DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN (SISTEMAS E INFORMÁTICA)

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIEROS EN SISTEMAS E INFORMÁTICA

TEMA: "MODELO PARA LA DETECCIÓN Y MITIGACIÓN DE ATAQUES DE SUPLANTACIÓN DE IDENTIDAD, UTILIZANDO APRENDIZAJE AUTOMÁTICO"

AUTORES: ESPINOZA PADILLA, BRYAN ALEJANDRO
SIMBA AMORES, JESSICA PAOLA

DIRECTOR: ING. FUERTES DÍAZ, WALTER MARCELO PhD

SANGOLQUÍ

2019

CERTIFICADO

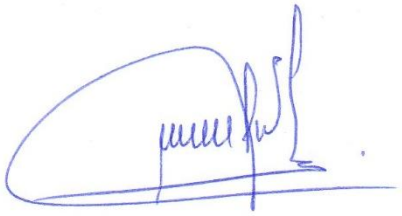
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERIA EN SISTEMAS

CERTIFICACIÓN

Certifico que el trabajo de titulación, *"MODELO PARA LA DETECCIÓN Y MITIGACIÓN DE ATAQUES DE SUPLANTACIÓN DE IDENTIDAD, UTILIZANDO APRENDIZAJE AUTOMÁTICO"* fue realizado por los señores ***Espinoza Padilla Bryan Alejandro y Simba Amores Jessica Paola***, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 2 de julio del 2019



Ing. Fuertes Díaz, Walter Marcelo PhD.

C.C.: 1707017701

AUTORÍA DE RESPONSABILIDAD

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERIA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, ***Espinoza Padilla Bryan Alejandro y Simba Amores Jessica Paola***, declaramos que el contenido, ideas y criterios del trabajo de titulación: ***"MODELO PARA LA DETECCIÓN Y MITIGACIÓN DE ATAQUES DE SUPLANTACIÓN DE IDENTIDAD, UTILIZANDO APRENDIZAJE AUTOMÁTICO"*** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz

Sangolquí, 2 de julio del 2019

Bryan Alejandro Espinoza Padilla
C.C.: 1718295346

Jessica Paola Simba Amores
C.C.: 1721680294

AUTORIZACIÓN

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERIA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Nosotros, ***Espinoza Padilla Bryan Alejandro*** *y* ***Simba Amores Jessica Paola*** *autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación:* ***"MODELO PARA LA DETECCIÓN Y MITIGACIÓN DE ATAQUES DE SUPLANTACIÓN DE IDENTIDAD, UTILIZANDO APRENDIZAJE AUTOMÁTICO"*** *en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.*

Sangolquí, 2 de julio del 2019

Bryan Alejandro Espinoza Padilla
C.C.: 1718295346

Jessica Paola Simba Amores
C.C.: 1721680294

DEDICATORIA

El proyecto de tesis lo dedicamos a:

Dios y la virgen de Guadalupe, quienes fueron nuestra guía en el proceso académico y por brindarnos salud, sabiduría y fe para continuar a pesar de las adversidades.

Nuestros padres por darnos su amor, comprensión, dedicación y sacrificio en todos estos años de desarrollo universitario, por los consejos y motivaciones para cumplir este sueño anhelado. Gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que ahora somos.

Nuestros hermanos y hermanas por el cariño, interés y motivación que a diario nos brindaron para culminar esta etapa.

Finalmente, a nuestros amigos y amigas que estuvieron apoyándonos y motivándonos cuando más lo necesitamos y por haber sido el motivo de alegrías y tristezas durante la vida universitaria.

AGRADECIMIENTO

Agradecemos a todas las personas que estuvieron pendientes de nuestro desempeño durante todos los años de Universidad, por el apoyo incondicional y por las palabras de motivación en su momento. Pero por, sobre todo, hacemos un agradecimiento especial a Dios y la virgen de Guadalupe que nos mantuvo juntos, llenos de sabiduría y entrega en el desarrollo del proyecto de investigación.

Agradecemos a nuestras familias, que siempre estuvieron apoyándonos en todo momento. Por la paciencia y ánimo para culminar y no rendirnos.

Agradecemos a nuestros amigos y compañeros que estuvieron a nuestro lado, especialmente aquellos que nos extendieron su mano en momentos difíciles y también aquellos que nos compartieron sus conocimientos.

Agradecemos el Ing. Walter Fuertes por acogernos en el desarrollo de este proyecto, por las guías que nos dio y la paciencia en la elaboración del proyecto. Además, del esfuerzo e interés entregado durante estos meses y por el conocimiento compartido con nosotros.

Finalmente, agradecemos a la Universidad de las Fuerzas Armadas ESPE, por darnos la oportunidad de formarnos académicamente con docentes entregados a la educación, al cumplimiento de valores e inspirarnos en el cumplimiento de nuestras metas y sueños.

ÍNDICE

CERTIFICADO ..... **I**

AUTORÍA DE RESPONSABILIDAD ..... **II**

AUTORIZACIÓN ..... **III**

DEDICATORIA ..... **IV**

AGRADECIMIENTO ..... **V**

ÍNDICE ..... **VI**

ÍNDICE DE TABLAS ..... **X**

RESUMEN ..... **XIII**

ABSTRACT ..... **XIV**

CAPÍTULO I **1**

INTRODUCCIÓN **1**

1.1. Antecedentes 1

1.2. Problemática 2

1.3. Justificación 3

1.4. Objetivos 5

1.4.1. Objetivo General 5

1.4.2. Objetivos Específicos 5

1.5. Alcance 5

1.6. Metodología de Investigación-Acción 6

CAPÍTULO II **8**

MARCO TEÓRICO **8**

2.1. Delito informático 8

2.2. Framework de detección y mitigación 9

2.2.1. Amenazas 10

2.2.2. Vulnerabilidades 12

2.2.3. Riesgo 13

2.2.4. Ataques de Ingeniería Social – Phishing 14

2.2.5. Ciclo de vida de Phishing 16

2.2.6. Caracterización de phishing 17

2.2.7. Técnicas de Machine Learning 18

2.2.8. Árbol de Decisión 21

2.2.9. Naive Bayes 22

CAPÍTULO III 25

ESTADO DEL ARTE 25

3.1. Mapeo Sistemático de Literatura 25

3.1.1. Definición de las preguntas de investigación 26

3.1.2. Planteamiento de la Estrategia de Búsqueda 26

3.1.3. Definición de los Criterios de Inclusión y Exclusión 28

3.1.4. Proceso de Selección de Estudios Primarios 29

3.2. Resultados 29

3.2.1. Técnicas de Machine Learning aplicadas por año 30

3.2.2. Técnicas de Machine Learning más empleadas 30

3.2.3. Descripción de las técnicas de Machine Learning 32

CAPÍTULO IV 35

DISEÑO E IMPLEMENTACIÓN DEL MODELO DE DETECCIÓN Y MITIGACIÓN ..35

4.1. Metodología de desarrollo....35

4.1.1. Ciclo 1: Viabilidad....36

4.1.2. Ciclo 2: Requisitos....36

4.1.3. Ciclo 3: Diseño e implementación....39

4.1.3.1. Arquitectura del Framework Jupyter....39

4.1.3.2. Arquitectura del modelo de detección y mitigación....40

4.1.3.3. Diagrama de Secuencia....42

4.1.3.4. Diagrama de Clases....44

4.1.3.5. Obtención de Data para el entrenamiento y detección....46

4.1.3.6. Clasificador de Naive Bayes (NB)....47

4.1.3.7. Clasificador de Árboles de Decisión....48

4.1.3.8. Selección de características....49

4.1.3.9. Diseño de la interfaz del modelo de detección y mitigación....51

CAPÍTULO V....60

EVALUACIÓN Y VALIDACIÓN DE RESULTADOS....60

5.1. Resultados....60

5.1.1. Etapa de entrenamiento del modelo....60

5.1.2. Etapa de detección del modelo....61

5.2. Evaluación del desempeño del modelo....62

5.3. Validación con otros modelos de clasificación de Machine Learning....66

5.4. Rendimiento y consumo de recursos....69

5.5. Discusión....72

CAPÍTULO VI 74

CONCLUSIONES Y RECOMENDACIONES 74

6.1. Conclusiones 74

6.2. Recomendaciones 75

REFERENCIAS BIBLIOGRÁFICAS 77

ÍNDICE DE TABLAS

Tabla 1. *Técnicas de Machine Learning con mayor uso*.....31

Tabla 2. *Requisito funcional RF_01*.....37

Tabla 3. *Requisito funcional RF_02*.....37

Tabla 4. *Requisito funcional RF_03*.....38

Tabla 5. *Requisito funcional RF_04*.....38

Tabla 6. *Información enlaces extraídos con PhishTank* 47

Tabla 7. *Características seleccionadas*.....50

Tabla 8. *Método leerCorreos()*.....53

Tabla 9. *Matrices de características* 57

Tabla 10. *Método de verificar el correo*.....58

Tabla 11. *Resultados al detectar con el modelo*.....58

Tabla 12. *Matriz de confusión*.....62

Tabla 13. *Valores para las ecuaciones de predicción*.....63

Tabla 14. *Resultados al emplear la matriz de confusión* 66

Tabla 15. *Algoritmo clasificador de Regresión Logística*.....67

Tabla 16. *Algoritmo clasificador Ficticio* 67

Tabla 17. *Algoritmo clasificador Random Forest*.....68

Tabla 18. *Resultado de los clasificadores de ML*.....68

Tabla 19. *Características de las computadoras*.....70

Tabla 20. *Consumo de recursos en la fase de entrenamiento del modelo* 70

Tabla 21. *Consumo de recursos en la fase de detección del modelo* 71

ÍNDICE DE FIGURAS

Figura 1. Ciclo de la investigación-acción según Kemmis 7

Figura 2. Delito informático 10

Figura 3. Ciclo de vida de phishing 16

Figura 4. Las 4 etapas del ciclo de modelado de Machine Learning (ML) 20

Figura 5. Flujo de aprendizaje 19

Figura 6. Inferencia de un modelo 20

Figura 7. Algoritmo de crecimiento para Árboles de Decisión 21

Figura 8. Modelo de un Árbol de Decisión 22

Figura 9. Estructura del clasificador Naive Bayes 23

Figura 10. Clasificador Naive Bayes 24

Figura 11. Fases para realizar un Mapeo Sistemático de Literatura 25

Figura 12. Fases para el Planteamiento de la Estrategia de Búsqueda 27

Figura 13. Técnicas de Machine Learning por año 30

Figura 14. Modelo de espiral común 35

Figura 15. Arquitectura del Framework Jupyter 40

Figura 16. Arquitectura del ambiente controlado del ataque 41

Figura 17. Arquitectura del modelo 41

Figura 18. Diagrama de secuencia para el entrenamiento del modelo 42

Figura 19. Diagrama de secuencia para la detección y mitigación 43

Figura 20. Diagrama de Clases 45

Figura 21. Matriz de datos .csv de pruebas 46

Figura 22. Diagrama de Flujo Naive Bayes 48

Figura 23. Diagrama de Flujo Arboles de Decisión 49

Figura 24. Pantalla para solicitar entrenamiento 51

Figura 25. Curva de aprendizaje de Naive Bayes 51

Figura 26. Curva de aprendizaje de Árboles de decisión 52

Figura 27. Interfaz de resultados de entrenamiento 53

Figura 28. Data de un correo de Gmail 56

Figura 29. Número de correos con phishing en la etapa de entrenamiento 61

Figura 30. Resultados de la etapa de detección 62

Figura 31. Porcentaje de detección de Clasificadores de ML 69

RESUMEN

El continuo surgimiento de nuevas tecnologías ha ocasionado el incremento en la delincuencia cibernética. Estos ciberataques se han convertido en amenazas graves, dando origen a nuevos malware o programas con código malicioso, en donde utilizan técnicas de Ingeniería Social con el fin de robar o destruir datos importantes. Por tal razón, los ciberdelincuentes aprovechan la ingenuidad de las personas para robar información confidencial, esto ha generado el incremento de fraudes durante los últimos cinco años. Frente a este escenario el presente proyecto se enfoca en el desarrollo de un modelo para la detección y mitigación de ataques de suplantación de identidad utilizando técnicas de Machine Learning. Cabe mencionar que el desarrollo se realizó en un ambiente controlado para garantizar la seguridad del entorno. Para la generación de correos infectados se extrajeron enlaces maliciosos de PhishTank. De modo que se realizó la extracción de las características de los correos para la fase de entrenamiento utilizando el algoritmo Naive Bayes. Luego se detectaron los correos infectados mediante el algoritmo de Árboles de Decisión con la finalidad de enviar a cuarentena los correos ilegítimos. Por último, se validó con los algoritmos de ML Random Forest, Regresión Logística y Clasificador Ficticio, con la finalidad de conocer el porcentaje de precisión en la detección de phishing de la solución propuesta en comparación con otros algoritmos de aprendizaje supervisado.

PALABRAS CLAVE:

- **INGENIERÍA SOCIAL**
- **SUPLANTACIÓN DE IDENTIDAD**
- **TÉCNICAS DE MACHINE LEARNING**
- **APRENDIZAJE SUPERVISADO**

ABSTRACT

The continuous emergence of new technologies has caused the increase in cybercrime. These cyber-attacks have become serious threats, giving rise to new malware or programs with malicious code, where they use social engineering techniques in order to steal or destroy important data. For this reason, cybercriminals take advantage of the ingenuity of people to steal confidential information; this has generated increased fraud during the last five years. Faced with this scenario, the present project focuses on the development of a model for the detection and mitigation of identity theft attacks using Machine Learning techniques. It is worth mentioning that the development was carried out in a controlled environment to guarantee the safety of the environment. For the generation of infected emails, malicious links were extracted from PhishTank. So the extraction of the mail characteristics for the training phase was carried out using the Naive Bayes algorithm. Then the infected emails were detected using the Decision Trees algorithm in order to quarantine the illegitimate emails. Finally, it was validated with the algorithms of ML Random Forest, Logistic Regression and Fictitious Classifier, with the purpose of knowing the percentage of accuracy in the phishing detection of the proposed solution in comparison with other supervised learning algorithms.

KEYWORDS:

- **SOCIAL ENGINEERING**
- **IDENTITY EXPLANATION**
- **MACHINE LEARNING TECHNIQUES**
- **SUPERVISED LEARNING**

CAPÍTULO I

INTRODUCCIÓN

1.1. Antecedentes

En los últimos años, la evolución de la tecnología de información ha ocasionado que el software se encuentre involucrado cada vez más en los productos y servicios de uso cotidiano como la salud, educación, gobernanza, creación y expansión de empresas (Aules Centeno et al., 2016). Esto ha provocado que la demanda sea cada vez más alta (Budnik, 2012), generando un crecimiento sin precedentes (97%) en la utilización del software y dispositivos electrónicos (Sanou, 2015). De manera similar el Instituto Nacional de Estadísticas (INE) señala que el 78.7% de los hogares disponen acceso al Internet, lo que ocasiona que las personas sean dependientes del software para el diario vivir y la utilización en el desarrollo productivo de empresas, gobiernos, instituciones educativas, entre otros.

Sin embargo, la creación de nuevas tecnologías ha provocado que la delincuencia cibernética aumente (Awan & Dahabiyeh, 2018) y sea calificada como el delito de rápido crecimiento a nivel mundial (Osuagwu, et al., 2015). Es decir, los ciberataques se han convertido en amenazas graves para todo el mundo debido al creciente uso de las aplicaciones, dando origen que nuevos malware o programas con código malicioso sean lanzados a diario por ciberdelincuentes a través de Internet con el fin de robar o destruir datos importantes (Chowdhury, Rahman, & Islam, 2017). De este modo los atacantes emplean ingeniosas técnicas para convencer a las víctimas que ingresen a páginas Web engañosas (Ndibwile, Kadobayashi, & Fall, 2017). Una de las técnicas más utilizadas por los ciberdelincuentes es la Ingeniería Social, mediante la cual

engañan a sus víctimas a revelar información de diferentes cuentas, ya sea correo electrónico o bancarias (Osuagwu, et al., 2015). Por sus delicadas consecuencias, la suplantación de identidad ha sido identificada como una de las amenazas más peligrosas, pues puede ocasionar divulgación de información sensible, lo que podría tener graves efectos tales como pérdidas financieras, interrupción de servicio, daños a la imagen pública o incluso paralizar una organización o nación por completo (Osuagwu, et al., 2015).

La suplantación de identidad es una forma de robo de información personal en línea. Los phishers, por ejemplo, usan Ingeniería Social para robar los datos de identidad propia de las víctimas y las credenciales de cuentas financieras. Otra forma de ataque de Ingeniería Social consiste en utilizar correos electrónicos ilegítimos para atraer a víctimas inocentes a sitios web falsos con la intensión de engañar a los usuarios a divulgar datos personales como números de tarjetas de crédito, nombres de usuarios de cuentas, contraseñas y números de seguridad social (Huang, Wang, & Liu, 2011).

1.2. Problemática

El informe de Ciberseguridad Tendencias 2017 menciona la detección de alrededor de 760 millones de ataques de malware distribuidos en países como: China, Turquía, Taiwán, Ecuador y Guatemala (APWG, 2016). Además, la documentación anual de la empresa Digiware revela que Ecuador es el cuarto país de Latinoamérica que recibe más ataques cibernéticos con un 11.22% (Digiware, 2016).

A su vez, el informe de Tendencias de Seguridad Cibernética en América Latina y el Caribe, enfatiza la existencia de dos problemas principales en Ecuador para la reducción de la

ciberdelincuencia: el primero se debe a la escasez de leyes sobre ataques informáticos, y la segunda es la falta de conciencia o educación sobre la seguridad informática. Esto conlleva a evidenciar en el período comprendido entre el 2008 y 2013 el aumento exponencial del 203% y 458% respectivamente de ciberdelitos registrados por la Unidad de Investigación del Delito Cibernético de la Policía Nacional (OEA, Symantec, & AMERIPOL, 2014). El 80% estuvo relacionado con la apropiación indebida de información personal mediante técnicas como el skimming (clonación de tarjetas), el phishing (suplantación de identidad) y la explotación de sistemas de pago en línea. La Policía Nacional informó que, en la segunda mitad del 2013, el país experimentó un incremento del 58.94% de incidentes de fraude electrónico (Ron, Rivera, Fuertes, Toulkeridis, & Díaz, 2019). A su vez las entidades del sector bancario han sido el objetivo de ataques informáticos, esto se refleja en los datos de la Policía Nacional con un registro del 38.48% de denuncias sobre fraudes de phishing, el skimming y el bitching (OEA, Symantec, & AMERIPOL, 2014). De manera similar, Digiware informó sobre el incremento del 1% al 30% de ataques de Ingeniería Social en el transcurso del último año, en sectores financieros, industriales y de comercio (Digiware, 2016).

