Análisis de la eficacia de *Machine Learning* para la identificación de *Phishing* y *Spam*

Carlos Andrés Becerra Madera, Juan Sebastián Vargas
Estudiantes Maestría en Seguridad de la Información
Departamento de Ingeniería de Sistemas y Computación
Universidad de los Andes. Bogotá, Colombia
Diciembre 2019

1. Contexto:

La frecuencia y complejidad de ciberataques va en crecimiento constante junto a la vanguardia tecnológica [2][4][6][11], causando un crecimiento acelerado basado en la creatividad y curiosidad de los propios atacantes. Es por esta razón, que las técnicas de detección de dichos ataques también se encuentra en un auge de cambio [4], donde las personas malintencionadas siempre mejoran la efectividad de la detección de dichos ataques de la mejor manera posible.

Es en este punto crucial en donde hallamos que el auge actual de *Machine Learning* (ML) aparece como una alternativa viable y adaptable a la problemática de detección de Phishing y Spam [6]. Aunque ya existen algunos estrategia de ML para solucionar problemas concretos como la detección de malware[1][3][7][8][9][10][12], ML propone una metodología para producir algoritmos de entrenamiento fiables basados en modelos de datos históricos, y así, poder ofrecer un modelo de clasificación de incidentes, agrupando las características conocidas por el histórico y también ayuda en la clasificación de las casuísticas de las nuevas complejidades de ataques informáticos [6].

El presente trabajo surge como una oportunidad de exploración en el mundo de ML, que a su vez busca demostrar la eficacia y utilidad de su uso comparando diferentes algoritmos para contestar preguntas puntuales.

Con base de enfoque, se ha determinado a priori el análisis de los ataques Phishing y Spam dada su similitud como definición y vector de ataque. La razón principal de elección de Phishing resulta de su naturaleza evolutiva y cambiante, y también de su enfoque de robo de información. Ahora, con respecto a Spam, se ha elegido como ataque de enfoque por su similitud en el canal de propagación que es el correo electronico, solo que este último busca el engaño de los usuarios.

2. Descripción de propuesta:

La razón principal de elección de dichos vectores de ataque fue resultado de sus características similares perceptibles (medios de reproducción, asociación entre ellos, entre otros, tipos de características), pero que con ayuda de distintas estrategias de ML pretendemos poder identificarlos más efectiva y eficazmente evaluando la validez de ML en el contexto de ciberseguridad.

Para lograr nuestro propósito, hemos determinado que partiendo de una base fiable de datasets de herramientas SIEM (*Security Information and Event Manager*) abiertas (*Open Source*), crearemos una base de conocimiento fiable para entrenar nuestra propuesta de algoritmos de ML, de esta forma crearemos modelos de conocimiento que tengan una tasa aceptable de identificación de estos dos tipos de ataques informáticos.

Una vez poseamos los resultados de al menos tres estrategias de algoritmos de ML y utilizando la misma base de conocimientos, realizaremos una comparación de los resultados obtenidos ejecutando cada experimento independientemente, y luego, determinaremos así la mejor estrategia de identificación de incidentes en cada contexto aplicable.

Como objetivo final, nos enfocaremos en los resultados obtenidos de los algoritmos de ML para concretar cuales son los mejores casos de éxito dependiendo del tipo de ciberataque y de la estrategia utilizada.

3. Justificación del Problema:

El Phishing y Spam son unas de las amenazas de seguridad más comunes en los servicios de internet, donde su objetivo principal es el de robar información privada, como nombres de usuario o contraseñas o detalles de tarjetas de crédito, distribuir software malicioso, evitar el pleno uso de las capacidades computacionales, denegar servicios y en algunos casos, con fines extorsivos. Estas amenazas se propagan principalmente por medio de correos electrónicos.

En la actualidad múltiples algoritmos de ML son utilizados para identificar patrones o predecir comportamientos en múltiples contextos (la seguridad no es la excepción). A pesar de que existen varias implementaciones para los tipos de ataques antes mencionados (Phishing, Spam), ninguna logra identificar con alta precisión los distintos ataques y se puede determinar que ciertos algoritmos o la forma en que son implementados afectan su detección.

En el 2016 la actividad de phishing fue la más alta monitoreada desde el 2004, llegando a cifras de 100.000 ataques por mes. El impacto tecnológico en la industria ha provocado que el número de sitios web haya incrementado y así mismo su necesidad para el desarrollo del día a día de los usuarios. En el segundo cuatrimestre del 2019, el sistema anti-phishing de Kaspersky evitó más de 130 millones de redirecciones a sitios falsos, y en el caso concreto de Colombia, se observó que más del 11,3% de los usuarios fueron objeto de ataques de este tipo. Estas redirecciones fueron bloqueadas gracias a una combinación de validación en bases de datos y aplicación de heurísticas, principalmente aplicadas a la URL y reputación del sitio.

En cuanto a los métodos más utilizados para la identificación de phishing, se destacan: análisis de *URL*, analisis de contenido y una combinación de ambos con *whitelists* y *blacklists.* Sin embargo, ML es principalmente usado para analizar únicamente la estructura de las direcciones URL y no tienen en cuenta el contenido o la estructura de las peticiones al servidor, etc. Una consecuencia de la estrategia causa que los atacantes encuentren la manera de evitar las validaciones conocidas que se les realizan a las URL (como longitud, número de puntos, número de dígitos, dominio, entre otras), lo cual a su vez genera muchos falsos negativos y en caso de direcciones válidas específicas, falsos positivos. Es por esta razón, que actualmente las herramientas de detección automática de phishing son capaces de identificar con un máximo de 90% sitios ilegítimos, pero también clasifican de forma errada a más del 40% de sitios legítimos [16][17].

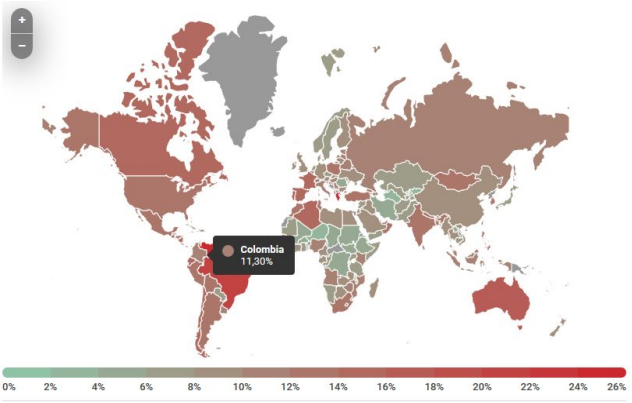


Imagen 1. Geography of phishing attacks, Q2 2019. Kaspersky

En el caso de Spam, los correos no solicitados representan aproximadamente más del 60% del tráfico global de correos electrónicos. Su impacto varía pero los casos más frecuentes incluyen: disminución de productividad de los empleados, utilización del ancho de banda, distribución de contenido con fines maliciosos (phishing, malware), entre otros.

De acuerdo a Nucleus Research, Spam genera pérdidas por empleado de alrededor de $712 cada año. Para poder clasificar este ataque, se han aplicado dos estrategias: la primera implica la creación de una base de conocimiento, lo que implica su constante actualización y depende mucho de qué tan recientes son los datos para afianzar la detección de nuevos patrones. La segunda aproximación implica el uso de ML que utilizando ciertos algoritmos, validan principalmente la existencia de frases o palabras clave que están presentes en la mayoría de correos, sin embargo, esta validación en la mayoría de los casos no es exhaustiva, lo que causa conclusiones imprecisas y erróneas.

