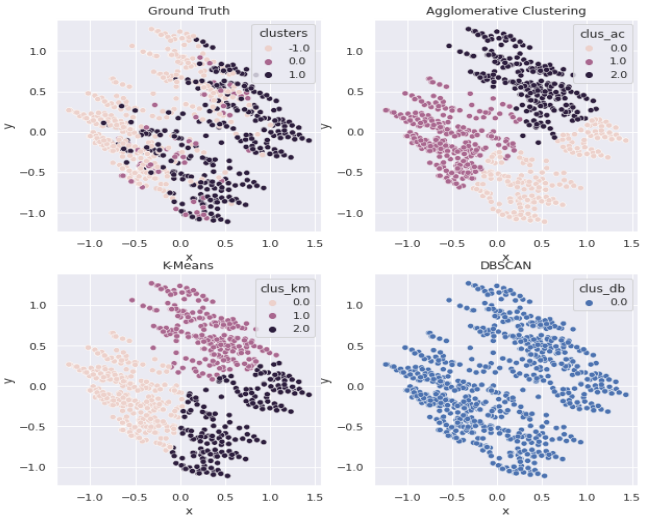

Fig.9. Clustering K=3 y PCA

Al aplicar diferentes algoritmos de clusterización podemos observar que utilizando TSNE se logra una mejor visualización y separación de grupos, para k=2 y k=3 los algoritmos k-means y Jerárquico Aglomerativo presentan una separación más uniforme de los grupos. El algoritmo DBSCAN al ser una algoritmo que agrupa en base a la

densidad de los grupos logra diferenciar 6 grupos diferentes lo que nos indica que no es el adecuado cuando se implementa con TSNE. De igual manera utilizando PCA se observa que existe una ligera separación de los grupos pero que no da brinda una visualización mejor que usando TSNE y DBSCAN solamente genera un grupo siendo el menos óptimo para este dataset. Lo que se propone es clasificar con un k=2 para diferenciar entre páginas legítimas y phishing evitando mayores inconvenientes a los usuarios a la hora de visitar una página web.

VI. CONCLUSIONES

El phishing es una de las grandes amenazas en la actualidad y que afecta enormemente a los usuarios a consecuencia de ataques, robos o estafas mediante páginas web. Sin embargo el uso de machine learning y técnicas de clasificación y agrupación nos ayudan a detectar estos sitios.

Los resultados entre estos clasificadores que se obtuvieron tienen una ligera diferencia debido a la forma de implementación que tienen y las características de los dataset con los que aprenden y luego comparan para determinar si una página es pishing o no. Después de experimentar con varios modelos se concluye que los métodos ensemble como árboles de decisión o gradient boost son los mejores en modelos en clasificar ente dataset. En investigaciones posteriores se podría implementar otros modelos, realizar un análisis para encontrar nuevas variables que nos indiquen páginas phishing, ampliar las pruebas con nuevos datasets y realizar una búsqueda más profunda para encontrar hyperparameters que nos ayuden a mejorar la efectividad de los modelos de clasificación .

VII. REFERENCIAS

[1]. Hota, H. S., Shrivas, A. K. y Hota, R. (2018). An Ensemble Model for Detecting Phishing Attack with Proposed Remove-Replace Feature Selection Technique.

[2].Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A., y Liang, Z, (2018), Detecting Phishing Websites via Aggregation Analysis of Page Layouts.

[3] Comparación de diferentes modelos de aprendizaje automático de clasificación para un conjunto de datos desequilibrado"Available:https://sitiobigdata.com/2019/12/24/comparacion-de-diferentes-modelos-de-aprendizaje-automatico-de-clasificacion-para-un-conjunto-de-datos-desequilibrado/

[4] Arnejo H, “Métodos para la mejora de predicciones en clases desbalanceadas en el estudio de bajas de clientes(CHURN)”,Available:http://eio.usc.es/pub/mte/descargas/proyectosfinmaster/proyecto_1469.pdf