Acorde a los informes sobre ciberseguridad expuestos durante los últimos años, se demuestra el alto índice de ataques en América Latina y el Ecuador. Frente a este escenario, se propone el desarrollo de un modelo que permita la detección y mitigación de ataques de Ingeniería Social del tipo suplantación de identidad.

1.3. Justificación

Acorde al informe de Digiware, evidencia al Ecuador como el cuarto país que recibe más ataques cibernéticos en Latinoamérica con un 11.22% (Digiware, 2016). A esto se suma el

incremento de amenazas del 1 al 30% de ataques de Ingeniería Social (Digiware, 2016). Incluso durante los últimos años debido a la gran acogida del uso de software, se generó una brecha en la seguridad informática de Instituciones públicas, privadas, de gobierno y educativas (Ron, Fuertes, Bonilla, Toulkeridis, & Diaz, 2018). Otro factor de inseguridad es la falta de conciencia en la sociedad, esto ocasiona que los ataques cibernéticos se incrementen significativamente en los últimos años (OEA, Symantec, & AMERIPOL, 2014).

Desde el punto de vista legal, la Ley N° 2002-67 de Comercio Electrónico Firmas electrónicas y mensajes de Datos (CONGRESO NACIONAL, 2002) proporciona el marco general de los delitos informáticos, en donde están tipificadas algunas contravenciones y sus sanciones. Sin embargo, no todas las violaciones informáticas se encuentran consideradas en esta ley. De manera que, al no considerar los delitos actuales dentro de las leyes con las sanciones correspondientes provoca que se incremente el número de ataques a las diferentes Instituciones (Bustamante, Fuertes, Tulkeredis, & Ron, 2018). Así lo muestra el informe de Tendencias de Seguridad Cibernética en América Latina y el Caribe, el 38.48% de denuncias sobre fraudes electrónicos en entidades públicas y privadas emplearon métodos como phishing, skimming y bitching (OEA, Symantec, & AMERIPOL, 2014).

Por lo expuesto en este apartado, se justifica el desarrollo del modelo para la detección y mitigación de ataques de suplantación de identidad de manera que permita contrarrestar el incremento de este tipo de ataques y sus posibles consecuencias.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un modelo utilizando técnicas de Machine Learning para la detección y mitigación de ataques de Ingeniería Social orientados a la suplantación de identidad.

1.4.2. Objetivos Específicos

- Identificar cuáles son los ataques más comunes sobre Ingeniería Social enfocados a la suplantación de identidad.
- Analizar las técnicas de Machine Learning empleadas en los últimos cinco años para la detección y mitigación de ataques de Ingeniería Social relacionados con la suplantación de identidad.
- Diseñar la arquitectura para la construcción del modelo de detección y mitigación.
- Implementar el modelo que permita la detección y mitigación de ataques de Ingeniería Social, mediante algoritmos de Machine Learning.
- Realizar la evaluación, validación y divulgación de los resultados obtenidos.

1.5. Alcance

El desarrollo del modelo de detección y mitigación para ataques de Ingeniería Social orientados a la suplantación de identidad, se llevará a cabo en la Universidad de las Fuerzas Armadas ESPE en el Laboratorio del Grupo de investigación de Sistemas Distribuidos, Ciberseguridad y Contenido (RACKLY). Conviene señalar, que se utilizará un entorno controlado para la simulación del ataque, por tal motivo se deben cumplir con las normas y reglamentos

vigentes que permitirán respaldar la seguridad del laboratorio y del proyecto. Con ello se menciona a continuación los procesos que se realizarán:

- Estudio sobre los ataques de Ingeniería Social orientados a la suplantación de identidad más común en los últimos cinco años.
- Selección de las técnicas adecuadas de Machine Learning para la detección y mitigación de ataques de Ingeniería Social enfocados a la suplantación de identidad.
- Diseño del modelo para la detección y mitigación de ataques de Ingeniería Social utilizando algoritmos de aprendizaje supervisado.
- Implementación del modelo para la detección y mitigación de ataques de Ingeniería Social, en un ambiente controlado en el laboratorio del Grupo de investigación RACKLY del departamento de Ciencias de la Computación de la ESPE.
- La validación del modelo se efectuará mediante una comparación con otros algoritmos de Machine Learning, esto permitirá validar el modelo y exponer los resultados sobre la solución para la detección y mitigación de ataques de Ingeniería Social orientados a la suplantación de identidad.

1.6. Metodología de Investigación-Acción

La Metodología Investigación-Acción fue descrita por Lewin en 1946 como una espiral que cumple las siguientes etapas: planificación, implementación y evaluación del resultado de la acción. A su vez posee un doble propósito: (1) Actuar para cambiar o mejorar la situación de un grupo de personas e (2) Investigar para generar conocimiento y compresión. Debido a que es considerada como un bucle recursivo y retroactivo de investigación y acción (Latorre, 2003). Lo

que conlleva a la comprobación de ideas en la práctica como medio de mejora en las condiciones sociales e incrementar el conocimiento (Latorre, 2003).

En relación al anterior párrafo se muestra en la **Figura 1** el ciclo de la investigación-acción. Tomado como referencia del modelo de Kemmis y el de Lewin, (1) Inicia con el desarrollo de un plan o la planificación del problema, para este proyecto hace referencia al incremento de ataques de Ingeniería Social en los últimos años, (2) Se realiza un acuerdo o cronograma para poner en práctica en plan, (3) Se observan los resultados o efectos, (4) Al tener los resultados se reflexiona sobre las mejoras o medidas que se deben tomar con respecto a la planificación y se mejora la investigación. A fin de que el ciclo se genere nuevamente.

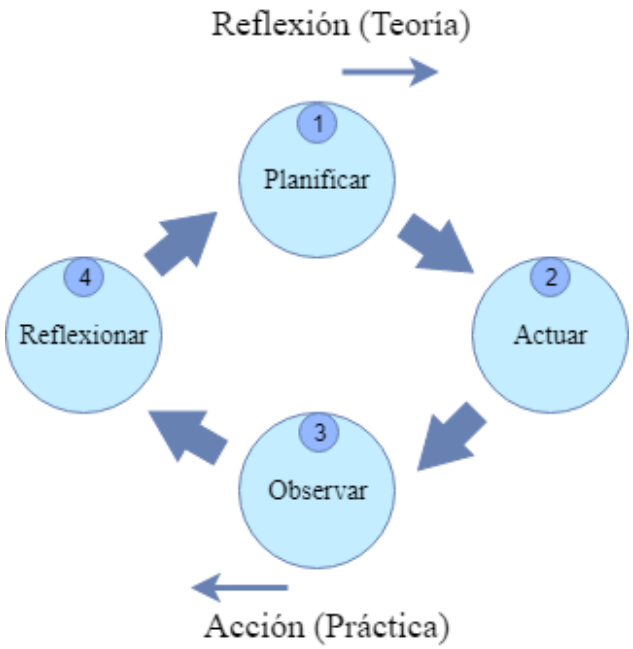


Figura 1. Ciclo de la investigación-acción según Kemmis

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se analizan los fundamentos teóricos sobre los crímenes cibernéticos, especialmente los ataques de Ingeniería Social. Además, se presenta una descripción de las técnicas de Machine Learning que son empleadas en la detección y mitigación de ataques de suplantación de identidad.

2.1. Delito informático

La Unión Internacional de Telecomunicaciones (ITU) define a un delito informático como aquél cuyo objeto o medio de ejecución es un sistema de software, de manera que presenta una estrecha relación con las tecnologías digitales. Además, se le conoce como una forma de fraude informático que utiliza el Internet para su desarrollo. (ITU, 2007). Esto comprende acciones que afectan perjudicialmente a un computador, los dispositivos o el software, mediante la apropiación ilícita de datos, interferencia a bancos electrónicos, y otras acciones relacionadas con la informática (Salgado, 2014). Es por ello que se detalla a continuación algunos tipos de cibercrimen:

- Malware: Según la NIST es un software destinado a realizar procesos no autorizados que tendrán un impacto perjudicial en la confidencialidad, integridad o disponibilidad de un sistema de información (NIST, 2015). Por ejemplo, virus, gusano, caballo de Troya u otra entidad basada en código que infecta un host, un Spyware o código malicioso (NIST, 2015).
- DDOS: Es un intento malicioso de interrumpir el tráfico normal de un servidor, servicio o red de destino. Esto se origina al invadir la infraestructura con una gran cantidad de tráfico de Internet, impidiendo que llegue a su destino (Cloudflare, 2019).

- Ingeniería Social: Es la acción de manipular a una persona a través de técnicas psicológicas o habilidades sociales para la extracción de información personal, el acceso a un sistema o el robo de un activo (Sandoval Castellanos, 2018). Por ejemplo, phishing engaña a las personas a fin de obtener información confidencial; baiting emplea dispositivos de almacenamiento como USB, CD en donde se encuentra instalado un software malicioso, que espera que la víctima inserta el dispositivo en un computador este se instala de inmediato y extrae los datos personales del usuario.

Otros delitos informáticos que se han desarrollado con mayor frecuencia en la actualidad son los siguientes: (1) Cyberbullying es el acoso mediante dispositivos digitales, como celulares, computadoras o tabletas, que son empleadas como medio para enviar, publicar o compartir contenido negativo, perjudicial o falso sobre una persona (ITU, 2016); (2) Cyber Sexting es el envío de contenido sexual a través de dispositivos móviles como celulares o tabletas y se propaga en redes sociales como Facebook, Instagram, Twitter, Snapchat, etc. (UNICEF, 2016); (3) Cyber Grooming lo realiza un adulto que se hace pasar por un niño o adolescente en las redes sociales, para atraer menores de edad con la finalidad de establecer citas, manipularlos y tener contacto sexual a través de amenazas emocionales (Pierdant, 2013); y (4) Cyber Terrorismo son ataques a las redes de grupos sociales con fines políticos-religiosos mediante el uso de tecnologías de la información (UNODC, 2012).

2.2. Framework de detección y mitigación

Una vez mencionados los crímenes cibernéticos, se toma como referencia a phishing proveniente de la Ingeniería Social, para la representación del marco de trabajo y la posible

solución al utilizar técnicas de Machine Learning, tal como se muestra en la **Figura 2**. De este modo se inicia con la identificación de amenazas y vulnerabilidades, esto genera un riesgo latente que podría desencadenar en un ataque de phishing. Este tipo de ataque posee un ciclo de vida, es decir, las fases que debe cumplir para que el ciberataque logre su desarrollo. A esto se añaden las propiedades que lo caracterizan y que permitirán identificarlo. Por último, se busca una solución al aplicar técnicas de Machine Learning como Naive Bayes y Árboles de Decisión para la detección y mitigación del ciberataque mencionado, a fin de contrarrestar su impacto.

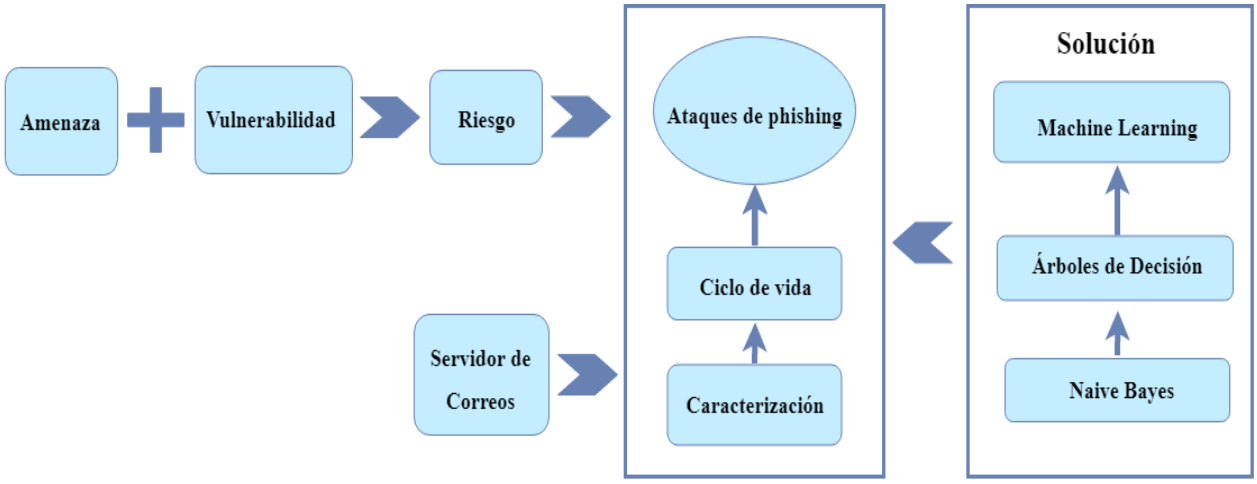


Figura 2. Framework de detección y mitigación

2.2.1. Amenazas

Según la norma ISO 27001 define una amenaza como un evento que puede afectar a los activos de información relacionados con recursos humanos, eventos naturales o fallas técnicas, ataques informáticos externos, infecciones con malware, una inundación, un incendio o cortes de energía eléctrica (ISOTools, 2017). Cabe mencionar que las amenazas se pueden presentar en

diferentes situaciones, ya sea de forma natural, por un agente humano o a causa de la tecnología (Ramos & Hurtado, 2011).

Amenazas Tecnológicas o Informáticas

Son los posibles daños que pueden tener las empresas en pérdidas irrecuperables de datos e información importante, debido a la falta de equipos adecuados exponen a las entidades a ser vulnerables a estafas. Otra amenaza es la gran variedad de programas que dañan los sistemas de manera intencionada conocidos como malware, virus, gusanos, ataques DDoS o también puede originarse un error, agujero o bug.

Amenazas Naturales

Este tipo de amenazas se enfocan en las ocasionadas por la naturaleza, es decir, catástrofes como: erupciones volcánicas, inundaciones, movimientos telúricos, incendios, etc.

Amenazas Humanas

Son amenazas directamente relacionadas con los humanos, personal de una entidad pública o privada, ex empleados, terroristas o intrusos que ocasionan amenazas al sistema, así como: fallos humanos de los usuarios. Además, de acuerdo con la ITU puede ser accidental o intencional, así como activa o pasiva. Una amenaza accidental no es premeditada, es decir, puede ser un fallo físico de un sistema o del software, mientras que una intencionada es realizada por una persona como un acto denominado ataque (ITU, 2007). La amenaza activa es la que ocasiona un cambio de estado, por ejemplo, alteración en los datos o destrucción de equipos físicos y una pasiva no ocasiona ningún cambio de estado, tal como las escuchas clandestinas (ITU, 2007).

2.2.2. Vulnerabilidades

Según la ISO 27001 define una vulnerabilidad como la posibilidad de materializar una amenaza sobre un activo de información (ISOTools, 2017). De manera similar, Gómez Vieites considera que “Es cualquier debilidad en el sistema informático que puede permitir a las amenazas causar daños y producir pérdidas en la organización” (Gómez Vieites, 2014, p.61). Esto se pueden originar debido a defectos de ubicación, la instalación de equipos, la configuración de los sistemas físicos o lógicos y el mantenimiento de los equipos (Gómez Vieites, 2014). Se menciona a continuación otros aspectos que pueden estar ligados a:

- Aspectos organizativos: implica la escasez de políticas de seguridad, la mala definición de las mismas o la falta de actualización.
- El factor humano: presentan poca formación o instrucción en el acceso a los recursos de los sistemas informáticos.
- Condiciones ambientales: corresponde a la deficiencia en las medidas de seguridad física y la poca protección contra ciertos incidentes, así como incendios.

En referencia a lo anterior, se identificaron dos medios de vulnerabilidad que aprovechan los creadores de phishing para realizar ataques de suplantación de identidad: (1) El Factor Humano como: los usuarios o víctimas, y (2) Los Sistemas Lógicos como: los servidores de correo electrónico y Sistemas de Nombres de Dominio o Domain Name System (DNS) que se describen a continuación.

Factor Humano

El factor humano o las personas de una organización representan el eslabón más débil en la seguridad informática, ya que pueden o no cumplir con las instrucciones dictadas (Gómez Vieites, 2014). Además, son propensas a realizar acciones que ocasionan agujeros en la seguridad de la red de una organización, la instalación de software malicioso o la revelación de información sensible a terceros (Gómez Vieites, 2014).

Sistemas Lógicos

Como lo establece Sabri Haddouche (Kaspersky Lab., 2018) los servidores de correo electrónico y DNS presentan vulnerabilidades que son aprovechadas por los delincuentes cibernéticos. Así lo evidenció un estudio de 30 servidores de correo como: Apple Mail, Microsoft Mail, Mozilla Thunderbird, Outlook 2016 y Yahoo Mail, en donde fue eliminado de DMARC y Mailsploit que permitía la protección de phishing. Por esa razón, no lograban validar la originalidad del envío de los correos electrónicos (Kaspersky Lab., 2018), esto ocasionó que los clientes reciban correos legítimos e ilegítimos propagando las campañas de phishing. En el caso de los servidores DNS poseen una memoria temporal, que almacena las resoluciones de los nombres obtenidos. Al consultar en la memoria caché de los servidores correspondientes, los ciberdelincuentes aprovechaban para confundir al servidor y enviar dominios ilegítimos (Labaca Castro, 2011).

2.2.3. Riesgo

La ITU define el riesgo como el efecto negativo que puede resultar de explotar una vulnerabilidad de seguridad, es decir, si se ejecuta una amenaza (ITU, 2007). El objetivo de la

seguridad es la reducción del impacto de un riesgo en acción, ya que, no se puede eliminar por completo el peligro. Para ello es necesario alertar el origen de una amenaza o vulnerabilidad a fin de aplicar medidas necesarias de seguridad como software de antivirus, firewalls, uso de contraseñas complejas, etc. (ITU, 2007).