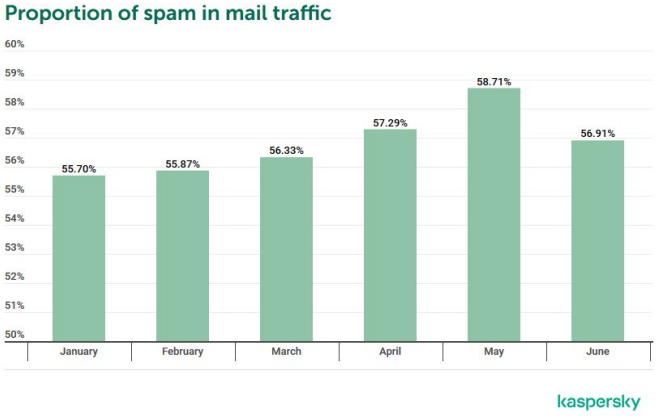


Imagen 2. Proporción de Spam en el tráfico mundial, Q1 2019-Q2 2019.

4. Objetivos:

- **General:**

Demostrar la efectividad de la utilización de técnicas de ML en la identificación de ciberataques como Phishing y Spam.

- **Específico #1:** Analizar el estado del arte en la detección de Phishing y Spam basado en algoritmos de ML.
- **Específico #2:** Proponer al menos tres estrategias de detección de Phishing y Spam utilizando una misma base de entrenamiento.
- **Específico #3:** Categorizar, comparar y concluir sobre los resultados y lecciones aprendidas derivado de los modelos de ML propuestos, especificando a su vez una métrica de efectividad.

5. Planeación detallada de actividades:

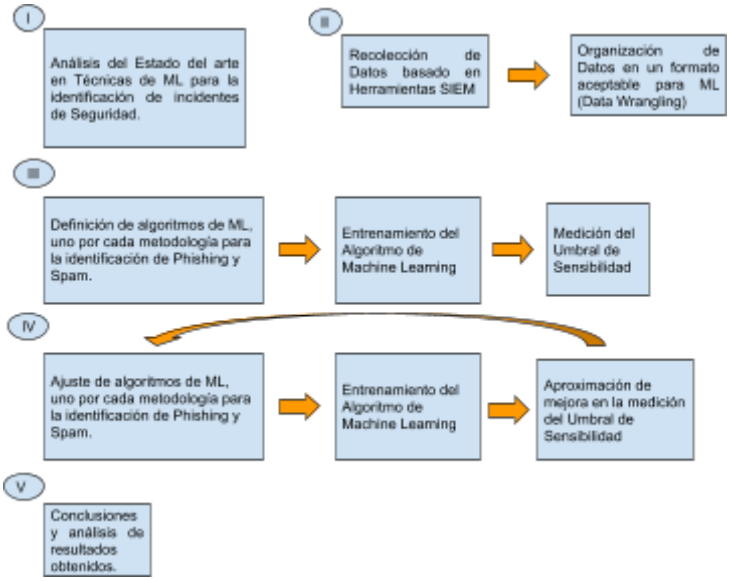


Imagen 3. Plan de actividades propuesto.

6. Componentes de la Solución:

Aprovechando el uso de librerías de ML (Scikit-Learn) pretendemos iniciar con un conjunto de algoritmos de ML, uno por cada tipo de estrategia, luego de analizar los resultados, pretendemos iterar el ciclo de vida natural para los algoritmos de ML, hallando así, estrategias que mejoren el umbral de identificación de Phishing y Spam. Finalmente, comentaremos los hallazgos.

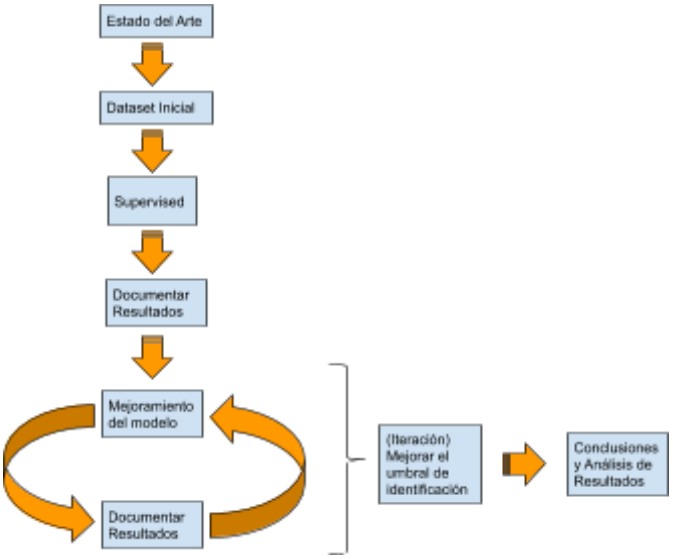


Imagen 4. Componentes de la metodología para dar respuesta al objetivo general.

7. Estrategia y Visión de la Solución:

El proyecto se enfoca en la aplicación de algoritmos de ML para la detección de Phishing y Spam. También, detalla el análisis de diferentes estrategias aproximaciones para la detección automática de estos eventos, validando así, la eficiencia de cada una de las propuestas y su comportamiento frente a diferentes escenarios. Adicionalmente, como parte de la implementación, se cuenta con un prototipo que realiza la verificación de Phishing, la cual usa como base un modelo de ML y clasifica páginas web como “phishing” o “no phishing”.

Como parte de nuestra visión, sugerimos que el presente proyecto pueda utilizarse como base para soportar a futuro otros tipos de ciberataques, pero aquella expansión será parte de un esfuerzo para otro proyecto futuro de la maestría MESI.

8. Estado del arte en la detección de Phishing y Spam

La detección automática de phishing es un problema que se ha venido tratando por más de 20 años y con ello, múltiples soluciones han sido propuestas. En términos generales, existen dos formas para clasificar un sitio web como “phishing” o “no phishing”.

La primera de ellas consiste en una validación contra bases de datos con listas blancas (whitelisting) y negras (blacklisting). Sin embargo, esta estrategia requiere una validación manual, puesto que personal dedicado se esfuerza en validar sitios reportados y en actualizar frecuentemente dichas listas. Por esta razón, este método es incapaz de reconocer nuevos sitios de phishing y de mantenerse actualizado a tiempo. Es por esta razón, que surge la segunda estrategia, la cual ha sido relevante y ampliamente utilizada por múltiples soluciones. Esta requiere el uso de técnicas y algoritmos de ML para clasificar de forma automática cada una de las páginas.

Sin embargo, el problema principal latente es la precisión de detección. La complejidad de tal problemática se encuentra en la evasión de falsos positivos (páginas legales que son clasificadas erróneamente como phishing) y falsos negativos (páginas phishing que no son

clasificadas como tal). Por lo cual se requiere del estudio de múltiples características de los datos usados y de cómo éstos ayudan al mejoramiento de la precisión del modelo. Asi, según se usen unas u otras características se obtendrán diferentes resultados.
Las estrategias más utilizadas se pueden clasificar en tres conjuntos:

- **Basadas en URL:** Corresponden a las características obtenidas del análisis exclusivo de las URL tales como: longitud, número de dígitos, uso de caracteres especiales, entre otros. Los modelos que usan este tipo de características son los más utilizados tanto experimental como comercialmente debido a su facilidad de análisis, y a la gran cantidad de información que la sola URL brinda en cuanto a detección de phishing. La siguiente imagen muestra una página de phishing y su dirección URL, la cual difiere mucho con respecto a la versión real.

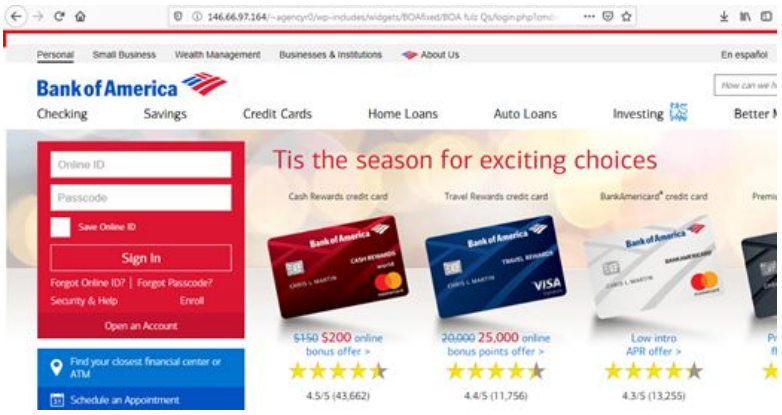


Imagen 5. Ejemplo de detección de Phishing basado en URL.

- **Basadas en contenido:** Corresponden a los patrones obtenidos del análisis del contenido incluyendo: HTML, scripts, CSS, cookies, peticiones, etc. A pesar de que esta estrategia es menos popular que la basada en URL, asegura un mayor uso de recursos y en promedio mucho más tiempo de procesamiento. Sin embargo, este medio brinda una mayor cantidad sustancial de información que permite clasificar con mayor precisión páginas phishing, disminuyendo el número de falsos positivos y falsos negativos. Esta estrategia consiste en la principal alternativa para los casos en la que la URL luce como un sitio legítimo, un ejemplo de ello se puede observar en la siguiente imagen:

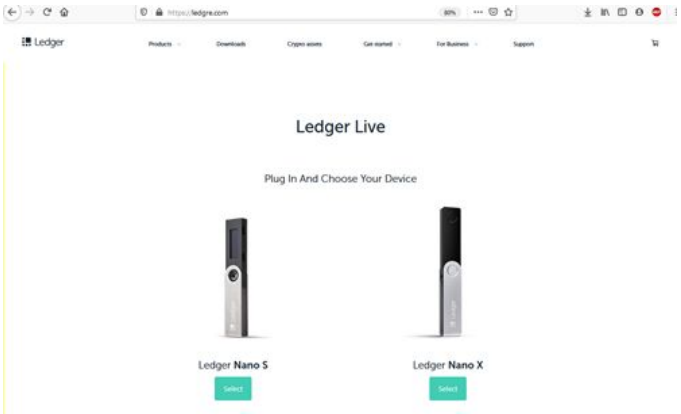


Imagen 6. Ejemplo de detección de Phishing con URL aparentemente auténtica.

- **Basadas en factores externos:** El conjunto de estas características son aquellas que tardan más tiempo en ser extraídas, pero pueden ser consideradas las más importantes y fiables para la detección de phishing, ya que sin importar los filtros que puedan saltarse los atacantes, estos factores son muy difíciles de manipular, tales como la edad del dominio, ranking del dominio o incluso la entidad certificadora (CA, Certification Authority). Por ejemplo, la imagen anterior responde a un dominio "*ledgre.com*" que intenta hacerse pasar por "*ledger.com*" que presta servicios para asegurar criptomonedas usando archivos html de una versión antigua de la página original. Sin embargo, al mirar la entidad certificadora, se observa que es "*Let's Encrypt*", una autoridad certificadora que al ser gratuita es usada por muchos sitios falsos para engañar a los usuarios.

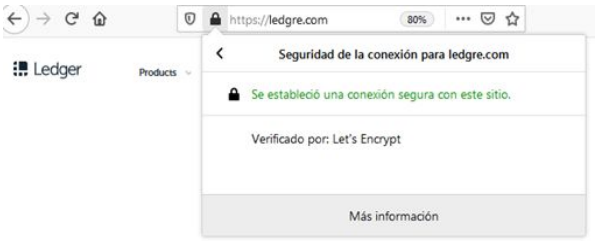


Imagen 7. Ejemplo de detección de Phishing analizando el certificado digital.

9. Diseño detallado de la solución:

Un estudio que ha dirigido el presente proyecto es el de Fon (FON MBAH, 2011), quien resumió las estrategias usadas por algunas de las investigaciones sobre detección de phishing. Sobre ellas, se escogieron las más utilizadas y relevantes para los modelos propuestos. La siguiente tabla muestra la distribución estratégica de las distintas soluciones supervisadas:

Column1	SUM	IP-Based URL	Domain Age	Number of Do-mains	Number of Sub-Domains	Token'JavaScript	Tag<Form>	Huge Number ofLinks	Image Source	MatchingDo-mains	Keywords	Foreign Anchors	Nil Anchor	Excess Dots inPage URL	ExcessSlashesin URL	SSL Certificate	Whois Lookup	Length of theHost name	Length of theEntire URL	Input Forms	HTTPS Proto-col	Iframes	Script Tags	Popup Windows	DomainTokenCount	Average domaintoken length	Longest DomainToken Length	PathTokenCount	AveragePathToken Length	LongestPathToken length	Top-LevelDo-main
A svm-based techniqueto detect phishing urls	3	1									1			1																	
Rule-based phish-ing attack detection	11	1					1	1						1		1		1	1	1		1						1			
Feature selection for improved phishing detection	9	1		1			1	1			1			1						1		1									
Protect sensitive sites from phishing attacks using features extractablefrom inaccessible phishing urls	8		1																						1	1	1	1	1	1	
Efficient prediction of phishingwebsites using supervised learning algorithms	8	1		1								1	1	1	1	1	1														
Detection ofphishing attacks: A machine learning approach	10	1	1	1	1	1	1	1	1	1	1																				
A lexical approach for classifying mali-cious urls	10													1				1	1						1	1	1		1	1	
Phishdef: Urlnames say it all	8	1												1			1		1						1			1			
Identifying suspicious urls: an application of large-scale onlinelearning	4																1	1													1
A frame-work for detection and measurement of phishing attacks	4	1																1													
Beyond blacklists: learning to detect malicious web sites from suspicious urls	4													1			1	1	1												
Phishing websitesdetection based on phishing characteristics in the webpage source code	7	1							1	1											1	1	1	1							
Using domaintop-page similarity feature in machine learning-based web phishing de-tection	3	1												1						1											
Tabsol: An efficient frame-work to defend tabnabbing	2						1													1											
Phishing website detectionusing url-assisted brand name weighting system	0																														
A novel phishingclassification based on url features	3			1	1																										
Detection of hiddenfraudulent urls within trusted sites using lexical features	0																														

Imagen 8. Tabla de correlación de características de detección.