2.2.4. Ataques de Ingeniería Social – Phishing

La Ingeniería Social (IS) se la conoce como el arte de extraer información confidencial mediante manipulación psicológica (Huber, et al., 2009). En el ámbito informático se define como la acción de manipular individuos, con el fin de inducirlos a la divulgación de información personal que puede ser útil para el ciberdelincuente (Bullée, et al., 2017). Además, la IS no requiere de una gran cantidad de conocimiento técnico. Sin embargo, los delincuentes cibernéticos han logrado el éxito y el incremento de dichos ataques mediante el engaño a los usuarios con aspectos comunes de la psicología humana, como la curiosidad, ingenuidad, etc., con ello han ocasionado grandes pérdidas de información (Huber, et al., 2009). Por este motivo, la Ingeniería Social continúa siendo el método de propagación de ataques informáticos más utilizado por los creadores y desarrolladores de malware, quienes aprovechan estas ventajas para engañar a los usuarios (Borghello, 2009). A fin de conocer el ataque de Ingeniería Social con mayor incidencia durante los dos últimos años, se realizó un análisis preliminar en base a reportes e informes en sitios de revistas de seguridad sobre ataques de Ingeniería Social, el cual indica que phishing es el ataque con mayor uso en el robo de información confidencial también conocido como suplantación de identidad. A continuación, se realiza un resumen de los informes de seguridad informática sobre ataques de phishing.

Los ataques de suplantación de identidad han originado desconfianza en entidades conocidas como es el caso de la plataforma de video streaming Netflix, que ha sido víctima de phishing de manera exponencial. Es decir, los ataques se incrementaron en un 25%, esto la convierte en la marca con mayor imitación durante el cuarto trimestre del 2018 (Onieva, 2019). El informe The State Of Phishing Defense evidencia a phishing en correos de facturación (Belani & Higbee, 2018), y revela que 6 de las 10 campañas de phishing fueron más efectivas en el 2018 con mayor ataque de phishing en los meses de junio y julio, debido a que es un período conocido como el fin del año fiscal para algunas empresas internacionales, esto evidencia que los empleados del área financiera son la comunidad más vulnerable y propensa a los ataques de suplantación de identidad (Belani & Higbee, 2018). Por último, se identificó otro caso de phishing en una nueva campaña detectada por los laboratorios de Panda Security, dirigido a teléfonos móviles, el mismo que se hizo pasar por la aplicación de música Spotify, con la finalidad de robar los datos de acceso de las cuentas de los usuarios (Valle, 2018).

Con lo expuesto en el párrafo anterior, es necesario resaltar que phishing es una técnica de Ingeniería Social utilizada por los ciberdelincuentes para obtener información confidencial como nombres de usuarios, contraseñas e información de las tarjetas de crédito. Para lograr el delito se hacen pasar por una comunicación confiable y legítima, por lo que los usuarios crédulos son engañados para ingresar datos personales y con ello realizar estafas y fraudes (Valle, 2013). Phishing, además, representa alrededor del 77% de todos los ataques orientados a la base social en el 2013 (Hadnagy, 2014). El entorno de ataque de phishing comúnmente es por correo electrónico, sin embargo, también se muestra indicios a través de llamadas telefónicas, mensajes de texto, páginas web y redes sociales (CERT-UK, 2015).

2.2.5. Ciclo de vida de Phishing

La **Figura 3** ilustra el ciclo de vida de un ataque de phishing (Ramos, 2011). (1) Inicia con el análisis de la información que el ciberdelincuente desea obtener, así como: datos de cuentas bancarias, contraseñas, números de tarjetas de crédito, etc.; (2) Una vez definido el propósito se desarrolla el código, es decir, crean páginas falsas idénticas a las legales con el fin de convencer a la víctima que los datos proporcionados están en un sitio legítimo; (3) Continúan con la propagación del ataque, para ello envían correos electrónicos masivos desde una cuenta falsa; (4) La infección se realiza al momento que el usuario abre el correo y accede al link o URL del sitio ilegítimo; (5) Por último, los delincuentes recolectan los datos para generar fraude o vender la información.

Figura 3. Ciclo de vida de phishing

2.2.6. Caracterización de phishing

Se describe a continuación algunas de las características más comunes que presentan los correos electrónicos infectados con phishing, tomado de (Valle, 2013) y (InfoSec, 2002):

- Utilizan nombres de compañías, empresas o instituciones existentes para enviar correos con intenciones fraudulentas que adoptan la imagen corporativa y la funcionalidad del sitio web, con la finalidad de confundir a la víctima;
- Incorporan nombres de empleados reales de una empresa como remitente de los correos falsos. El receptor confirma la veracidad del correo al visualizar el nombre de la persona que trabaja en dicha entidad;
- Emplea direcciones de sitios web con la apariencia original, sin embargo, el correo fraudulento conduce al lector a páginas con contenido e información legal falsa;
- El delincuente aloja el sitio web fraudulento en un servidor, que a su vez recoge la información requerida en un cierto intervalo de tiempo. En algunos casos los ciberdelincuentes amenazan con pérdidas económicas o el bloqueo de la cuenta;
- Algunos correos presentan avisos importantes como la actualización urgente de información personal o una alerta.
- La línea de asunto puede contener caracteres numéricos u otras letras para evitar los filtros de spam;
- El contenido de los mensajes suele ser atractivo en lugar de amenazante, por ejemplo, una recompensa o premio;

- Los correos fraudulentos pueden contener un formulario, para que el destinatario ingrese información personal o financiera. Normalmente implica la ejecución de scripts para el envío de la información a bases de datos o áreas de almacenamiento temporal donde los delincuentes cibernéticos recopilan la información solicitada.

2.2.7. Técnicas de Machine Learning

Según la UNESCO los sistemas de Machine Learning (ML) basados en datos pueden hacer predicciones perfectas siempre que los datos sean similares o muy parecidos (UNESCO, 2019). De manera similar MathWorks define a Machine Learning como una técnica de análisis de datos, que enseña a las computadoras a realizar procesos que pueden ser naturales para las personas y los animales mediante el aprendizaje de la experiencia (MathWorks, 2019). Es por ello que, "El aprendizaje automático (ML) es una rama de la Inteligencia Artificial que aplica sistemáticamente algoritmos para sintetizar las relaciones subyacentes entre datos e información" (Awad & Khanna, 2015, p.1). Se debe considerar que los modelos pueden ser predictivos si pretenden hacer predicciones en el futuro, o descriptivos para la obtención del conocimiento de datos (Smola & Vishwanathan, 2008).

En este contexto la **Figura 6.**, describe el ciclo de modelado de Machine Learning, el mismo que inicia con la etapa de administración de datos en la que se recopila un conjunto de información para el uso. Una vez que se recopila, se exploran los datos para comprender mejor su estructura y significado. Normalmente, necesitan ser limpiados para que sean útiles, esto puede implicar el formateo y la vectorización, es decir, es un proceso destinado a convertir los datos en construcciones matemáticas para que los modelos de ML comprenden. Una vez limpiados los

datos, se preparan para ser cargados en el entorno de programación y finalmente, se dividen en subconjuntos de capacitación y validación.

En la etapa del Modelo de entrenamiento de la **Figura 6.**, se decide el enfoque de aprendizaje, así como: predicción, agrupamiento, etc. Por lo general, se estudian las características disponibles en el conjunto de datos de entrenamiento y se diseñan nuevas en el caso de ser necesario. Finalmente, se realiza la selección del o los algoritmos apropiados para entrenar el modelo. En esta etapa se examina el algoritmo elegido en dos fases: el aprendizaje y la inferencia (Shukla, 2017). La etapa de aprendizaje cumple con el proceso que se muestra en la **Figura 4.** , (i) inicia con el entrenamiento del conjunto de datos, (ii) los datos se transforman en una representación o una lista de vectores, (iii) la lista de vectores se procesa con un algoritmo de aprendizaje, y (iv) se construye el modelo con los parámetros obtenido del algoritmo de aprendizaje (Shukla, 2017).

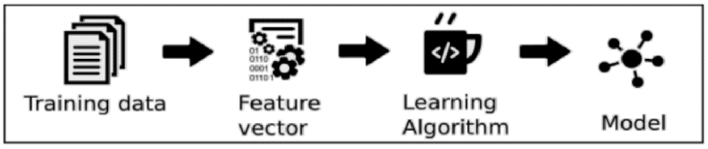


Figura 4. Flujo de aprendizaje

Fuente: (Shukla, 2017)

La fase de inferencia cumple con el proceso que se muestra en la **Figura 5** en donde se emplea el modelo generado en la etapa de aprendizaje. Esto se utiliza para convertir los datos en un vector de características a fin de generar un modelo y obtener la predicción de resultados (Shukla, 2017). El proceso toma menos tiempo que el de aprendizaje y consiste en probar el modelo con nuevos datos y observar el rendimiento, como se muestra en la figura referida.

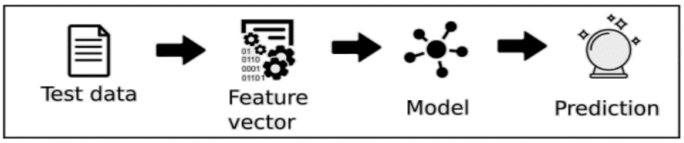


Figura 5. Inferencia de un modelo

Fuente: (Shukla, 2017)

En la tercera etapa de la **Figura 6.**, se evalúan los resultados del modelo con diferentes enfoques o algoritmos. Para lo cual, se califica la tasa de error, tasa de precisión y el porcentaje de detección del modelo con la utilización de la matriz de confusión. Además, la evaluación de la eficiencia del tiempo se realiza mediante una comparación en relación con los resultados de cada algoritmo empleado. A fin de realizar ajustes en los parámetros del modelo hasta cumplir con el objetivo inicial.

En la etapa final, se emplean nuevos datos y se monitorean los resultados. Aquí se realizan las predicciones o deducciones sobre datos que no se consideraron antes o no fueron explorados, en el caso de mejorar el modelo se debe iniciar con el ciclo nuevamente.

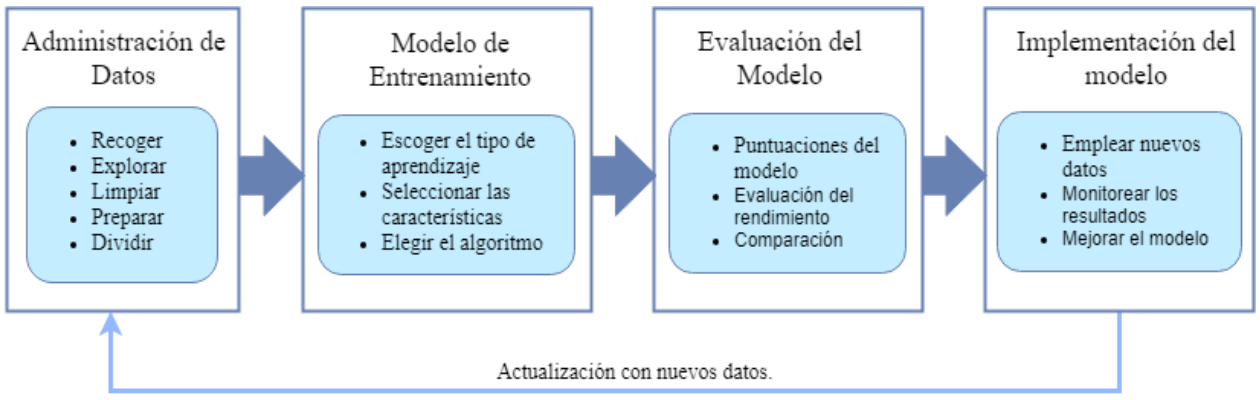


Figura 6. Las 4 etapas del ciclo de modelado de Machine Learning (ML)

Fuente: (Chang, 2017)

2.2.8. Árbol de Decisión

El Árbol de Decisión (DT) es un modelo estadístico que se utiliza como clasificador (Mohammed, et al., 2016). Esta técnica de Machine Learning es empleada para catalogar los datos en clases y representar con una estructura de árbol (Mohammed, et al., 2016). Tiene origen en la raíz del árbol y divide los datos en características con mayor ganancia de información de manera iterativa. Se repite este proceso de división en cada nodo secundario hasta que las hojas estén puras, es decir, que cada nodo pertenezca a la misma clase (Shukla, 2017). Se debe tener en cuenta que los Árboles de Decisión provienen de un método de aprendizaje supervisado, por tal razón requiere un conjunto de datos para el entrenamiento y la generación del clasificador.

La construcción del clasificador de Árboles de Decisión inicia mediante un algoritmo sintetizado de crecimiento como lo ilustra la **Figura 7**, el mismo que permite conseguir una partición del espacio R^d. A su vez el algoritmo entrena mediante un conjunto de datos definidos por la **Ecuación 1**.

$$\mathrm{D_n} = \left\{(\mathrm{A_i}, \mathrm{B_i}) \mid \mathrm{A_i} \in \mathrm{R^d}, \mathrm{B_i} \in \{0, 1 \ldots, \mathrm{K}-1\}, \mathrm{i} = \{1, \ldots, \mathrm{n}\}\right\} \qquad \textbf{Ecuación 1}$$

Conformado por n pares (A_i, B_i), donde A_i es un vector y B_i es la clase asociada a A_i.

```
T = (Conjunto de datos de entrenamiento, (Dn))  /* El árbol inicia con una hoja*/
    while
        seleccionar t de T criterio de selección (t)
        t = (t_L , t_R) criterio de partición (t, t_L , t_R)
        reemplazar t en T con (t_L , t_R)
    end
        podar T
        elegir el subárbol T' de T
```

Figura 7. Algoritmo de crecimiento para Árboles de Decisión

El árbol obtenido corresponde a los nodos internos (preguntas) y las hojas. Además, se debe considerar que el crecimiento del árbol clasificador depende de tres parámetros (Minguillón & Pujol, 2002): (i) Condición de parada: detiene el crecimiento del árbol. Puede depender de la estructura, el número de iteraciones o la profundidad; (ii) Selección del nodo: la función de selección depende de las características, aunque se puede emplear la pureza del nodo o el nodo con mayor población, etc.; (iii) Partición del nodo: Se emplea funciones de impureza, ya que es una característica crítica en el crecimiento del árbol, debido a que, al realizar una mala elección se traslada a los subárboles del nodo. A fin de ilustrar un ejemplo claro del resultado de la creación de un Árbol de Decisión se muestra en la **Figura 8**.

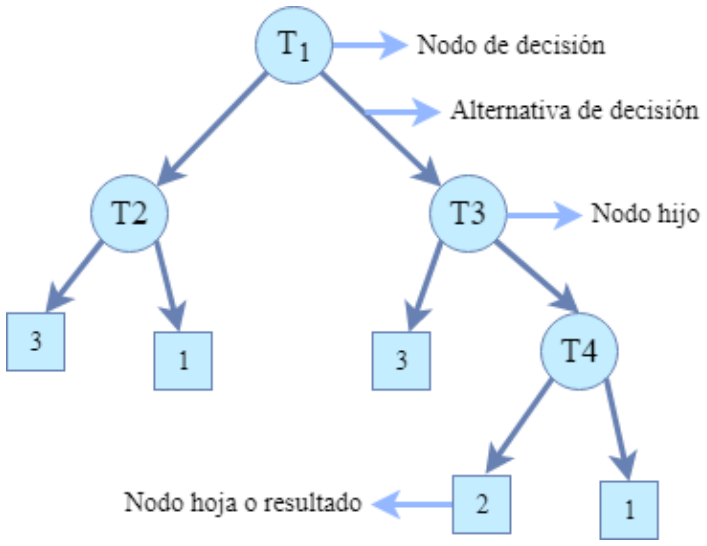


Figura 8. Modelo de un Árbol de Decisión

2.2.9. Naive Bayes

El clasificador Naive Bayes tiene origen a finales de los ochenta, con el objetivo de comparar la capacidad predictiva con la de otros métodos sofisticados (Cestnik, et al. 1987). En la

actualidad se le conoce a Naive Bayes, como un clasificador probabilístico simple, que proporciona precisión en la categorización de los datos en tiempo real (Chandra, et al. 2007). Se basa en la aplicación del teorema de Bayes, es decir, asigna un objeto en la clase que posee mayor probabilidad y las variables o atributos son independientes al valor de la clase principal (Domingos & Pazzani, 1996). Sea $X_1, X_2, \dots, X_n$ variables o atributos que posibilitan la predicción del valor de la clase Z como lo muestra la **Figura 9**. De manera que la estructura del clasificador Naive Bayes se basa en la existencia de un nodo raíz o clase y nodos hojas o atributos que poseen un único nodo padre.

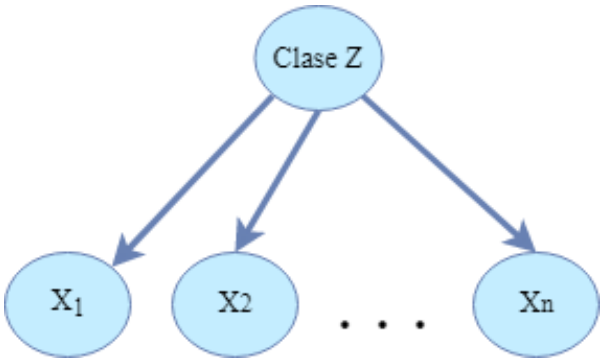


Figura 9. Estructura del clasificador Naive Bayes

Fuente: (Domingos & Pazzani, 1996)

Por esa razón, es conveniente describir el proceso de clasificación de Naive Bayes, en donde se emplea la **Ecuación 2** para obtener las probabilidades y a su vez verificar que los atributos sean independientes a la clase Z, a fin de que los nodos $\{x_1, x, \dots, x_n\}$ pertenezcan a dicha clase.

$$P = (Z = z/x_1, x_2, \dots, x_n) = \frac{P(Z = z)P(x_1/Z = z)P(x_2/Z = z) \dots P(x_n/Z = z)}{P(x_1, x_2, \dots, x_n)}$$

Ecuación 2

Dentro del contexto anterior, es necesario determinar el tipo de parámetros que posee una red Naive Bayes. (1) Atributos discretos: la estimación para la probabilidad de $P(Z)$ y la distribución $P(X_1/Z)$ como lo muestra la **Figura 10**, donde relaciona la frecuencia relativa de ocurrencia dentro del conjunto de datos. De manera que el valor estimado para la probabilidad se obtiene con la **Ecuación 3**:

$$P(x_i/Z = z) = \frac{n(x_i, Z = z)}{n(Z = z)} \qquad \textbf{Ecuación 3}$$

Donde, $n(x_i, Z = z)$ representa el número de casos del conjunto de datos para X_1 cuando considera el valor de x_i y la clase Z toma el valor z y $n(Z = z)$; (2) Atributos continuos: posee dos alternativas: (i) se aplica un método de discretización de la variable continua, y (ii) se emplea una distribución para cada variable predictora, así como el método gaussiano, con la media y varianza para el valor de la variable de la clase.