Es necesario resaltar sobre las características identificadas, que ellas fueron agrupadas por popularidad, y así mismo se prescindieron de aquellas que requerían consultas exhaustivas en motores de búsqueda y análisis a profundidad de los componentes de la página. Lo

anterior se realizó con el fin de optimizar el tiempo por consulta y de evaluar el desempeño de modelos con mayor desempeño.
Con base en el análisis anterior, resultó posible identificar las estrategias más relevantes para el proyecto. A continuación se definen cada una de las seleccionadas:

1. **Subdominio:** Se refiere a la cadena correspondiente al subdominio visible de la URL.
2. **Dominio:** Se refiere al dominio principal de la página.
3. **Sufijo:** Se refiere a la extensión del dominio, lo que incluye el *Top Level Domain* y algunas veces el *Second Level Domain*, garantizando el tratamiento individual de dominios como “ejemplo.com” y “ejemplo.com.co”.
4. **Dígitos en total:** Se refiere al número de caracteres numéricos que hay en toda la URL.
5. **Dígitos en host:** Se refiere al número de caracteres numéricos que hay en la primera parte de la URL (host) excluyendo el path.
6. **Longitud total:** Se refiere a la longitud de toda la URL.
7. **Longitud host:** Se refiere a la longitud de la primera parte de la URL (host) excluyendo el path.
8. **Es Latín:** Se refiere a la validación de que la URL no tenga caracteres de otros idiomas o conjuntos de caracteres. La idea es evitar URLs que confundan a los usuarios al aprovecharse de los IDN (Internationalized Domain Name) que permiten diferentes alfabetos como el cirílico, y que a su vez permiten confusiones imperceptibles al humano, ya que por ejemplo el caracter U+0430 que corresponde a la letra “a” minúscula en cirílico luce igual que la “a” en latín y puede ocurrir que aunque dos URL luzcan como “paypal.com”, una sea legal y la otra phishing.
9. **Días de creación:** Se refiere al número de días que el dominio lleva desde su dia de registro. Usualmente los sitios de phishing duran muy poco tiempo y cambian de dominio, por lo que son generalmente “nuevos”.
10. **Esquema:** Se refiere a si es usado http o https
11. **Existe dominio en página:** Se refiere al hecho de que si el dominio aparece en el contenido de la página.
12. **Entidad certificadora:** Se refiere al nombre de la entidad que certifica al dominio. Esa característica es relevante ya que muchos sitios usan autoridades certificadoras gratuitas para lucir “seguros”.
13. **Tag Input:** Se refiere a si el contenido de la página contiene el tag “input”, debido a que normalmente se quieren captar datos sensibles en los sitios phishing.
14. **Tag Form:** Se refiere a si el contenido de la página contiene el tag “form”, debido a que normalmente se quieren captar datos sensibles en los sitios phishing y muchas veces están en un formulario.
15. **Tag Textarea:** Se refiere a si el contenido de la página contiene el tag “textarea”, debido a que normalmente se quieren captar datos sensibles en los sitios phishing.
16. **Tag iframe:** Se refiere a si el contenido de la página contiene el tag “iframe”, ya que a veces es utilizado para hacer creer a los usuarios que el sitio es legítimo cargando la pagina original dentro de sí mismo.
17. **@ Símbolo:** Se utiliza para identificar ataques semánticos al analizar el enlace al que se va a acceder.
18. **Número de links:** Se refiere al número de enlaces (anchors <a>) que tiene una página.
19. **Cadenas sospechosas en el path:** Se refiere a si en el path de la URL contiene cadenas sospechosas como: "www", "http", ".com", ".org"
20. **Es IP:** Se refiere a si el host corresponde a una dirección IP o si es un nombre de dominio.

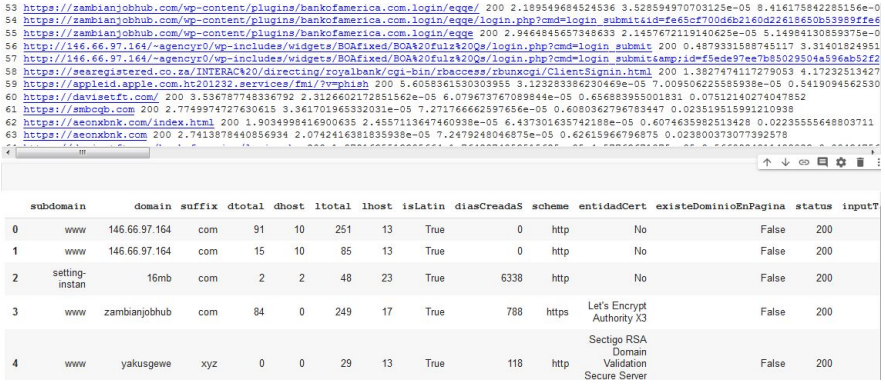
21. **Redirigido:** Se refiere a si la URL redirige a otro sitio o no.
22. **Número de subdominios:** Se refiere al número de subdominios de la url.
23. **Número de palabras clave:** Se refiere al número de palabras clave que se encuentran en el contenido de la página, tales como: 'bank', 'secure', 'ebay', 'coin', 'webscr', 'log in', 'sign in', 'account', 'paypal', 'sulake', 'facebook', 'orkut', 'santander', 'master-card', 'warcraft', 'visa', 'update', 'verify'. Esta lista puede crecer tanto como se requiera.
24. **Enlaces externos:** Se refiere al porcentaje de enlaces que corresponden a enlaces externos sobre el total de enlaces de la página.
25. **Enlaces nulos:** Se refiere al porcentaje de enlaces nulos "javascript:void(0)" o "" sobre el total de enlaces de la página.
26. **Número de puntos:** Se refiere al número de puntos usado en toda la URL.
27. **Número de Barras inclinadas:** Se refiere al número de barras inclinadas presentes en toda la URL.
28. **Tag script:** Se refiere a si en el contenido de la página hay tags "script" sin el atributo src.
29. **Alert/Popup:** Se refiere a si la página genera cuadros de alerta.
30. **Guiones:** Se refiere a si hay guiones presentes en la URL.
31. **Alexa Rank:** Se refiere al ranking del dominio en el sitio alexa.com.
32. **Token más largo del host:** Al tokenizar el host (separándolo por "."), se refiere al token más largo.
33. **Token más largo del path:** Al tokenizar el path(separándolo por "/"), se refiere al token más largo.
34. **Número de tokens en path:** Al tokenizar el path(separándolo por "/"), se refiere al número de tokens.
35. **Promedio de longitud de tokens en path:** Al tokenizar el path(separándolo por "/"), se refiere al promedio de longitud de los tokens.
36. **Promedio de longitud de tokens en host:** Al tokenizar el host (separándolo por puntos), se refiere al promedio de longitud de los tokens.
37. **Extensión del path:** Se refiere a la extensión del último elemento del path.
38. **Longitud del path:** Se refiere a la longitud del path.
39. **Tag Javascript:** Se refiere a si la página contiene *javascript*.
40. **Imágenes externas:** Se refiere al porcentaje de imágenes que no provienen del mismo servidor, sino de un recurso externo.

Comparando con las estrategias del estudio de Fon, se concluye cuáles de ellas eran usadas por cada uno de los modelos de investigaciones previas:

1. A svm-based technique to detect phishing urls: 20, 23, 26.
2. Rule-based phish-ing attack detection: 20, 13, 17, 26, 12, 7, 6, 13, 16, 34, 31.
3. Feature selection for improved phishing detection: 20, 3, 13, 17, 23, 26, 13, 16, 30.
4. Protect sensitive sites from phishing attacks using features extractable from inaccessible phishing urls: 9, 3, 31, 32, 33, 34, 35, 36.
5. Efficient prediction of phishing websites using supervised learning algorithms: 20, 9 , 3, 24, 25, 26, 27, 12.
6. Detection of phishing attacks: A machine learning approach: 20, 9, 3, 21, 38, 13, 17, 39, 11, 23.
7. A lexical approach for classifying malicious urls: 3, 26, 7, 6, 36, 32, 33, 35, 30, 31.
8. Phishdef: Url names say it all Identifying suspicious urls: an application of large-scale online learning: 20, 9, 3, 26, 6, 34, 30, 38.
9. Identifying suspicious urls: an application of large-scale online learning : 9, 7, 3, 37.
10. Phishing websites detection based on phishing characteristics in the webpage source code: 20, 39, 11, 10, 16, 28, 29.

Con base en ello, se puede observar las mejoras que nuestro modelo propone, siendo ellas: “Dígitos en total”, “Dígitos en host”, “Es latín”, “Tag form”, “Tag textarea”, “Símbolo @”, “Cadenas sospechosas en el path”, “Redirigido” e “Imágenes externas”.

Con respecto a nuestra solución, se utilizó para el entrenamiento y las pruebas dos escenarios: Ambos usan la base de datos de *phishtank* [28] para las url de phishing, pero el primer escenario utiliza el “*Top 500 sites de Alexa*” [29] para sitios legítimos, mientras que el segundo escenario utiliza un histórico de sitios consultados por los investigadores. Dichos escenarios permiten observar el comportamiento de los modelos según sus resultados y evalúa en un ambiente real.



	subdomain	domain	suffix	dtotal	dhost	ltotal	lhost	isLatin	diasCreadaS	scheme	entidadCert	existeDominioEnPagina	status	inputT
0	www	146.66.97.164	com	91	10	251	13	True	0	http	No	False	200	
1	www	146.66.97.164	com	15	10	85	13	True	0	http	No	False	200	
2	setting-instan	16mb	com	2	2	48	23	True	6338	http	No	False	200	
3	www	zambianjobhub	com	84	0	249	17	True	788	https	Let's Encrypt Authority X3	False	200	
4	www	yakusgewe	xyz	0	0	29	13	True	118	http	Sectigo RSA Domain Validation Secure Server	False	200	

Imagen 9. Captura de resultados del modelo propuesto.

Al implementar el clasificador *Random Forests* (algoritmo de ML supervisado) por cada estrategia analizada en el paper de Fon, se puede comparar sus resultados independientes en las categorías de precisión y pérdida de logs:

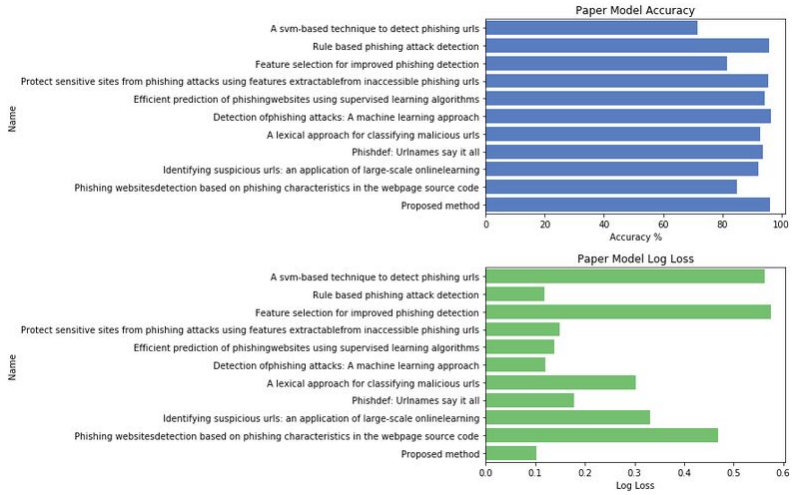


Imagen 10. Gráfico comparativo de precisión y pérdida de log.

Lo gráficos representan la precisión (predicciones cuyo valor predicho es igual al valor real) y el log de pérdida (nivel de incertidumbre de la predicción basándose, el cual detalla el rendimiento del modelo).

Si bien se observa que en algunos de los modelos presentados existe una alta precisión (más de 90%), este no resulta ser un factor determinante, dado que la clasificación solo posee dos estados, siendo respectivamente Verdadero o Falso. Es decir, no hay forma de ver los puntos intermedios, lo que resulta en problemas con predicciones nuevas. A diferencia del segundo criterio, los diferentes valores del Log de pérdida demuestran que el modelo propuesto es el que menor nivel de incertidumbre tiene y por lo tanto, posee mayor certeza en sus resultados.

Con respecto al desempeño del modelo propuesto, se realizó un perfilamiento con el fin de conocer qué datos debían ser modificados o eliminados para mejorar la eficiencia y precisión del modelo, dicho análisis se puede observar en la siguiente imagen:

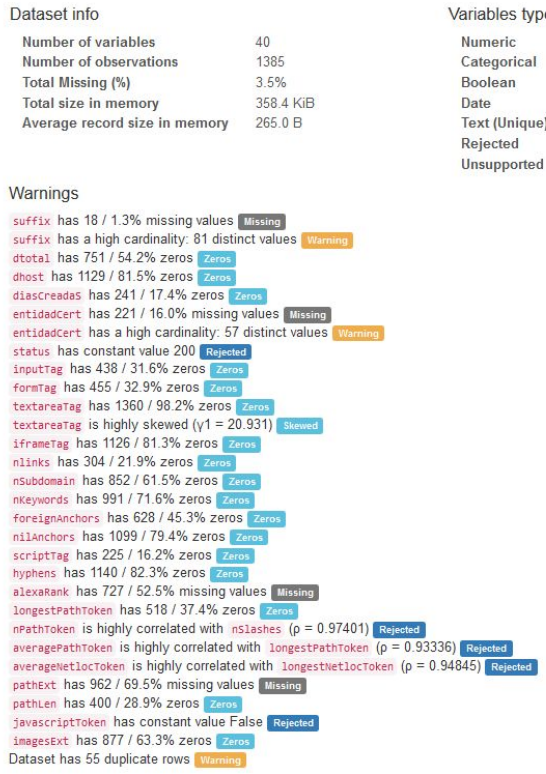


Imagen 11. Perfilamiento de desempeño del modelo de ML propuesto.