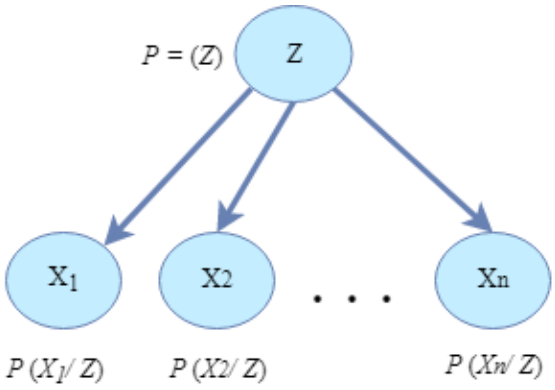


Figura 10. Clasificador Naive Bayes

CAPÍTULO III

ESTADO DEL ARTE

El desarrollo de este capítulo permite conocer el estado del arte sobre las técnicas de Machine Learning en la detección de ataques de Ingeniería Social. Inicia con el desarrollo de las fases del Mapeo Sistemático de Literatura, seguido de la exposición de resultados y finaliza con la identificación de las técnicas de Machine Learning con mayor uso en la detección de ataques de suplantación de identidad.

3.1. Mapeo Sistemático de Literatura

Se empleó el método de investigación conocido como Mapeo Sistemático de Literatura o Systematic Mapping Study (SMS), con el objetivo de conocer de manera general sobre las técnicas de ML empleadas en la detección de ataques de Ingeniería Social. Para ello, se toma como referencia las guías de Kitchenham (Kitchenham & Charters, 2007), se menciona en la **Figura 11**:

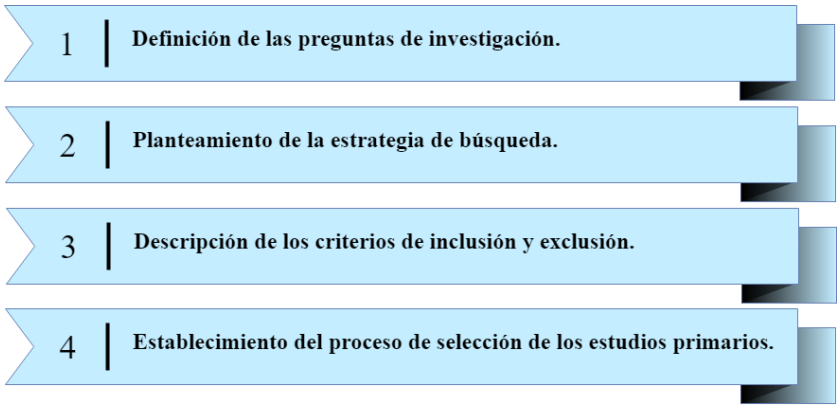


Figura 11. Fases para realizar un Mapeo Sistemático de Literatura

Fuente: (Kitchenham & Charters, 2007)

3.1.1. Definición de las preguntas de investigación

El alcance del SMS se define mediante las preguntas de investigación o Research Questions (RQ). De esta manera se orientó la búsqueda y el SMS, al responder las siguientes interrogantes:

RQ1. ¿Cuáles son las consecuencias de los ataques de suplantación de identidad en la sociedad?;

RQ2. ¿Qué técnicas de ML son empleadas en los estudios primarios para detectar ataques de Ingeniería Social orientados a la suplantación de identidad?;

RQ3. ¿Cuál es la aplicación de las técnicas de ML para detectar ataques de Ingeniería Social?;

La primera pregunta de investigación RQ1, reconoce de manera general el impacto que ocasiona los ataques de suplantación de identidad a nivel global. Tiene por propósito identificar los trabajos relacionados, para la obtención de las palabras claves y la búsqueda bibliográfica. RQ2 permite identificar las técnicas de ML orientadas en la detección de ataques de suplantación de identidad y RQ3 analiza la aplicación de las técnicas de ML dentro del grupo de estudios primarios, a fin de conocer los algoritmos con mayor acogida en la detección de dichos ataques.

3.1.2. Planteamiento de la Estrategia de Búsqueda

Se define en la **Figura 12** la estrategia de búsqueda o las actividades para la obtención de los estudios primarios, acorde a las guías de Kitchenham (Kitchenham & Charters, 2007).

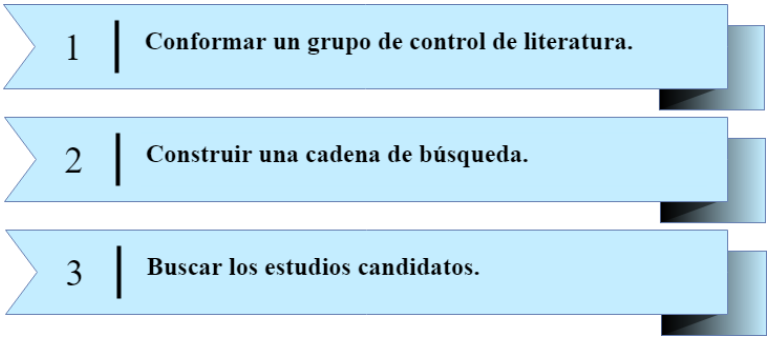


Figura 12. Fases para el Planteamiento de la Estrategia de Búsqueda

Fuente: (Kitchenham & Charters, 2007)

La **Conformación del Grupo de Control** inicia con una revisión de literatura básica, con el propósito de formar un grupo de estudios relacionados. Para ello, se consideraron los siguientes criterios: estudios publicados en revistas y congresos de alto impacto, bases digitales de editores de ciencias de la computación como IEEE Computer Society (http://www.computer.org/), ACM (http://dl.acm.org/), Springer (http:// www.springerlink .com/) y Science Direct (https://www.sciencedirect.com). A fin de realizar una validación cruzada, en donde se obtuvo 6 estudios para el Grupo de Control (GC).

La **Construcción de la Cadena de Búsqueda** se realizó con la identificación de los términos comunes y sus sinónimos. Para lo cual, se buscaron palabras relacionados con ataques de Ingeniería Social y técnicas de ML dentro de los estudios de control, que a su vez se agrupan a través de operadores lógicos: OR para agregar sinónimos y AND para agregar nuevos contextos. La cadena establecida para la búsqueda fue:

("social engineering" OR "social engineering attack" OR "phishing" OR "phishing attack") AND ("machine learning" OR "technical machine learning" OR "automatic learning techniques" OR "automatic learning" OR "supervised machine learning").

La **Búsqueda de Estudios Candidatos** se estableció con las siguientes consideraciones: (i) fuentes de información: artículos y revistas internacionales entre los años 2011 a 2018 dado que se requiere información actual, (ii) publicaciones escritas en inglés, ya que las revistas y congresos internacionales aceptan artículos solo en este idioma, (iii) bases digitales consideradas: Scopus, IEEE, ScienceDirect y ACM. El resultado fue de 183 artículos relacionados con las técnicas de ML en ataques de suplantación de identidad.

3.1.3. Definición de los Criterios de Inclusión y Exclusión

Constituyen las características que deben cumplir los estudios candidatos para ser considerados o no (Kitchenham & Charters, 2007) como parte de los estudios primarios. Para lo cual, se estableció lo siguiente.

Criterios de inclusión: Se consideran los artículos que: (i) Estén relacionados con ataques de Ingeniería Social que empleen técnicas de ML, (ii) Describan la aplicación de una o varias técnicas de ML para la detección o mitigación de ataques de suplantación de identidad. (iii) Enfaticen la importancia de utilizar técnicas de ML en la detección de ataques de Ingeniería Social, y (iv) Expongan los resultados obtenidos al emplear técnicas de ML en la detección y mitigación de dichos ataques.

Criterios de exclusión: Se excluyeron a los artículos que: (i) No presenten relación con ataques de Ingeniería Social y la aplicación de técnicas de ML para contrarrestar dichas amenazas,

(ii) No describan la aplicación de una o varias técnicas de ML en la detección o mitigación de ataques de suplantación de identidad. (iii) No enfaticen la importancia de utilizar técnicas de ML en la detección de ataques de Ingeniería Social, (iv) No expongan los resultados al emplear técnicas de ML en la detección y mitigación de dichos ataques.

Se añadieron filtros generales para el proceso de selección: (i) Los estudios duplicados en congresos y revistas, serán seleccionados aquellos que se hayan publicado en revistas, debido a que el desarrollo es amplio y detallan los resultados de manera minuciosa. (ii) Se excluyeron los libros o documentos debido a que presentan información conceptual.

3.1.4. Proceso de Selección de Estudios Primarios

La selección de los estudios se realizó mediante la aplicación de los criterios de inclusión y exclusión, así como también los filtros definidos en la sección anterior.

Selección de Estudios Se aplicaron los criterios de inclusión y exclusión a nivel del título, resumen y palabras claves. Esto dio como resultado un listado de 183 estudios seleccionados.

Selección de Estudios Primarios Se verificó el cumplimiento de los criterios de inclusión y exclusión en el artículo completo. Como resultado se obtuvo un listado de 49 estudios primarios, de modo que 130 estudios seleccionados fueron descartados.

3.2. Resultados

Los resultados obtenidos del SMS, se divide en tres partes: La primera, se muestra el número de artículos extraídos por año. La segunda, se detalla la frecuencia de aplicación de las

técnicas más utilizadas de ML en el contexto de detección de ataques de Ingeniería Social. La tercera parte, se describen las técnicas más empleadas dentro del grupo estudios primarios.

3.2.1. Técnicas de Machine Learning aplicadas por año

El desarrollo de la revisión sistemática de literatura permitió seleccionar 49 estudios primarios para conocer el estado del arte e identificar las técnicas de ML con mayor aplicación en los últimos ocho años, es decir, desde el año 2011 hasta 2018. En la **Figura 13** se muestra el número de documentos por año de publicación, lo que evidencia que durante el 2011 los investigadores emplearon técnicas de ML para la detección de ataques de suplantación de identidad en un 16%, mientras que durante el 2017 y 2018 se mantuvo una media menor al 50% en comparación al 2011.

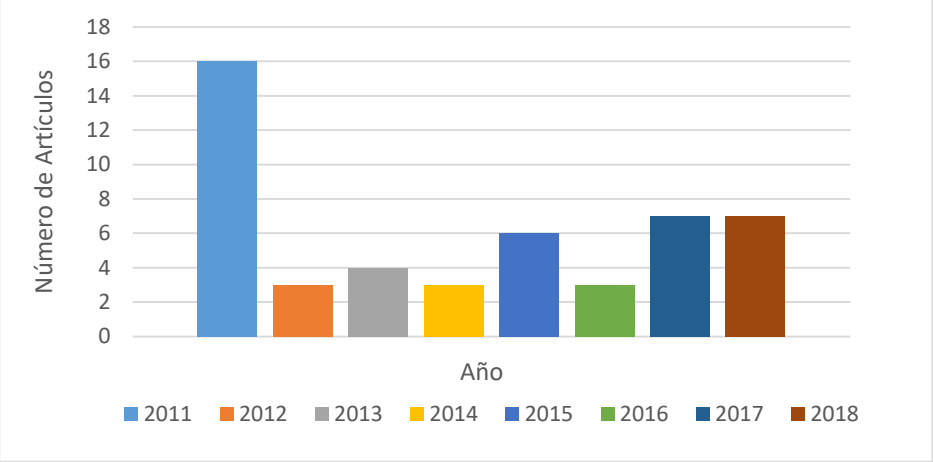


Figura 13. Técnicas de Machine Learning por año

3.2.2. Técnicas de Machine Learning más empleadas

La **Tabla 1** contiene las cuatro primeras técnicas de ML con mayor uso del grupo de estudios primarios. Esto corresponde al uso de SVM y Naive Bayes con 19 veces, mientras que Árboles de Decisión y Random Forest fueron utilizadas 12 veces. Es importante notar que, dentro

de los hallazgos, se identificó que el 18.37% asocian más de dos técnicas de ML para obtener mejores resultados como el caso de (Ke, et all., 2016), fusionaron a Feature Selection, Naive Bayes y Support Vector Machine para la obtención del 97% en la detección de ataques de phishing, mientras que el 81.63% de los estudios, realizaron una comparación previa de las técnicas.

Tabla 1.
Técnicas de Machine Learning con mayor uso

Paper	**Support Vector Machine**	**Naive Bayes**	**Decision Tree**	**Random Forest**
(Abu-Nimeh et al., 2007), (Hamid, Abawajy, & Kim, 2013)	X			X
(Basnet & Sung, 2012)	X	X	X	
(Islam & Abawajy, 2013), (Ke, Li, & Vorobeychik, 2016), (Yamak, Saunier, & Vercouter, 2016), (L'Huillier, Weber, & Figueroa, 2010)	X	X		
(Anandita, Yadav, et al., 2017), (Jain & Gupta, 2018)	X	X		X
(Zhang, et al., 2017), (Liu et al., 2015), (Gyawali, et al., 2011), (Belabed, et al., 2012), (Gowtham & Krishnamurthi, 2014)	X			
(Chen, et al., 2014)	X		X	
(Chung, et al., 2016), (Aburrous, et al., 2010), (Toolan & Carthy, 2010)			X	
(Verma & Rai, 2015), (Verma & Dyer, 2015), (Ma, et al., 2009)	X	X	X	X
(Douzi, et al., 2017), (Ismail, et al., 2014), (Mourtaji, et al., 2017), (Zhan & Thomas, 2011), (Zhang, et al., 2011)		X		
(Ravula, et al., 2011)		X	X	
(Basnet, et al., 2012), (Al-Janabi, et al., 2017)		X		X
(Wu, et al., 2017)			X	X
(Sanglerdsinlapachai & Rungsawang, 2010)		X	X	X
(Wu et al., 2018)	X		X	X
Número de veces utilizadas	19	19	12	12

3.2.3. Descripción de las técnicas de Machine Learning

Se describe a continuación las cuatro primeras técnicas de ML identificadas como las más utilizadas en el grupo de estudios primarios, así como también algunas observaciones sobre la aplicación de cada una de las técnicas.

Support Vector Machine (SVM)

Es un algoritmo de clasificación y análisis de regresión que examina la información sin procesarla, de manera que enlista patrones. Además, SVM es considerada como un clasificador lineal binario no probabilístico, que construye un prototipo para categorizar nuevos datos de prueba (Anandita et all., 2017).

El campo de aplicación de SVM que se identificó en el grupo de estudios primarios es el robo de información personal y fraude, ya que 13 de los 19 estudios detectaron el robo de información, mientras que el valor restante se vinculó en la detección del fraude económico. En cuanto a la aplicación del algoritmo se registró que el 57.89% de los estudios primarios realizaron comparaciones con otras técnicas, el 21.06% prefirieron utilizar la técnica de manera independiente, el 15.79% agruparon técnicas como el caso de (Liu et al., 2015) que combinaron SVM, AdaBoost y K-Nearest Neighbour, mientras que el 5.26% iniciaron con un análisis y luego agruparon las técnicas idóneas. En referencia al objetivo del algoritmo, el 89.47% lo emplea en la detección de ataques de suplantación de identidad, y el 10.53% en la identificación de características para el entorno de aprendizaje del algoritmo clasificador.

El resultado que obtuvieron los autores al aplicar la técnica SVM en la detección de ataques de phishing fueron los siguientes: el estudio de (Liu et al., 2015) con el 70% de aciertos, mientras

que el estudio de (Anandita et all., 2017) se encuentra en el rango del 92.08 al 98.8% en la precisión de detección de contenido legítimo y por último el estudio de (L'Huillier, G. et al., 2010) obtiene el 99.32% considerado como el porcentaje con mayor precisión en la detección de phishing en correos electrónicos.

Naive Bayes

En el grupo de estudios primarios la técnica Naive Bayes se enfocó en la detección de delitos como el robo de información personal y fraude, para lo cual, el 73.68% de los estudios primarios realizaron una comparación de Naive Bayes con otras técnicas de ML como SVM, Árboles de Decisión, etc., el 16.66% realizaron una comparación y adoptaron las técnicas idóneas, el 5.26% los autores fusionaron algunas técnicas de ML para realizar la detección del ataque. Por último, el 5.26% emplearon a Naive Bayes de manera independiente como es el caso de (Ismail, I., et al., 2014), donde tuvieron un 95% de precisión en la detección de ataques de suplantación de identidad.

En relación al análisis descrito en el párrafo anterior se extrajeron las ponderaciones de veracidad en la detección de phishing, así es el caso de (Al-Janabi, M. et al., 2017) donde obtuvieron un 51% de precisión, este es el estudio con el menor porcentaje de aproximación en la identificación del contenido legítimo o ilegítimo se páginas Web, mientras que el porcentaje más alto en la detección lo obtuvo el estudio de (Yamak, Saunier, & Vercouter, 2016), con un 99.6% enfocado a sitios Web.

Árbol de Decisión

El campo de aplicación de Árboles de Decisión es el robo de información personal y el fraude, en entornos de correo electrónico, sitios web o URL. Por esa razón los estudios de (Yusoff & Jantan, 2011) y (Chung, et al., 2016), utilizan la técnica de Árboles de Decisión de ML para detectar ataques de phishing.

En consecuencia, a lo antes mencionado, se exponen los resultados extraídos en la predicción de ataques de suplantación de identidad, como es el caso de (Aburrous, et al., 2010) que obtuvieron un 76.5% al comparar con otras técnicas de ML como Heuristic, AdaaBost, Neural Network y Random Forest. El rango promedio en la detección fue el 86.3 al 97.4% al realizar una comparación previa al desarrollo, mientras que el estudio de (Ma, et al., 2009) obtuvo el 99.5% de predicción en el contenido legítimo e ilegítimo de correos electrónicos.