Luego de remover las columnas innecesarias (o estaban fuertemente correlacionadas con otra) y de corregir valores nulos, se pudo observar la primera gráfica que correlaciona las diferentes características usadas por nuestro modelo:

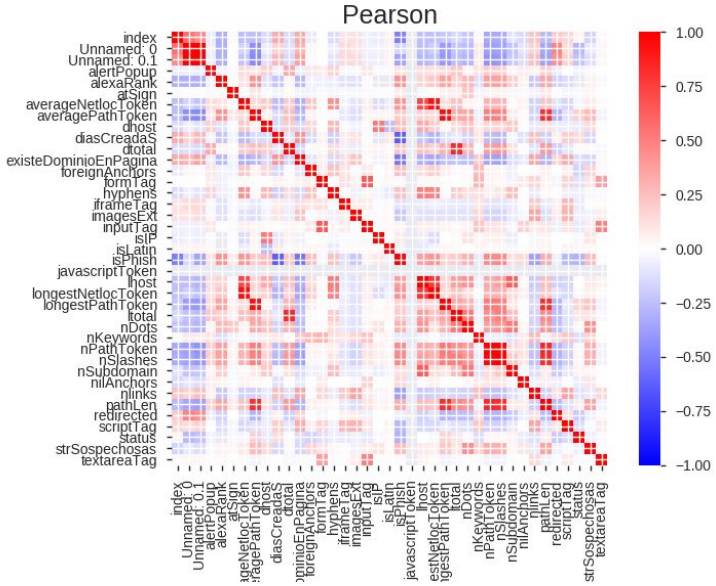


Imagen 12. Características del modelo propuesto.

El análisis derivado de la anterior gráfica detalla que la columna/fila llamada isPhish muestra la correlación de las otras características con las páginas que son phishing. Se observa

entonces, que las características que más influyen para definir una página como phishing o no son: la edad del dominio (días Creada), si el dominio aparece en el contenido de la página (existeDominioEnPagina) y el ranking de Alexa (alexaRank).

Con estos datos, fue posible analizar los diferentes algoritmos de ML para conocer su desempeño. El conjunto de datos se dividió tomando un 80% para el entrenamiento y un 20% para las pruebas. Cabe anotar que solo se usaron algoritmos de aprendizaje supervisado ya que son los que permiten clasificar según el aprendizaje de un conjunto definido y clasificado previamente. Es decir, entrena conociendo el conjunto al que hace parte cada registro. Los resultados fueron los siguientes:

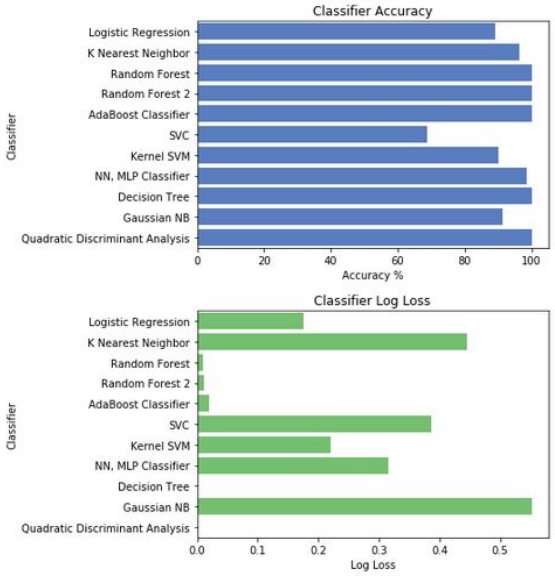


Imagen 13. Comparación de precisión y log de pérdida en los tipos de algoritmos de ML.

De la gráfica anterior se puede observar que los mejores resultados en cuanto a precisión se obtienen al usar Random Forest, Adaboost Classifier, Árbol de decisión y QDA. Para las siguientes implementaciones se decidió utilizar un clasificador de Random Forest ya que presenta muy poca pérdida y alta precisión.

11. Detalle del prototipo:

Se implementó un prototipo para los browsers (Extensión Firefox) que evalúa cada página con el modelo creado e informa el porcentaje identificación de phishing. Para el primer escenario definido, usando únicamente las páginas del ranking de alexa para definir el conjunto de páginas “no phishing”, se observa que aunque los resultados de precisión y de Log Loss son muy buenos en general, en la práctica no lo son, debido a que puede clasificar eficientemente las páginas de la lista del ranking de alexa, pero una vez se consulta otro recurso incluso en los mismos dominios, son clasificados como phishing.
Por ejemplo, al consultar cualquier recurso de w3schools el resultado es el siguiente:

Imagen 14. Ejemplo de detección del prototipo para Phishing.

Lo anterior interpreta que hay un 96,5% de probabilidad que la página sea phishing según el algoritmo aunque ésta no lo sea en realidad. El error se debe a que el dataset consultado en alexa lista únicamente la URL de la página principal y no tiene asociado ningún otro recurso. Por lo tanto, el modelo prioriza esto para evaluar cualquier petición. Lo que quiere decir que si se consultara https://w3schools.com el algoritmo analizaría el sitio y la probabilidad de que sea phishing es mucho más baja.
Gracias a esta prueba se evidenció que trabajar con las URL de las páginas listadas en el ranking de alexa no es del todo efectivo. Por lo tanto, para hacer más equitativo el dataset se decidió obtener un conjunto de datos más apropiado, registrando las consultas del grupo de trabajo a sitios legales y volviendo a evaluar el modelo generado.
Gracias a esto se pudo mejorar la población del dataset con datos más regulares y reales. De una población de páginas de “no phishing” de alrededor de 350 se pasó a cerca de 800.

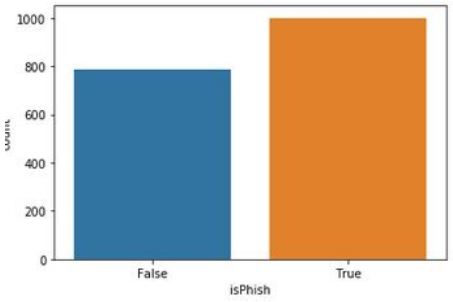


Imagen 15. Resultado del mejoramiento del dataset.

Al realizar de nuevo la prueba, el modelo obtiene una probabilidad de que la página sea phishing, y este resultado ha disminuido significativamente cuando se compara con la prueba anterior. Sin embargo, aún sigue clasificándose erróneamente como phishing:

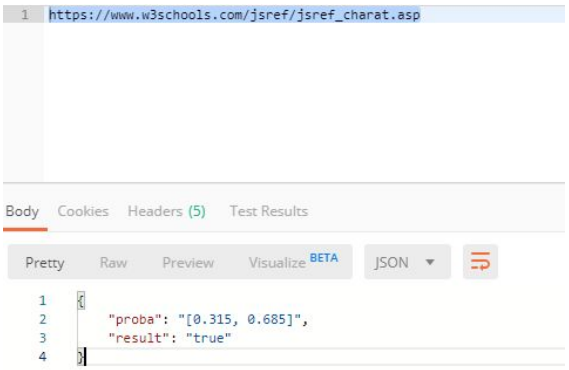


Imagen 16. Resultados de la segunda prueba del prototipo.