Random Forest

La técnica Random Forest (RF) fue aplicada para la detección de robo de información personal y fraude, ya que 8 de los 12 estudios que emplean RF se enfocaron en el primer delito, mientras que los 4 estudios restantes fueron aplicados para la detección de fraude. De manera que el 83.33% de los estudios analizaron a RF de manera independiente, el 8.33% acogieron algunas técnicas para obtener mejores resultados en la predicción y el 8.33% realizaron una comparación con un grupo de técnicas seleccionadas. Los resultados obtenidos en cuanto al desempeño de RF en la detección de ataques de phishing muestra que el estudio con mayor aceptación en la precisión es (Basnet, et al., 2012) con el 99.7% en la predicción en sitios Web.

CAPÍTULO IV

DISEÑO E IMPLEMENTACIÓN DEL MODELO DE DETECCIÓN Y MITIGACIÓN

En este capítulo se expone la metodología de desarrollo, la descripción de cada ciclo de vida del proyecto, la aplicación de técnicas de Machine Learning empleadas en la construcción del prototipo y el diseño e implementación del modelo de detección.

4.1. Metodología de desarrollo

Para el diseño e implementación se utilizó la Metodología de Desarrollo Espiral, propuesto por Barry Boehm (Boehm, 1988) considerado como un modelo evolutivo que se acopla a la naturaleza iterativa al realizar prototipos. Su característica principal es el desarrollo rápido de versiones completas. Presenta una secuencia de actividades que pretenden aplicar retrospectiva a cada una de ellas. El ciclo de vida inicia con la viabilidad del proyecto, la siguiente espiral representa los requerimientos, en el tercero se define el diseño y finaliza con la evaluación. Se debe tener en cuenta que cada ciclo se divide en cinco secciones como lo muestra la **Figura 14**:

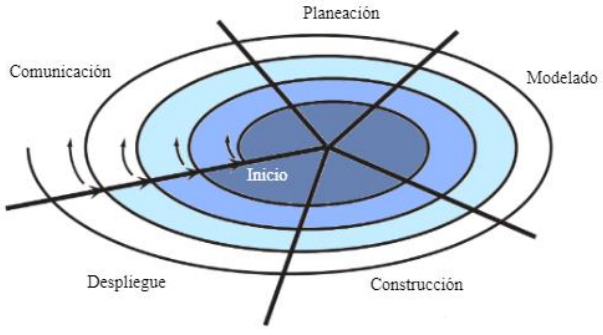


Figura 14. Modelo de espiral común

Fuente: (Pressman, 2010)

4.1.1. Ciclo 1: Viabilidad

La viabilidad del proyecto se especificó mediante el análisis del problema central "Incremento de ataques de suplantación de identidad" descrito en el **CAPÍTULO I**. Por esa razón, se efectuó una recapitulación de los fraudes ocasionados por dichos ataques en los apartados **1.3** y **2.2.4.** Un ejemplo real es el caso de la plataforma de video streaming Netflix, que fue víctima de phishing de manera exponencial durante el cuarto trimestre del 2018 (Onieva, 2019). Además, los informes como el de Tendencias de Seguridad Cibernética en América Latina y el Caribe menciona otros fraudes ocasionados por dicho ataque (ver en **1.3**).

Por lo antes mencionado se hizo una revisión sistemática de literatura (ver **CAPÍTULO III**), a fin de conocer la factibilidad de emplear técnicas de ML en el desarrollo del modelo de detección y mitigación de ataques de suplantación de identidad. Esto se refleja en la **Figura 13**, en donde los dos últimos años se utilizaron técnicas de ML en un 50% en relación al año 2011. Además, se identificó las cuatro técnicas con mayor uso como se muestra en la **Tabla 1**, que permitió seleccionar los clasificadores Naive Bayes y Árboles de Decisión para el entrenamiento y detección de ataques de phishing.

4.1.2. Ciclo 2: Requisitos

En el ciclo 2 se estableció la especificación de requisitos para el desarrollo del modelo de detección y mitigación de ataques de suplantación de identidad. Esto proporciona un mayor entendimiento por escrito sobre la solución del problema. Por tal razón, los requisitos funcionales se describen desde la **Tabla 2** hasta la **Tabla 5**.

Tabla 2.
Requisito funcional RF_01

Id. Requerimiento	**RF_01**
Nombre	Entrenar el algoritmo de detección.
Descripción	Permite el entrenamiento del modelo con datos previamente definidos, acorde a la probabilidad de ocurrencia.
Entradas	Matriz con datos de prueba.
Salidas	Muestra el porcentaje de detección.
Proceso	Leer la matriz con los datos de aprendizaje. Entrenar el algoritmo.
Precondiciones	Tener las características para el algoritmo. Haber definido la matriz con datos de entrenamiento.
Post condiciones	Se obtendrá el porcentaje de detección de ataques de suplantación de identidad.
Efectos colaterales	Procesar pocos datos para el aprendizaje. El algoritmo tendrá un porcentaje bajo en la predicción.
Prioridad	Alta
Rol que lo ejecuta	Equipo de desarrollo

Tabla 3.
Requisito funcional RF_02

Id. Requerimiento	**RF_02**
Nombre	Extraer información de los correos
Descripción	Le permite al modelo agrupar mediante la generación de una matriz de datos reales o características. Considerando que es un entorno controlado.
Entradas	Lista de correos
Salidas	Matriz de características
Proceso	1. Lectura de correos 2. Se aplican filtros 3. Construcción de la matriz de característica.
Precondiciones	Conexión a un servidor de correos. Recepción de correos.
Post condiciones	Se crea un archivo con los datos reales.
Efectos colaterales	1. Lectura errónea de correos. 2. Datos inconsistentes. 3. Matriz incompleta.
Prioridad	Alta
Rol que lo ejecuta	Equipo de desarrollo

Tabla 4.
Requisito funcional RF_03

Id. Requerimiento	**RF_03**
Nombre	Detectar ataque de suplantación de identidad
Descripción	El modelo procesa la información de la matriz, mediante el uso de los clasificadores.
Entradas	Matriz de características de correos.
Salidas	Mensaje que permite saber si el correo se encuentra o no infectado de phishing.
Proceso	1. Ingresar el vector de datos al clasificador. 2. Mensaje con la respuesta de es o no phishing.
Precondiciones	Datos en la matriz de características de correos.
Post condiciones	Mensaje de confirmación.
Efectos colaterales	1. Vector sin datos de características. 2. Predicción en cero.
Prioridad	Alta
Rol que lo ejecuta	Equipo de desarrollo

Tabla 5.
Requisito funcional RF_04

Id. Requerimiento	**RF_04**
Nombre	Envío de correos infectados a cuarentena.
Descripción	El modelo se encarga de enviar los correos infectados a cuarentena.
Entradas	Detección obtenida por los clasificadores de ML.
Salidas	Mensaje de correo puesto en cuarentena.
Proceso	1. Identificar si el correo se encuentra infectado. 2. Re direccionar el correo a una carpeta de cuarentena.
Precondiciones	Disponer de correos. Recibir la matriz de características.
Post condiciones	Correo enviado a cuarentena.
Efectos colaterales	1. Envío de correos no infectados. 2. Correos legítimos puestos en cuarentena.
Prioridad	Alta
Rol que lo ejecuta	Equipo de desarrollo

Al definir los requisitos funcionales del prototipo, es necesario especificar el nivel de prioridad para el desarrollo. Para ello se debe considerar el flujo de aprendizaje y entrenamiento de

ML, esto implica que se distribuyan de la siguiente manera: (1) RF_01, (2) RF_02, (3) TF_03 y (4) RF_04.

Cabe mencionar que el modelo tiene requisitos no funcionales como el rendimiento en el proceso de aprendizaje, el consumo de recursos y el tiempo de respuesta. Debido a que es necesario conocer las condiciones en las que el modelo cumple favorablemente con su desempeño en el aprendizaje y detección.

4.1.3. Ciclo 3: Diseño e implementación

Según (Pressman, 2010), el diseño de la arquitectura constituye la distribución de los datos, los elementos y las interrelaciones entre los componentes, a fin de mostrar la distribución que posee un sistema, aplicación o modelo de software. En este contexto, la siguiente sección se describe las arquitecturas de: el Framework Jupyter que se emplea en el desarrollo del modelo, el entorno controlado del ataque de Ingeniería Social y el modelo como tal para la detección y mitigación de ataques de suplantación de identidad.

4.1.3.1. Arquitectura del Framework Jupyter

La arquitectura que se muestra en la **Figura 15**, se empleó para el desarrollo del modelo propuesto en la **Figura 16**. Lo que corresponde a un sistema basado en la implementación del Kernel como un motor para la ejecución de lenguajes de programación (Python, Ruby, Julia, R, etc.). Esto implica que el proceso debe contener un sistema de colas fundamentado en ZeroMQ, es decir, realiza una comunicación por mensajes a otros sistemas (Jypiter, 2019). El navegador que se utilizó para interactuar con el Framework, logra una comunicación con el servidor a través de un

Websocket para la optimización del tráfico. Además, posee notebooks para el almacenamiento, lectura y modificación de archivos con diferentes extensiones según el entorno de desarrollo (Jypiter, 2019).

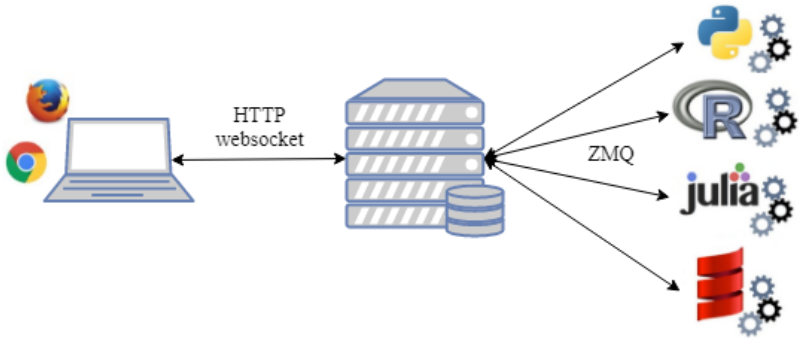


Figura 15. Arquitectura del Framework Jupyter

Fuente: (Zaforas, 2016)

4.1.3.2. Arquitectura del modelo de detección y mitigación

En la **Figura 16**, se aprecia la arquitectura que corresponde al entorno controlado del ataque de suplantación de identidad. Inicia con la simulación del envío de correos por el ciberdelincuente mediante el Protocolo Simple de Transmisión de Correo o Simple Mail Transfer Protocol (SMTP) a un servidor de correos víctima. Estos son administrados por el Protocolo de Acceso a Mensajes de Internet o Internet Message Access Protocol (IMAP) que recibe el computador del usuario. El modelo extrae la data de los correos con el algoritmo de clasificación Naive Bayes y continúa con el análisis de la información mediante el clasificador de Árboles de Decisión para la detección de correos ilegítimos. Por último, si el mensaje que recibió la víctima corresponde a un correo infectado se emplean medidas de prevención necesarias (mitigación). Cabe mencionar que el

modelo se desarrolló en base a una rama de la Inteligencia Artificial que es Machine Learning, empleando algoritmos de aprendizaje supervisado como Naive Bayes y Árboles de Decisión.

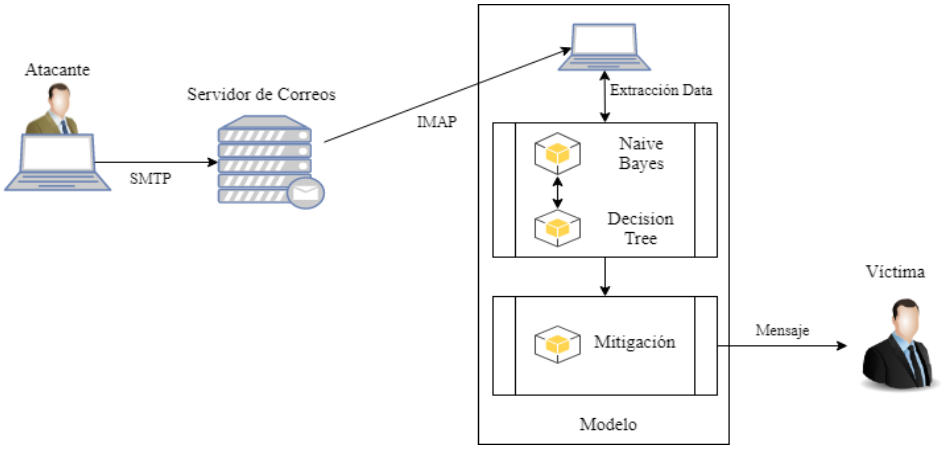


Figura 16. Arquitectura del ambiente controlado del ataque

La arquitectura del modelo se explica en la **Figura 17**. Este, inicia con la lectura de los correos recibidos y realiza la extracción de la información requerida. La data ingresa al algoritmo de Naive Bayes, a fin de extraer las características y categorizar los datos. El flujo continúa con el envío de la información al algoritmo de Árboles de Decisión para la detección del ataque. Por último, se efectúa una revisión de los resultados a fin de enviar a cuarentena los correos con phishing.

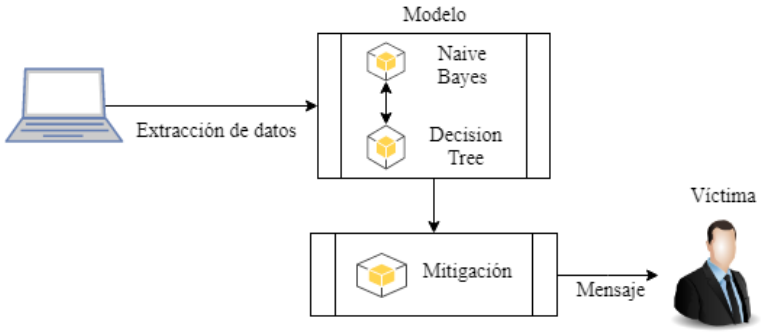


Figura 17. Arquitectura del modelo

4.1.3.3. Diagrama de Secuencia

Según (Sommerville, 2011), los diagramas de secuencia se utilizan principalmente para realizar el modelado de las interacciones entre los actores y los objetos. Además, involucra la relación entre los objetos durante una tarea determinada. De acuerdo con esta definición, se muestra a continuación dos diagramas de secuencia que corresponden al entrenamiento del modelo y la detección de correos infectados.

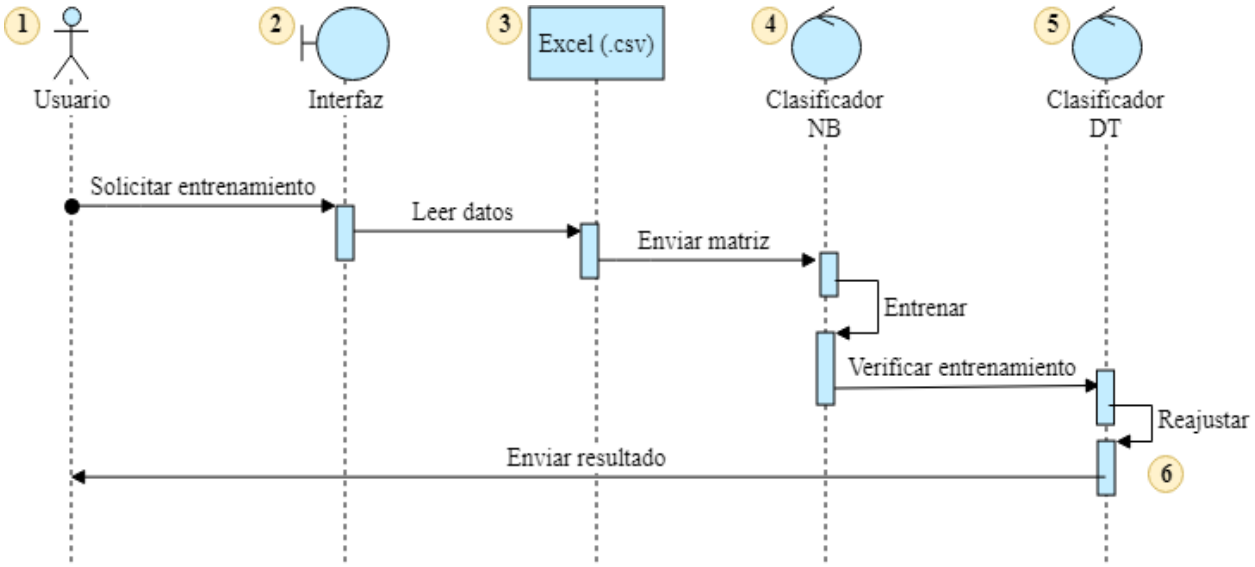


Figura 18. Diagrama de secuencia para el entrenamiento del modelo

El diagrama de secuencia de la **Figura 18**, muestra las acciones que cumple el modelo antes de realizar la detección. (1) Inicia con una petición del usuario, (2) solicita el entrenamiento del modelo mediante una interfaz gráfica. (3) Esta requiere la lectura de los datos o las características definidas mediante la matriz de datos que se encuentra almacenada en un archivo de Excel con extensión csv. (4) Luego se envía los datos al Clasificador NB o Naive Bayes para el entrenamiento del modelo. (5) Si el porcentaje de predicción es menor a 85%, se realiza una verificación de

reajuste con el Clasificador DT o Árboles de Decisión y (6) envía un mensaje de respuesta con el porcentaje de predicción que alcanzó el algoritmo en el entrenamiento.

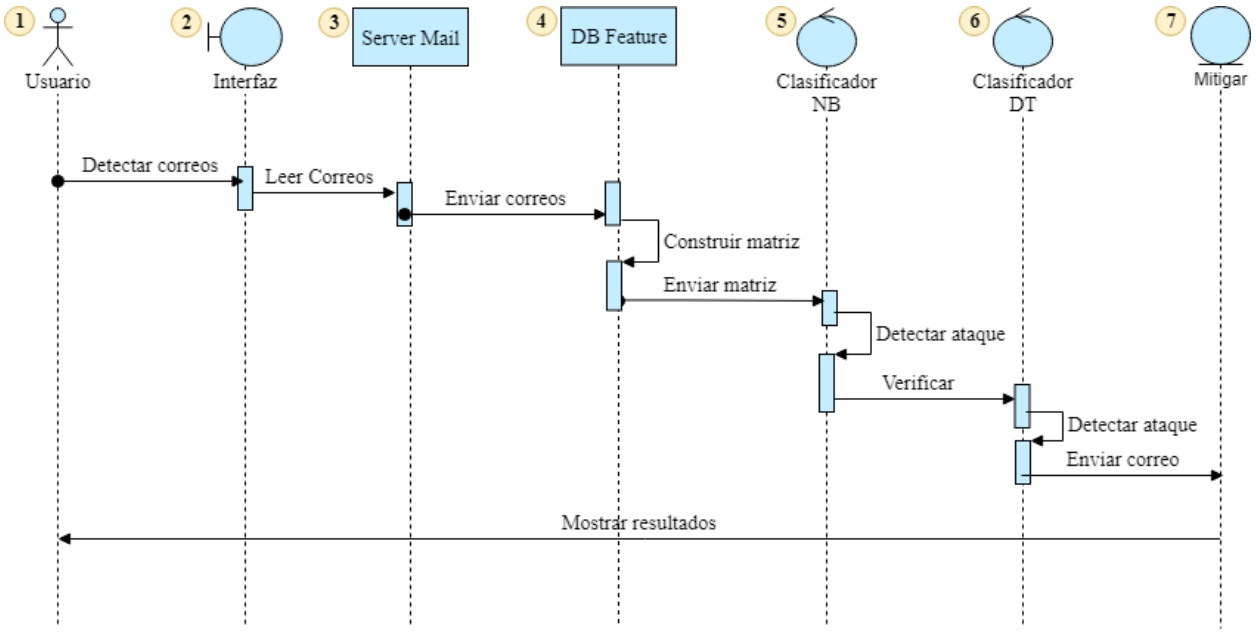


Figura 19. Diagrama de secuencia para la detección y mitigación

La **Figura 19** representa el diagrama de secuencia que corresponde a la detección y mitigación de correos electrónicos. (1) Inicia con la petición del usuario de detectar phishing en los correos que se encuentran en la bandeja de entrada, (2) mediante una interfaz gráfica. (3) Solicita la lectura de los correos al servidor. (4) Los mails son enviados a una base de datos feature, en donde se extraen 18 características de cada correo y se genera una matriz. (5) Dicha matriz se envía al Clasificador NB para el análisis de la data. (6) Una vez terminado interviene el clasificador DT en la detección y (7) en el caso de que el correo se encuentre infectado de phishing se envía a cuarentena (mitigar). Por último, retorna un mensaje con el resultado de cada correo analizado.

4.1.3.4. Diagrama de Clases

Según (Sommerville, 2011) y (Pressman, 2010), el diagrama de clases representa los objetos con sus atributos, operaciones y relaciones o asociaciones con otras clases. Además, posee una visión estática o estructural de un sistema. En base a estos preceptos, la **Figura 20** muestra el diagrama de clases del modelo de detección y mitigación de ataques de suplantación de identidad.

A continuación, se describen algunas de las clases definidas en el diagrama de la **Figura 20**. La clase *FrameDetection* está compuesta por *Tkinter* que contiene atributos y métodos de componentes gráficos para la construcción de la interfaz que utiliza el usuario en la interacción con el prototipo de la aplicación. La lectura del archivo Excel se realizó con los métodos declarados en la clase *Pandas*. Mientras que las clases *Naive Bayes* y *Árboles de Decisión* se encargaron de obtener los datos de los correos, entrenar y detectar en base a la información adquirida. La lectura de los correos electrónicos se realizó mediante la clase *Imaplib*, ya que posee métodos como: login() para el inicio de sesión, imap_ssl() permitió la conexión con el protocolo IMAP4 al servidor de correos, el método select() obtuvo los correos y search() efectuó la búsqueda de los mismos, con la finalidad de realizar la conexión y acceder a la cuenta de correo. Por último, la clase *Matplotlib* se empleó para graficar los resultados.

Cabe destacar que el diagrama de clases representa la estructura, relaciones y atributos que requiere el prototipo con el servidor de correos, el envío masivo de mensajes infectados para el entorno controlado de detección y los algoritmos de clasificación de ML. Por esa razón, el diagrama relaciona las clases con todos los objetos definidos para el desarrollo.

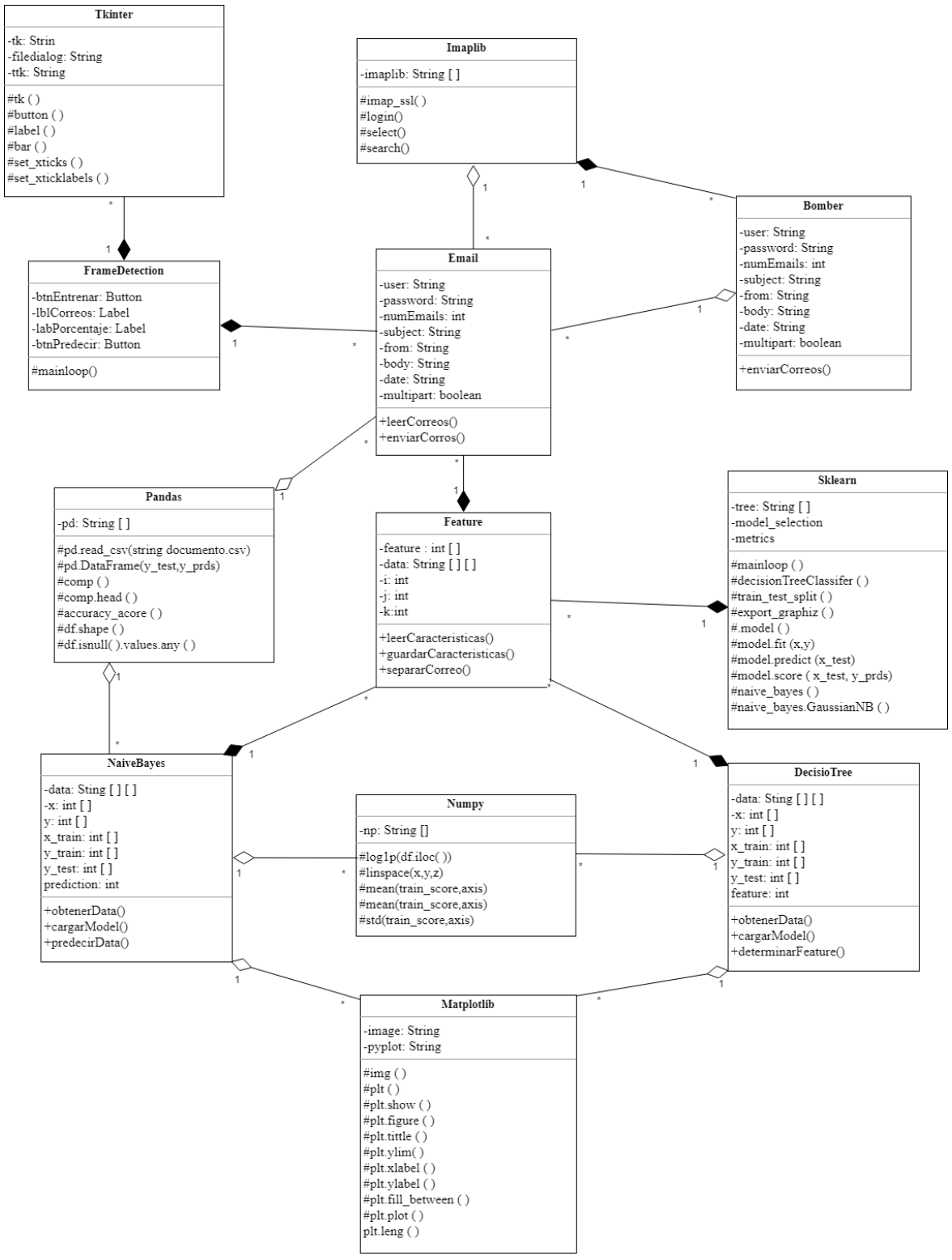


Figura 20. Diagrama de Clases

4.1.3.5. Obtención de Data para el entrenamiento y detección

La matriz de datos de la **Figura 21** se empleó en la etapa de entrenamiento del modelo. Esta se generó con información para 100 correos con todos los posibles casos de ocurrencia que se pueden presentar en los correos con o sin phishing. Cabe mencionar que se utiliza la teoría de Pareto (Delers, 2016) con los 100 correos previamente definidos, lo que refleja que el 20% de las causas producen el 80% de los efectos. En otras palabras, el 20% de la matriz de datos es responsable del 80% de la detección de ataques de suplantación de identidad. Por tal razón, el 20% de los datos son importantes para el entrenamiento del modelo.

1	F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,F13,F14,F15,F16,F17,F18,F19
77	0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
78	1,1,1,0,0,1,0,0,1,1,1,1,0,0,0,0,0,0,0
79	1,1,1,0,0,1,0,0,1,1,1,0,1,0,0,0,0,0,0
80	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0
81	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0
82	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0
83	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
84	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,1,1,0
85	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0
86	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0
87	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0,0
88	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0
89	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,1,0
90	1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
91	1,1,1,0,0,0,1,0,1,1,1,1,0,0,0,0,0,0,0
92	1,1,1,0,0,0,1,0,1,1,1,0,1,0,0,0,0,0,0
93	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0
94	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0
95	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0
96	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
97	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,1,1,0
98	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0
99	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,1,0
100	1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,0
101	1,1,1,1,0,1,0,1,0,0,0,0,0,0,0,1,1,1,0

Figura 21. Matriz de datos .csv de pruebas

La obtención de data para la detección de ataques de Ingeniería Social se realizó en un entorno controlado mediante el envío y recepción de correos electrónicos. La generación de correos infectados con phishing para esta etapa se crearon mediante la extracción de enlaces del servidor PhishTank (OpenDNS, 2005), debido a que es un sistema de verificación de phishing que se basa en la comunidad informática, en donde los usuarios envían enlaces ilegítimos y otros realizan una votación o valoración respecto al nivel de fraude que ocasionó. La **Tabla 6** recopila información sobre la extracción de los enlaces que permitió la generación de 1325 correos falsos.

Tabla 6.
Información enlaces extraídos con PhishTank

Enlaces con phishing	**15000**
Fecha de extracción desde	10 /10/2008
Fecha de extracción hasta	25/01/2019
Número de correos generados con phishing	1325

4.1.3.6. Clasificador de Naive Bayes (NB)

El proceso de aprendizaje o entrenamiento se realizó mediante el clasificador NB que se muestra en el diagrama de flujo de la **Figura 22**. Inicia con la preparación de la data de las características de los correos electrónicos que se encuentran almacenados en un archivo Excel .csv. Luego realiza el proceso de pre-entrenamiento que tendrá como resultado la obtención del porcentaje de predicción. El resultado se valida con un rango menor al 95% y mayor al 90%, debido a que los estudios primarios valoraron como aceptable el rango propuesto para el proceso de entrenamiento. Si cumple con la condición toma el valor de las características. Caso contrario, realiza un reajuste y regresa al proceso de pre-entrenamiento.

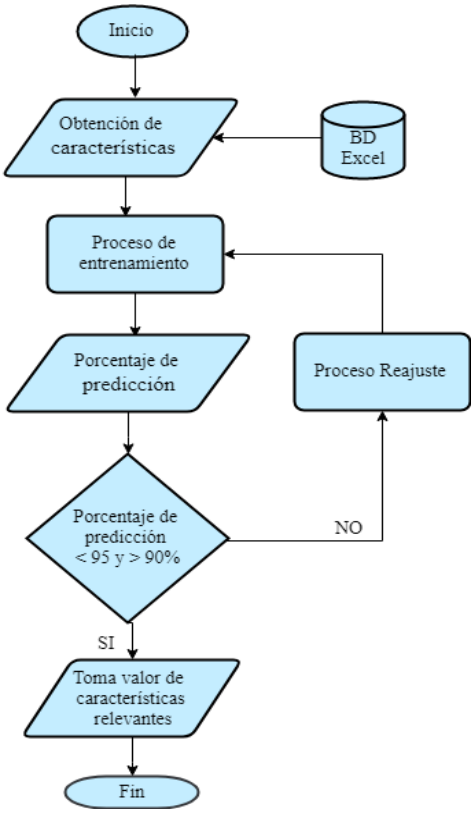


Figura 22. Diagrama de Flujo Naive Bayes

4.1.3.7. Clasificador de Árboles de Decisión

El clasificador de Árboles de Decisión se emplea en la detección de ataques de Ingeniería Social como lo muestra el diagrama de la **Figura 23**. Inicia con la obtención de la data que se encuentra almacenada en un archivo Excel csv. Continúa con el proceso de entrenamiento hasta obtener el porcentaje de predicción que permitirá la detección de los ataques de phishing. El valor obtenido debe pertenecer al rango menor a 100% y mayor al 95%, en el caso de cumplir con la condición se obtiene el modelo de detección, caso contrario se realiza un ajuste y regresa al proceso de entrenamiento.

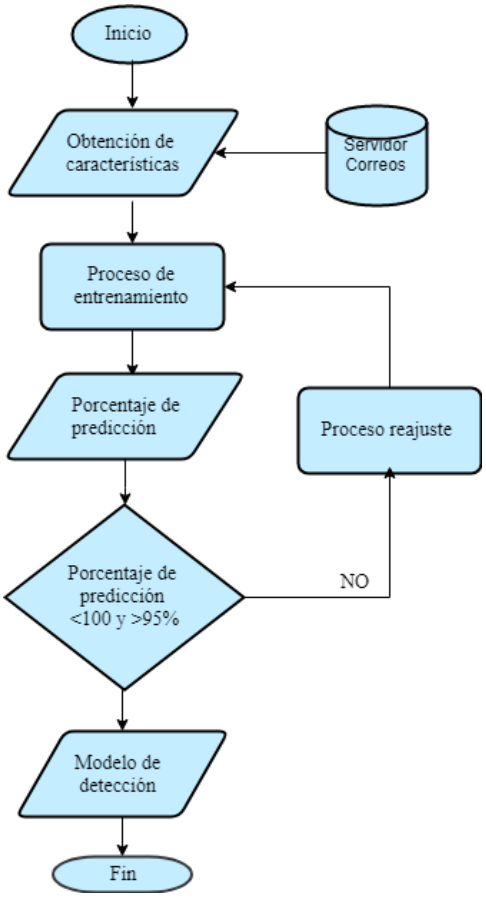


Figura 23. Diagrama de Flujo Arboles de Decisión

4.1.3.8. Selección de características

Las características se obtuvieron mediante una síntesis que se realizó en conjunto con el mapeo sistemático de literatura desarrollado en el **CAPÍTULO III**. En donde se recapituló una lista amplia de características acorde al medio de análisis, así como características para el contenido de una página web, el correo electrónico, de imágenes o archivos adjuntos.

El resultado del párrafo anterior se describe en la **Tabla 7**, que muestra las características que se extrajeron al realizar el estado del arte. Los atributos son: F1 representa la posibilidad de

una URL en el cuerpo del correo; F2 la presencia del caracter especial arroba en la URL; F3 equivale a la dirección válida de emisión de un correo, es el atributo con mayor influencia ya que los correos infectados no poseen este atributo; F4 analiza el número de puntos en la URL; F5 evalúa el tamaño de la URL que no debe superar a 74; F6 y F7 verifican el tipo de protocolo de transferencia de datos; F8 constituye si el cuerpo del correo posee el nombre del propietario de la cuenta; F9, F10, F11, F12, F13, F14 y F15 se basan en la identificación de palabras claves o aquellas que se utilizan con mayor frecuencia en los correos infectados con phishing; F16 valida la presencia de JavaScript; F17 el uso de CSS y F18 expone si el correo presenta un evento que podría dirigir al usuario a otro enlace oculto o ilegítimo.

Tabla 7.
Características seleccionadas

Características	**Alias**
Posee URL	F1
Posee @	F2
<joe@bloggs.com>.Formato de correo	F3
Número de puntos en la URL > 4	F4
Tamaño de URL>=74	F5
https en la URL	F6
http en la URL	F7
Nombre del usuario en el cuerpo del correo	F8
Las palabras “clic aquí” o “actualizar”	F9
Las palabras “confirmar, verificar, proteger, notificar, registrar, inconveniente, vigente, alerta”	F10
Las palabras “seguridad, inicio de sesión, banco, cuenta, actualización, incluir, webs, en línea”	F11
La palabra “suspensión” en el asunto	F12
La palabra “verificar” en el asunto	F13
La palabra “débito” en el asunto	F14
La palabra “banco” en el asunto	F15
La utilización de JavaScript o no.	F16
El uso de CSS o no	F17
Evento onclick	F18

4.1.3.9. Diseño de la interfaz del modelo de detección y mitigación

Se detalla a continuación los requisitos funcionales del modelo. RF_01: Entrenar el algoritmo de detección, este requisito se cumple mediante el desarrollo de una interfaz como se muestra en la **Figura 24**. Esta posee el [Botton Entrenamiento] que llama al método entrenar() de los dos clasificadores de Machine Learning NB y DT.

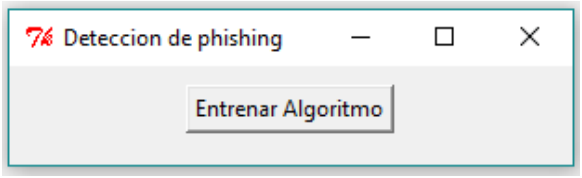


Figura 24. Pantalla para solicitar entrenamiento

Una vez que se solicita el entrenamiento de los algoritmos. Se presenta la curva de aprendizaje que obtuvo para el primer clasificador NB como se evidencia en la **Figura 25**. El cual, logra un aprendizaje del 90% (curva verde) y es necesario el reajuste del modelo.