Este resultado se debe principalmente al número de datos analizados y al tamaño del dataset. Sin embargo, el prototipo acepta todos los resultados y muestra de forma diferente los resultados basados en diferentes niveles.
Esta implementación está desarrollada como un complemento temporal para Firefox y los resultados se muestran de la siguiente manera en el servidor:

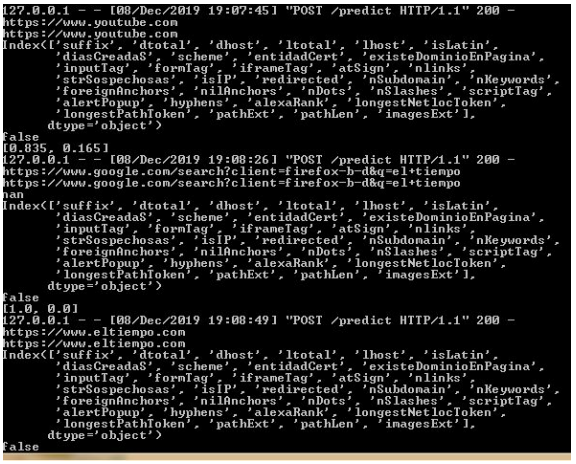


Imagen 17. Muestra del lado del servidor del resultado del análisis del prototipo.

Del lado del cliente, dicha prueba se puede observar de la siguiente manera:

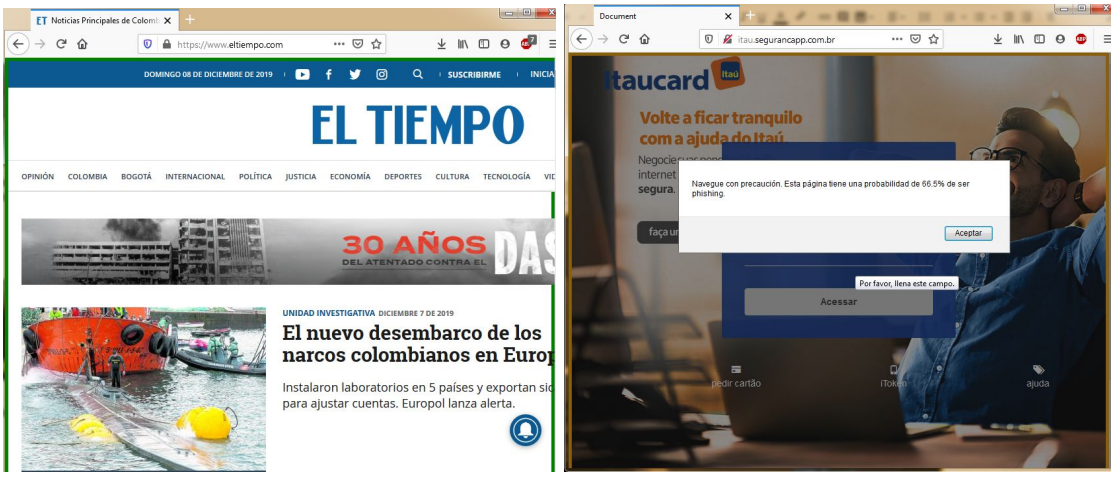


Imagen 18. Página auténtica vs Phishing analizada por el prototipo.

En términos generales, el modelo propuesto detecta con alta probabilidad los sitios de phishing, pero con el dataset actual, también detecta múltiples falsos positivos (como el caso de w3schools mencionado previamente) que pueden ser solucionados al incrementar el número de registros del *dataset*.

12. Estructura del Manejo de Costos y Presupuesto:

Para la estructura de costos hemos decidido contemplar dos escenarios en donde ofrecemos dos servicios, uno para personas naturales y otro para empresas, dichos servicios se distribuyen en una solución completamente basada en Cloud y basada en infraestructura On-premise (Para mayor detalle, favor remitirse al Excel adjunto).
Vale la pena resaltar que la solucion Cloud posee un menor coste de aprovisionamiento y de mantenimiento, maximizando las ganancias netas, razón por la cual es la estrategia comercial elegida.
A continuación se expone la diferencia de costos hallada, representando la fórmula que mejor describe el comportamiento del negocio:

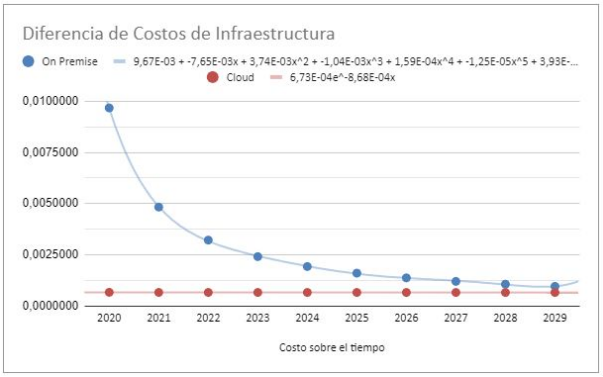


Imagen 19. Gráfica diferencial de costos, On-premise vs Cloud.

Tomando como base la demanda, hemos hallado los ingresos de cada tipo de cliente basado en un precio base ($ 2 USD para personas naturales y $ 3000 para empresas), los resultados fueron:

Precio en USD

Año	Clientes Personas naturales	Clientes empresas
2020	600	12
2021	1.200	24
2022	1.800	36
2023	2.400	48
2024	3.000	60
2025	3.600	72
2026	4.200	84
2027	4.800	96
2028	5.400	108
2029	6.000	120

Ingresos Anuales

Año		Personas Naturales	Clientes Empresa	Total
2020		14.400	432.000	446.400
2021		28.800	864.000	892.800
2022		43.200	1.296.000	1.339.200
2023		57.600	1.728.000	1.785.600
2024		72.000	2.160.000	2.232.000
2025		86.400	2.592.000	2.678.400
2026		100.800	3.024.000	3.124.800
2027		115.200	3.456.000	3.571.200
2028		129.600	3.888.000	4.017.600
2029		144.000	4.320.000	4.464.000

Imagen 21. Tabla de clientes y de ingresos por año proyectado a 10 años.

Finalmente, obtuvimos el siguiente flujo de caja, el cual demuestra la viabilidad del emprendimiento:

Año	2020 1	2021 2	2022 3	2023 4	2024 5	2025 6	2026 7	2027 8	2028 9	2029 10
Ingresos										
Ingresos Persona Natural + Cliente Empresa	$ 446.400,00	$ 892.800,00	$ 1.339.200,00	$ 1.785.600,00	$ 2.232.000,00	$ 2.678.400,00	$ 3.124.800,00	$ 3.571.200,00	$ 4.017.600,00	$ 4.464.000,00
Costos	$ 118.207,05	$ 126.991,84	$ 136.401,24	$ 146.360,62	$ 156.924,81	$ 168.154,08	$ 180.114,76	$ 192.879,80	$ 206.529,43	$ 221.151,89
TOTAL COSTOS FIJOS										
Cloud	3.894,83	7.744,43	11.594,03	15.443,63	19.293,23	23.142,83	26.992,43	30.842,03	34.691,63	38.541,23
Alquiler	*2.628,12*	*2.706,96*	*2.788,17*	*2.871,82*	*2.957,97*	*3.046,71*	*3.138,11*	*3.232,26*	*3.329,22*	*3.429,10*
Mano de Obra	*52.411,92*	*52.303,68*	52.303,68	52.303,68	52.303,68	52.303,68	52.303,68	52.303,68	52.303,68	52.303,68
Administracion	146,01	150,39	154,90	159,55	164,33	169,26	174,34	179,57	184,96	190,51
Suministros (Agua, Luz, Gas, Internet)	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00	1.000,00
Material de Oficina	500,00	500,00	500,00	500,00	500,00	500,00	500,00	500,00	500,00	500,00
TOTAL COSTOS VARIABLES										
Seguros Sociales	7.861,79	7.845,55	7.845,55	7.845,55	7.845,55	7.845,55	7.845,55	7.845,55	7.845,55	7.845,55
Marketing	*49.764,39*	*54.740,83*	*60.214,91*	*66.236,40*	*72.860,04*	*80.146,05*	*88.160,65*	*96.976,72*	*106.674,39*	*117.341,83*
Depreciacion										
UAI	328.193	765.808	1.202.799	1.639.239	2.075.075	2.510.246	2.944.685	3.378.320	3.811.071	4.242.848
- Impuestos [38,5%]	126.354	294.836	463.078	631.107	798.904	966.445	1.133.704	1.300.653	1.467.262	1.633.497
+ Depreciacion										
+ Desembolso										
- Intereses										
- Amortizacion										
FCN	201.839	470.972	739.721	1.008.132	1.276.171	1.543.801	1.810.981	2.077.667	2.343.808	2.609.352