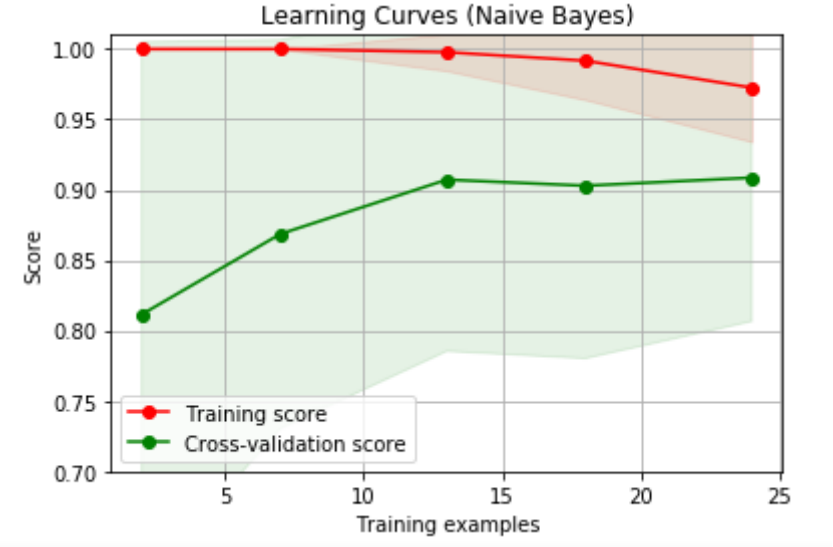


Figura 25. Curva de aprendizaje de Naive Bayes

Se realizó lo solicitado en el párrafo anterior, un reajuste. Para ello se empleó el clasificador de Árboles de Decisión, que obtuvo como resultado en la nueva curva de aprendizaje el 96.77% de predicción, como se puede visualizar en la **Figura 26**. Además, cabe señalar que el aprendizaje inicia en un 92.5% y logra la estabilidad del aprendizaje con el 96.77% al entrenar con la matriz de características de 100 correos.

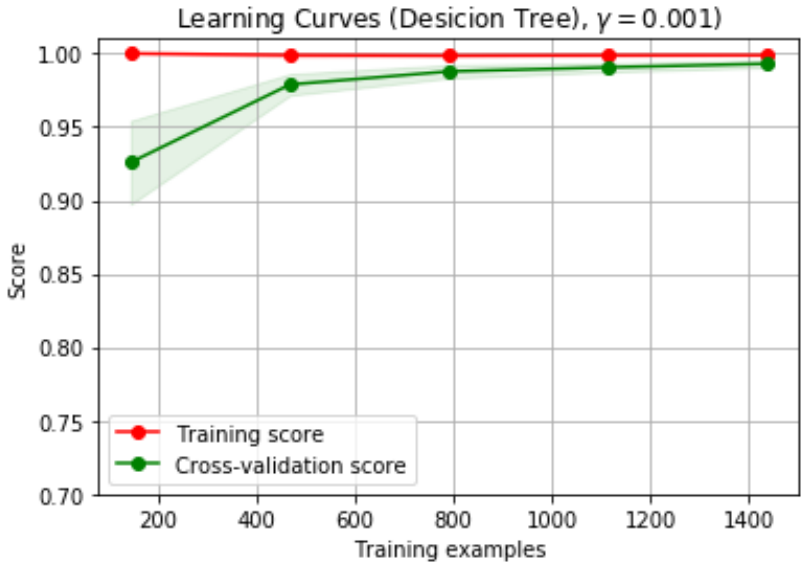


Figura 26. Curva de aprendizaje de Árboles de decisión

Los resultados se muestran en la interfaz de la ***Figura 27***, que detalla el número de características empleadas en la etapa de aprendizaje y el porcentaje de predicción que corresponde al 96.77%. Este valor se emplea en la detección de los correos generados en el entorno controlado. Para ello, la interfaz de la ***Figura 27*** posee un botón, para la detección de los correos. Al dar clic, internamente se llama a los métodos correspondientes, así como los clasificadores de ML, la lectura de correos y la obtención de la matriz de características.

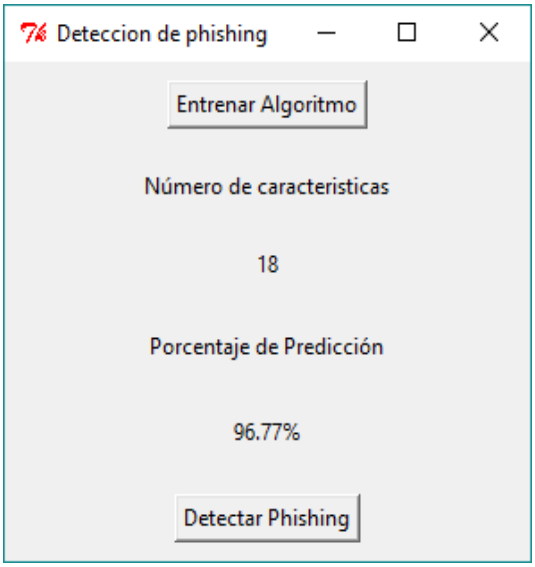


Figura 27. Interfaz de resultados de entrenamiento

Como se mencionó en el párrafo anterior, al presionar el botón Detectar Phishing se procede con la ejecución del método descrito en la **Tabla 8**. En donde se obtienen los correos de la cuenta de Gmail creada para el desarrollo del ambiente controlado, que a su vez contiene un total de 1325 correos, de los cuales 530 correos se encuentran infectados con phishing y 795 no poseen phishing.

Tabla 8.
Método leerCorreos()

```
def leerCorreos():
    import numpy as np
    user =""
    password =""
    atributos = 18
    valor=0
    M = imaplib.IMAP4_SSL('imap.gmail.com')
    M.login('tesispruebas50@gmail.com', 'tesispruebas')
    M.select()
    yp, message_numbers = M.search(None, 'ALL')
    count = len(message_numbers[0].split(" "))
    print ('Numero de correos encontrados ')
    print count
    matriz_mails=np.zeros((19, count))  # Matriz de ceros
atributos=['F1','F2','F3','F4','F5','F6','F7','F8','F9','F10','F11','F12','F13','F14','F15','F16','F17','F18']
    f = open ('mail.txt','w')
    columna=0
```

```
i=0
fila =-1
matrizFinal=[]
for num in message_numbers[0].split():
        matriz=[]
        typ, data = M.fetch(num, '(RFC822)')
        header = email.message_from_string(data[0][1])
        #Separar el mail
        correo=separarCorreo(header['From'])
        fila=fila+1
        payload=header.get_payload()
        body=extract_body(payload)
        if header['Subject'] is None:
           header['Subject']="******Vacio******"
           texto=header['Subject']
           matriz.insert(0,0)
           matriz.insert(1,0)
        else:
           texto=header['Subject'].decode('utf-8')
           matriz.insert(0,1)
           if texto.find("@") >= 0:
              matriz.insert(1,1)
           else:
              matriz.insert(1,0)
        if header['From'] is None:
           header['From']="******Vacio******"
           envia=header['From']
           matriz.insert(2,0)
        else:
           envia=header['From']
           if (envia.find('<') > 0 and envia.find('>') > 0):
              matriz.insert(2,1)
           else:
              matriz.insert(2,0)
        if texto.find(".")>=4:
              matriz.insert(3,1)
        else:
           matriz.insert(3,0)
        if len(texto)>74:
           matriz.insert(4,1)
        else:
           matriz.insert(4,0)
        if body.find("https")>0:
           matriz.insert(5,1)
        else:
           matriz.insert(5,0)
        if body.find("http")>0:
           matriz.insert(6,1)
        else:
           matriz.insert(6,0)
        if body.find("Bryan")>0:
           matriz.insert(7,1)
        else:
           matriz.insert(7,0)
```

```
            if (body.find("clic")>0 or texto.find("actualizar")>0) :
                matriz.insert(8,0)
            else:
                matriz.insert(8,1)
            if (body.find("confirmar")>0 or body.find("verificar")>0 or body.find("proteger")>0 or
body.find("notificar")>0 or body.find("registrar")>0 or body.find("incoveniente")>0 or body.find("vigente")>0
or body.find("alerta")>0) :
                matriz.insert(9,0)
            else:
                matriz.insert(9,1)
            if (body.find("seguridad")>0 or body.find("banco")>0 or body.find("cuenta")>0 or
body.find("actualizacion")>0) :
                matriz.insert(10,0)
            else:
                matriz.insert(10,1)
            if (body.find("suspención")>0 or body.find("suspención")>0 or body.find("SUSPENCION")>0 or
body.find("suspención")>0 or body.find("Suspencion")>0 ) :
                matriz.insert(11,1)
            else:
                matriz.insert(11,0)
            if (body.find("Verificar")>0 or body.find("verificar")>0 or body.find("VERIFICAR")>0) :
                matriz.insert(12,0)
            else:
                matriz.insert(12,1)
            if (body.find("Débito")>0 or body.find("Debito")>0 or body.find("DEBITO")>0) :
                matriz.insert(13,1)
            else:
                matriz.insert(13,0)
            if (body.find("Banco")>0 or body.find("BANCO")>0 or body.find("Banco")>0) :
                matriz.insert(14,1)
            else:
                matriz.insert(14,0)
            if (body.find("script")>0) :
                matriz.insert(15,1)
            else:
                matriz.insert(15,0)
            if (body.find("html")>0) :
                matriz.insert(16,1)
            else:
                matriz.insert(16,0)
            if (body.find("onclick")>0) :
                matriz.insert(17,1)
            else:
                matriz.insert(17,0)
            if header['Date'] is None:
                header['Date']="******Vacio******"
                fecha=header['Date']
            else:
                fecha=header['Date']
            data2 = {'F1': [matriz[0]],'F2': [matriz[1]],'F3': [matriz[2]],'F4': [matriz[3]],'F5': [matriz[4]],'F6':
[matriz[5]]
                  ,'F7': [matriz[6]],'F8': [matriz[7]],'F9': [matriz[8]],'F10': [matriz[9]],'F11': [matriz[10]],'F12':
[matriz[11]]
```

```
                ,'F13': [matriz[12]],'F14': [matriz[13]],'F15': [matriz[14]],'F16': [matriz[15]],'F17':
[matriz[16]],'F18': [matriz[17]]}
            df2 = pd.DataFrame(data2,columns =
['F1','F2','F3','F4','F5','F6','F7','F8','F9','F10','F11','F12','F13','F14','F15','F16','F17','F18'])
            df2.to_csv('example2.csv')
            if fila>0  and fila<1326:
                print ("Correo #"+num)
                print ("Array envia")
                print ("Envia: "+envia)
                verificar(fila)
```

Al emplear el método definido en la **Tabla 8**, el resultado fue la extracción de la data de los correos, para lo cual se necesitó efectuar la conexión al servidor de correos. Dicha conexión se realizó mediante el protocolo IMAP4 y los métodos mencionados en el apartado **4.1.3.4.** La estructura del correo es un objeto como se muestra en la **Figura 28**, con la siguiente organización: M: [#correo,From,Date,Body]. En esta etapa se empleó el requerimiento RF_02: Extraer información de los correos.

Figura 28. Data de un correo de Gmail

La información recopilada en la clase Email extrae las 18 características (ver en **4.1.3.8**), para la construcción de la matriz. La

Tabla *9* es una muestra de la generación de la matriz de atributos que representa un arreglo con la información de las 18 características. El resultado son los valores binarios para cada uno de los atributos, con equivalencia de 1 si posee la particularidad de esa columna o 0 si no la posee. Este proceso se realiza hasta obtener la información de todos los correos que se encuentran en la cuenta de Gmail.

Tabla 9.
Matrices de características

Correo #1 Arreglo envía [1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0] Correo #2 Arreglo envía [1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0]

Una vez conformada la matriz con la información pertinente de los correos electrónicos, se emplea el método de la **Tabla 10** para identificar si el correo se encuentra infectado o no con phishing. Este método realiza la lectura de un archivo Excel .csv, hasta el correo 1325. Luego verifica la data del correo e inicia con el análisis. Continúa con la generación del modelo y la predicción del ataque, para ello aplica el clasificador de Árboles de Decisión. Finaliza con una respuesta de si o no es phishing. En esta etapa se empleó el requisito RF_03: Detectar ataques de suplantación de identidad.

Tabla 10.
Método de verificar el correo

```
Algoritmo verificar
  Repetir
    Leer caracteristicas_csv
  Hasta que caracteristicas_csv=1325
  x<-caracteristicas
  z <-resultado
  x_entrenamiento<-train_test_split( x,z,0.30,18)
  x_prueba<-train_test_split( x,z,0.30,18)
  y_ entrenamiento <- entrenamiento _prueba( x,z,0.30,18)
   y_test<- entrenamiento _prueba( x,z,0.30,18)
   model<-Arboles_Desicion ( 1,5)
   model_fit<-ajuste(x_entrenamiento,y_entrenamiento)
   resultado<-prediccion(x_prueba,y_prueba)
  Si resultado==0 Entonces
     Escribir "No Phishing"
  SiNo
     Escribir "Phishing"
  Fin Si
FinAlgoritmo
```

Por último, se alcanzaron los siguientes resultados en la detección de cada correo como se muestra a continuación en la **Tabla 11**. Inicia con el número de correos analizados, seguido del emisor del correo, y por último indica si “Es Phishing” o “No es Phishing”. Adicional se muestra el número de correos infectados con phishing.

Tabla 11.
Resultados al detectar con el modelo

```
****************************************
Correo #1240
Envia: "Debian Bug Tracking System" <owner@bugs.debian.org>
Es Phishing
('#Phishing ', 528)
('#No Phishing ', 712)
****************************************
Correo #1241
Envia: Debian testing watch <noreply@release.debian.org>
Es Phishing
('#Phishing ', 529)
('#No Phishing ', 712)
****************************************
Correo #1242
Envia: Ansgar <ansgar@debian.org>
Es Phishing
```

```
('#Phishing ', 530)
('#No Phishing ', 712)
****************************************
Correo #1243
Envia: "Debian Bug Tracking System" <owner@bugs.debian.org>
No es Phishing
('#Phishing ', 530)
('#No Phishing ', 713)
****************************************
```

CAPÍTULO V

EVALUACIÓN Y VALIDACIÓN DE RESULTADOS

En este capítulo se describen los resultados que se obtuvo al evaluar y validar el modelo de detección y mitigación. Luego se presenta el procedimiento estadístico correspondiente a la matriz de confusión, la misma que permite conocer el grado de desempeño y la precisión del modelo en la detección de ataques de suplantación de identidad. Así mismo se muestra la validación del modelo mediante una comparación con otros clasificadores de Machine Learning, tales como Random Forest, Regresión Logística y el Clasificador Ficticio. Finaliza con un reporte sobre el rendimiento y consumo de recursos en dos computadoras utilizadas.

5.1. Resultados

La evaluación de resultados se divide en dos fases, la primera corresponde al proceso de entrenamiento del modelo y la segunda es la detección de ataques de Ingeniería Social. A continuación, se muestra los siguientes resultados obtenidos por el modelo de detección y mitigación.

5.1.1. Etapa de entrenamiento del modelo

Cabe mencionar que se realiza una comparación con emails infectados y no infectados, ya que en esta fase los datos de entrenamiento son conocidos, así como también el resultado que debe alcanzar el modelo.

Dicho lo anterior, se inicia con el entrenamiento del modelo en base a 100 correos conocidos, en donde se obtuvo lo siguiente como se ilustra en la **Figura 29**: 17 correos fueron

detectados con phishing, mientras que el valor real de correos infectados fue de 20 según lo previamente identificado. Los correos legítimos fueron identificados 83. Sin embrago, la cantidad real de correos sin phishing fue de 80. Esto representa que la detección de correos con phishing posee un error equivalente al 3%, en los 100 correos utilizados.

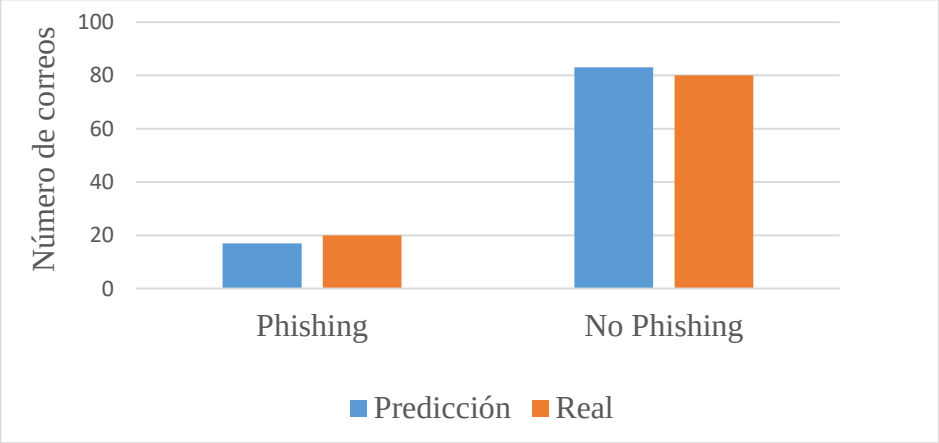


Figura 29. Número de correos con phishing en la etapa de entrenamiento

5.1.2. Etapa de detección del modelo

En esta etapa se emplearon 1325 correos, que fueron sometidos al proceso de detección con el algoritmo de Clasificación Árboles de Decisión, los resultados se muestran en la **Figura 30**. Como se puede observar 530 correos están infectados con phishing, 795 no se encuentran infectados, 17 correos fueron identificados como Falsos Positivos que corresponden a los mensajes que no son phishing sin embargo el modelo los detectó como infectados. Por último, los correos que pertenecen a los Falsos Negativos fueron 27, que constituyen los correos con phishing pero fueron considerados como legítimos. Por tanto, el modelo logra una precisión del 96.77% en la detección de ataques de Ingeniería Social.