Imagen 22. Flujo de caja proyectado a 10 años.

13. Conclusiones:

1. Machine Learning es una **alternativa viable** para la identificación automática de incidentes de seguridad como Phishing y Spam.
2. **Disminuye el tiempo de identificación** de incidentes de seguridad.
3. **Cada tipo de incidente debe abordarse de forma distinta** ya que cada uno posee una complejidad diferente.
4. Para el caso de ***phishing***, es necesario contar con un **dataset reciente y con un gran número de registros,** con el fin de mejorar el umbral de detección.
5. Para el caso de ***spam***, el ***dataset debe ser adaptativo*** al ambiente final del usuario.
6. **Para comercializar el producto**, resulta favorable el ofrecer un **Multi tenant de Cloud,** pues disminuye los costos de operación.
7. **La gran ventaja de nuestro producto** es también la naturaleza del correo

electrónico, es decir, el hecho de ser **asíncrono**.

14. Trabajo Futuro

Del presente trabajo se espera que pueda ser una base para continuar investigando sobre la detección de otros incidentes de seguridad que pueden beneficiarse de algoritmos de ML.

15. Referencias:

[1] Amos, B., Turner, H., & White, J. (2013). Applying machine learning classifiers to dynamic Android malware detection at scale. *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC).* Sardinia, Italy: IEEE.

[2] Banoth, L., Teja, S. K., Saicharan, M., & Chandra, N. J. (2017). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *International Journal of Research.* Edupedia Publications.

[3] Barreno, M., Nelson, B., Joseph, A. D., & Tygar, J. D. (2010). The security of machine learning. *Machine Learning*, 121-148.

[4] Biswas, S. (2018). Intrusion Detection Using Machine Learning: A Comparison Study. *International Journal of Pure and Applied Mathematics* , 101-114.

[5] Borges, Raymond C., Beaver, J. M., Buckner, M., Morris, T., Adhikari, U., & Pan, S. (2014). Machine Learning for Power System Disturbance and Cyber-attack Discrimination. *2014 7th International Symposium on Resilient Control Systems (ISRCS).* Denver, CO, USA: IEEE.

[6] Dua, S., & Du, X. (2011). *Data Mining and Machine Learning in Cybersecurity.* Boston: Auerbach Publications Boston.

[7] Livadas, C., Walsh, B., Lapsley, D., & Strayer, T. (2006). Using Machine Learning Techniques to Identify Botnet Traffic. *Proceedings. 2006 31st IEEE Conference on Local Computer Networks.* Tampa, FL, USA: IEEE.

[8] Rieck, K., Trinius, P., Willems, C., & Holz, T. (2011). Automatic Analysis of Malware Behavior using Machine Learning. *Journal of Computer Security*, 639-668.

[9] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., . . . Hakimian, P. (2011). Detecting P2P Botnets through Network Behavior Analysis and Machine Learning. *2011 Ninth Annual International Conference on Privacy, Security and Trust.* Montreal, QC, Canada: IEEE.

[10] Shike, M., & Xiaojin, Z. (2015). Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (págs. 2871-2877). Madison WI: University of Wisconsin-Madison.

[11] Singh, J., & Nene, M. (2013). A Survey on Machine Learning Techniques for Intrusion Detection Systems. *International Journal of Advanced Research in Computer and Communication Engineering*.

[12] Takahashi, T. (2018). *Android Application Analysis Using Machine Learning Techniques.*

[13] Wehenkel, L. (1997). Machine learning approaches to power-system security assessment. *IEEE Expert*.

[14] Machine Learning Crash Course 2 Hours. Eureka on Youtube. https://www.youtube.com/watch?v=b2q5OFtxm6A&list=PL9ooVrP1hQOHUfd-g8GUpKI3hHOwM_9Dn&index=2 .

[15] Abutair, H. Y. A., & Belghith, A. (2017). Using Case-Based Reasoning for Phishing Detection. *Procedia Computer Science*, *109*, 281–288. https://doi.org/10.1016/j.procs.2017.05.352

[16] Chanti, S., & Chithralekha, T. (2019). Classification of Anti-phishing Solutions. *SN Computer Science*, *1*(1). https://doi.org/10.1007/s42979-019-0011-2

[17] E., B., & K., T. (2015). Phishing URL Detection: A Machine Learning and Web Mining-based Approach. *International Journal of Computer Applications*, *123*(13), 46–50. https://doi.org/10.5120/ijca2015905665

[18] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029

[19] Yi, P., Guan, Y., Zou, F., Yao, Y., Wang, W., & Zhu, T. (2018). Web Phishing Detection Using a Deep Learning Framework. *Wireless Communications and Mobile Computing*, *2018*, 1–9. https://doi.org/10.1155/2018/4678746

[20] Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina. *Proceedings of the 16th International Conference on World Wide Web - WWW '07*. https://doi.org/10.1145/1242572.1242659

[21] Büber, E. (2018, May 21). Phishing URL Detection with ML. Retrieved from https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5

[22] Norte, Jose. (2010). Spam Classification Using Machine Learning Techniques - Sinespam.

[23] Bae, S. I., Lee, G. B., & Im, E. G. (2019). Ransomware detection using machine learning algorithms. *Concurrency and Computation: Practice and Experience*. https://doi.org/10.1002/cpe.5422

[24] Comparison of machine learning methods in email spam detection. (n.d.). Retrieved September 23, 2019, from https://www.matchilling.com/comparison-of-machine-learning-methods-in-email-spam-detection/

[25] Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, *5*(6), e01802. https://doi.org/10.1016/j.heliyon.2019.e01802

[26] Spam and phishing in Q2 2019. (2019, September 11). Retrieved September 23, 2019, from https://securelist.com/spam-and-phishing-in-q2-2019/92379/

[27] MBAH, K. F. (2011). *A PHISHING E-MAIL DETECTIONAPPROACH USING MACHINELEARNING TECHNIQUES.* Retrieved from https://unbscholar.lib.unb.ca/islandora/object/unbscholar%3A8107/datastream/PDF/view

[28]https://www.phishtank.com/

[29]https://www.alexa.com/topsites