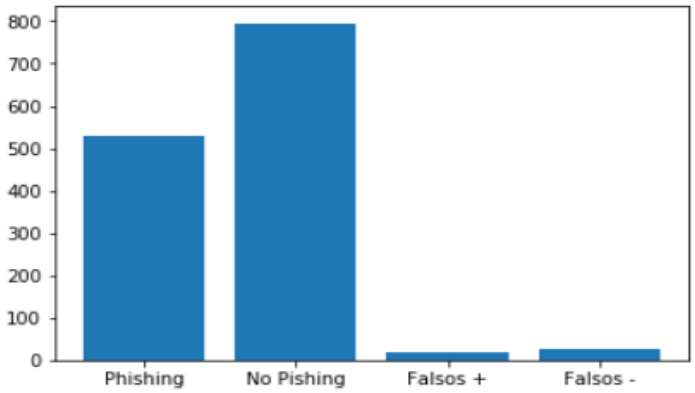


Figura 30. Resultados de la etapa de detección

5.2. Evaluación del desempeño del modelo

En este apartado se describe el proceso estadístico y el análisis e interpretación de los resultados de la etapa de detección de correos con phishing. Para ello, se emplearon algunas ecuaciones de acuerdo con (Kohavi & Provost, 1998) con el propósito conocer el grado de desempeño del algoritmo Árboles de Decisión, y la precisión del modelo en la clasificación del ataque de suplantación de identidad. La representación de la matriz de confusión se muestra en la **Tabla 12**, que interpreta el número de predicciones por categoría y filas en instancias reales.

Tabla 12.
Matriz de confusión

		Predicción	
		Positivo	Negativo
Real	Verdadero	Verdaderos Positivos (TP)	Falsos Negativos (FN)
	Falso	Falsos Positivos (FP)	Verdaderos Negativos (TN)

En la **Tabla 13** se detallan los valores que corresponden a cada variable de la matriz de confusión tanto para TP, TN, FP y FN. Cabe recalcar que los valores se obtuvieron durante la ejecución del modelo en la etapa de detección de phishing con los 1325 correos. La información sobre la **Tabla 12** se especifica a continuación:

Tabla 13.
Valores para las ecuaciones de predicción

Número de predicciones	**Valor**
TN	795
FP	17
FN	27
TP	530

La **Ecuación 4** representa la Tasa de Verdaderos Positivos, True Positive Rate (TPR) o sensibilidad, que proporciona los casos positivos (Kohavi & Provost, 1998). Es decir, los correos que se encuentran infectados con phishing y fueron identificados correctamente por el algoritmo de detección.

$$TPR = \frac{TP}{FN + TP} \qquad \textbf{Ecuación 4}$$

$$TPR = \frac{530}{27 + 530} = 0.9515 * 100\% = 95.15\%$$

La Tasa de Falsos Negativos o False Negative Rate (FNR) definida por la **Ecuación 5** proporciona los casos positivos o correos infectado que fueron detectados de manera incorrecta por el clasificador (Kohavi & Provost, 1998).

$$FNR = \frac{FN}{FN + TP}$$ **Ecuación 5**

$$FNR = \frac{26}{26 + 530} = 0.0467 * 100\% = 4.67\%$$

La **Ecuación 6** constituye la Tasa de Negativos Verdaderos, True Negative Rate (TNR) o especificidad (Kohavi & Provost, 1998). Hace referencia a los casos negativos que el algoritmo los clasificó como correctos, en otras palabras, los correos que fueron detectados como phishing pero no poseen el ataque de Ingeniería Social.

$$TNR = \frac{TN}{TN + FP}$$ **Ecuación 6**

$$TNR = \frac{795}{795 + 26} = 0.9683 * 100\% = 96.83\%$$

Por el contrario, la Tasa de Falsos Positivos o False Positive Rate (FPR) corresponde a la **Ecuación 7**, donde se obtienen los casos negativos o erróneos que fueron clasificados como positivos por el algoritmo (Kohavi & Provost, 1998), es decir, los correos legítimos que se los catalogó como phishing.

$$FPR = \frac{FP}{TN + FP}$$ **Ecuación 7**

$$FPR = \frac{17}{795 + 17} = 0.0209 * 100\% = 2.09\%$$

Las ecuaciones mencionadas anteriormente corresponden a la matriz de confusión sin embargo para establecer una mejor comprensión se describen las siguientes ecuaciones para

conocer la exactitud de predicción, la tasa de errores y la precisión del valor predictivo (Kohavi & Provost, 1998). La exactitud (AC) se representa en la **Ecuación 8**, la misma que proporciona los casos que el modelo los clasificó de manera correcta.

$$AC = \frac{TN + TP}{TN + FN + FP + TP} \qquad \textbf{Ecuación 8}$$

$$AC = \frac{795 + 530}{795 + 27 + 17 + 530} = 0.9678 * 100\% = 96.78\%$$

La tasa de errores (ER) que proporciona la **Ecuación 9** son los casos que no fueron clasificados de manera correcta (Kohavi & Provost, 1998).

$$ER = \frac{FN + FP}{TN + FN + FP + TP} \qquad \textbf{Ecuación 9}$$

$$ER = \frac{27 + 17}{795 + 27 + 17 + 530} = 0.0321 * 100\% = 3.21\%$$

Por último, la **Ecuación 10** muestra el valor predictivo positivo o precisión (P) que representa los casos positivos predichos correctamente (Kohavi & Provost, 1998).

$$P = \frac{TP}{FP + TP} \qquad \textbf{Ecuación 10}$$

$$P = \frac{530}{17 + 530} = 0.9689 * 100\% = 96.89\%$$

Los resultados obtenidos al aplicar las ecuaciones anteriores se describen en la **Tabla 14**. De manera que el análisis de los resultados se realizó en base a los 1325 correos empleados, 795

correos legítimos y 530 correos infectados en la detección de phishing al utilizar Naive Bayes y Árboles de Decisión. Esto muestra que el 95.15% equivalente a 504 correos fueron detectados como infectados de manera correcta (TPR), mientras que el 2.09% (FPR) aproximadamente 11 correos fueron detectados incorrectamente como correos legítimos. Cabe mencionar que al calcular el número de correos acorde al porcentaje equivalente se origina una pérdida debido a los decimales, por esa razón se menciona un valor aproximado que presente la proyección de los resultados obtenidos. En relación a los correos sin phishing la detección fue del 96.83% (TNR) lo que equivale a 769. Sin embargo, la detección no acertó el 4.67% (FNR). En términos generales el modelo presenta una precisión en la predicción del 96.78%, con una tasa de error del 3.21%, lo que refleja que el porcentaje real para la predicción en la detección es del 96.89%.

Tabla 14.
Resultados al emplear la matriz de confusión

Tasa de Verdaderos Positivos (TPR)	95.15%
Tasa de Falsos Positivos (FPR)	2.09%
Tasa de Negativos Verdaderos (TNR)	96.83%
Tasa de Falsos Negativos (FNR)	4.67%
Precisión	96.78%
Tasa de error	3.21%
Valor predictivo	96.89%

5.3. Validación con otros modelos de clasificación de Machine Learning

La validación del modelo de detección y mitigación se realizó mediante la comparación con otros clasificadores de Machine Learning. Se consideró a Random Forest, ya que fue uno de los clasificadores con mayor aceptación en el desarrollo de la revisión sistemática de literatura (ver apartado **3.2.2)**. Además, se consideraron otros clasificadores que se encuentran dentro del grupo

de aprendizaje supervisado de ML así como: Regresión logística, que consiste en conseguir una función de variables independientes para clasificar. Para este caso separar los correos en uno de los dos grupos (Phishing o no Phishing) determinados por los valores de las variables dependientes o las 18 características (Fiuza Pérez & Rodríguez Pérez, 2000) y el clasificador Ficticio que se fundamenta en clasificar aleatoriamente utilizando una distribución uniforme (Pedregosa et al., 2011).

La validación inicia con el entrenamiento de los tres algoritmos mencionados anteriormente. Estos fueron comparados con el algoritmo de Árboles de Decisión del modelo desarrollado. Para ello se empleó la matriz inicial de 100 correos de aprendizaje. La detección se realizó al usar los algoritmos correspondientes a cada clasificador, que se define según el pseudocódigo desde la **Tabla 15** hasta la **Tabla 17**.

Tabla 15.
Algoritmo clasificador de Regresión Logística

```
1 Algoritmo Regresion_Logica
2   Leer x_entrenamiento
3   Leer y_entrenamiento
4   modelo<-lr_ajuste(x_entrenamiento,y_entrenamiento)
5   y_prediccion<-lr_predicion(x_entrenamiento)
6   respuesta<-pd_marco_datos(y_entrenamiento,y_prediccion)
7   porcentaje<-puntuacion_precisión ( y_entrenamiento,y_prediccion)
8   Escribir "Porcentaje de predicción es: " porcentaje
9 FinAlgoritmo
```

Tabla 16.
Algoritmo clasificador Ficticio

```
1 Algoritmo Clasificador_Ficticio
2   Leer x_entrenamiento
3   Leer y_entrenamiento
4   modelo<-dummy_ajuste(x_entrenamiento,y_entrenamiento)
5   y_prediccion<-dummy_prediccion(x_entrenamiento)
6   respuesta<-pd_marco_datos(y_entrenamiento,y_prediccion)
7   porcentaje<-puntuacion_precisión ( y_entrenamiento,y_prediccion)
8   Escribir "Porcentaje de predicción es: " porcentaje
9 FinAlgoritmo
```

Tabla 17.
Algoritmo clasificador Random Forest

```
Algoritmo Clasificador_Arboles_Aleatorios
  Leer x_entrenamiento
  Leer y_entrenamiento
  modelo<-clf_ajuste(x_entrenamiento,y_entrenamiento)
  y_prediccion<-clf_prediccion(x_entrenamiento)
  respuesta<-pd_marco_datos(y_entrenamiento,y_prediccion)
  porcentaje<-puntuacion_precisión ( y_entrenamiento,y_prediccion)
  Escribir "Porcentaje de predicción es: " porcentaje
FinAlgoritmo
```

Como se mencionó anteriormente al considerar la aplicación de los tres clasificadores de ML en la validación del modelo de detección y mitigación de ataques de Ingeniería Social, se logró constatar el acercamiento positivo en la decisión de ataques con la utilización de Naive Bayes y Árboles de Decisión. Para ello, la **Tabla 18** refleja los resultados sobre el porcentaje de detección. El clasificador con menor porcentaje fue el algoritmo Ficticio con un 61.29%, seguido del algoritmo de Regresión Logística con un 90.32% y el por último el algoritmo que tiende acercarse en la predicción al modelo es Random Forest con el 93.54%.

Tabla 18.
Resultado de los clasificadores de ML

Método	**Porcentaje de predicción**
Árboles de Decisión	96.77%
Regresión Logística	90.32%
Ficticio	61.29%
Random Forest	93.54%

La **Figura 31** representa de manera gráfica el resultado de comparar el algoritmo de Árboles de Decisión con los clasificadores de Regresión Logística, Clasificador Ficticio y Random Forest. El resultado que se obtuvo fue el 61.29% con el Clasificador Ficticio, este resultado representa el porcentaje de detección con menor acercamiento en la predicción de correos infectados con

phishing, mientras que los algoritmos de Regresión Logística y Random Forest con el 90.32% y 93.54% respectivamente, representan un acercamiento al resultado obtenido con el modelo propuesto. Sin embargo, como se puede visualizar el prototipo desarrollado posee el mejor porcentaje en la detección de ataques de suplantación de identidad con el 96.77% al emplear dos algoritmos durante el proceso de entrenamiento y detección.

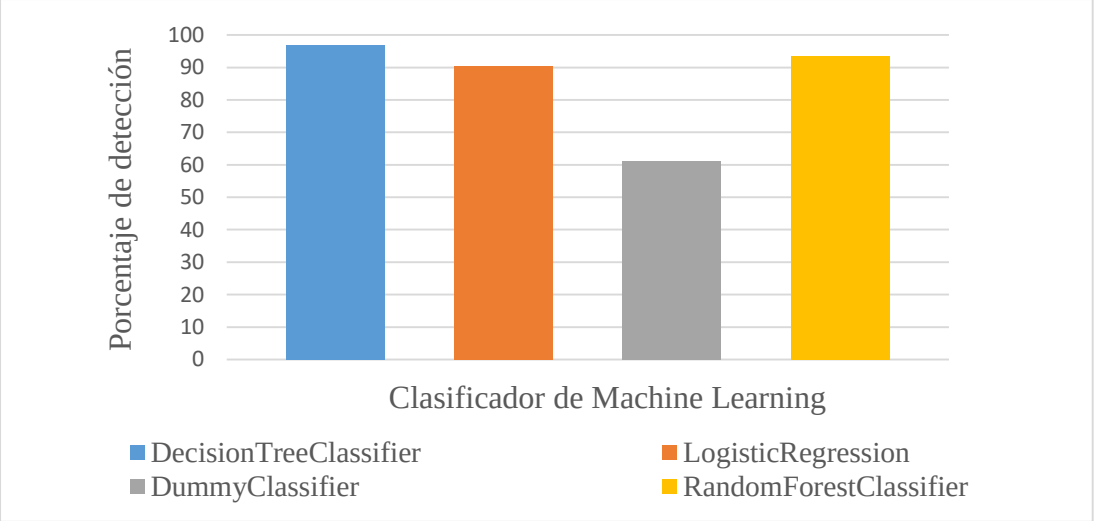


Figura 31. Porcentaje de detección de Clasificadores de ML

5.4. Rendimiento y consumo de recursos

El rendimiento y consumo de recursos fue examinado en dos fases: la primera durante el entrenamiento del modelo y la segunda en la detección del ataque de suplantación de identidad. Para ello se utilizó dos computadoras con las características que se describen en la **Tabla 19**. Es necesario tener en cuenta que poseen recursos, capacidad y año de adquisición diferente, ya que se desea conocer las condiciones en las que el modelo puede desempeñase adecuadamente durante el aprendizaje y detección.

Tabla 19.
Características de las computadoras

Características	**PC1**	**PC2**
Marca	DELL	Toshiba
Modelo	Inspiron 7548	Satellite S855-S5381
Procesador	Intel CORE i7	Intel CORE i7
Generación	5ta generación	3era generación
Memoria Ram	16 GB	8 GB
Disco Duro	1T	1T
Tarjeta gráfica	AMD Radeon R7 M270	Intel HD Graphics 4000
Sistema Operativo	Windows 10	Windows 10
Tipo de sistema	64 bits	64 bits

Se empleó la herramienta del Administrador de tareas nativa de Windows en condiciones estables, es decir, el PC1 con el 2% en CPU, 26% de memoria y en Disco el 0%. Para el caso del PC2 con el 3% en CPU, el 46% de uso en memoria y el 1% en Disco. Al considerar las condicionales iniciales se muestra en la **Tabla 20** el resultado del consumo de recursos y el rendimiento durante la fase de entrenamiento.

Una breve comparación es el consumo en CPU, el PC1 ocupa un rango del 23 al 65% en un tiempo de 28 segundos, mientras que PC2 empleó un rango de 44 al 100% en CPU por un lapso de tiempo de 48 segundos. El tiempo en memoria se mantuvo constante pero el porcentaje de uso no fue el mismo en las dos computadoras, el PC1 utilizó el 26%, mientras el PC2 empleó el 48%.

Tabla 20.
Consumo de recursos en la fase de entrenamiento del modelo

Porcentaje de consumo	**PC1**	**Tiempo (s)**	**PC2**	**Tiempo (s)**
CPU	23 – 65%	28	44 – 100%	48
Disco	23 – 40%	20.4	40 – 80%	39
Memoria	26%	-	48%	-

Wi-Fi	6 – 40%	13.2	30 – 100%	22

Las condiciones iniciales de las dos computadoras para la fase de detección fueron las siguientes: el CPU del PC1 inició con el 3%, en memoria con el 26% y en Disco con 1%. Para el caso del PC2, el 4% en CPU, el 50% de uso en memoria y el 2% en el consumo del Disco. Se muestra en la **Tabla 21**, el rendimiento durante el proceso de detección de phishing en los 1325 correos. En comparación con la etapa de entrenamiento el consumo de CPU del PC1 es menor en 4 segundos, de manera similar el PC2 reduce el tiempo empleado en el aprendizaje, pero incrementa el uso de recursos en un 12%.

Tabla 21.
Consumo de recursos en la fase de detección del modelo

Porcentaje de consumo	**PC1**	**Tiempo (s)**	**PC2**	**Tiempo (s)**
CPU	7 – 100%	24	56 – 81%	16.15
Disco	60%	8	20%	11
Memoria	28%	-	52%	-
Wi-Fi	37.14%	2.6	51%	5

Por último la **Tabla 20** y **Tabla 21**, muestran que durante el proceso de entrenamiento o aprendizaje del modelo el consumo de los recursos fue mayor tanto en la PC1 como la PC2 durante lapsos de tiempo diferentes, teniendo en cuenta que la PC2 utiliza el doble de los recursos en comparación con la PC1. Mientras que en la etapa de detección el tiempo de consumo de los recursos varía acorde a la cantidad de mensajes que analice el modelo. Para los 1325 correos los recursos fueron mayores para la PC1 y la PC2 en referencia a la etapa de aprendizaje. La comparación realizada refleja que las condiciones óptimas para el entrenamiento y detección del modelo deben ser muy similares al PC1, ya que, al tener los recursos suficientes, permitirá que el prototipo logre un desempeño adecuado en las etapas mencionadas anteriormente.

5.5. Discusión

Los resultados obtenidos en el presente proyecto, muestran que el modelo logró un porcentaje de detección del 96.77% en identificar si un correo es o no phishing. A su vez se encuentra en el rango aceptable de los estudios primarios obtenidos en la revisión sistemática de literatura.

Por lo expuesto anteriormente, el modelo cumple con la hipótesis planteada que hace referencia al incremento en la detección y mitigación de ataques de suplantación de identidad en correos electrónicos. Sin embargo, existe un porcentaje de error del 3.21% que no detecta adecuadamente el ataque y genera desconfianza. Esto se debe principalmente por al número de características que se emplean en el análisis de un correo infectado con phishing y la cantidad de datos que se emplean en la fase de entrenamiento. Para reducir el error, se debería profundizar en el análisis de las características que posee un correo infectado y validar el número de datos necesarios para que el aprendizaje no sea sesgado o presente un nivel bajo de entrenamiento.

En referencia al párrafo anterior la selección de los clasificadores de ML, se realizó en base a la revisión sistemática de literatura, en donde se obtuvieron los cuatro clasificadores de aprendizaje supervisado con mayor aceptación por parte de los investigadores. Sin embargo, se debe tener en cuenta que la tecnología se desarrolla de manera exponencial y ocasiona que se vuelva obsoleto o poco útil en la solución del problema. Por ese motivo, se podría mejorar el modelo al emplear técnicas de ML con aprendizaje no supervisado o Deep Learning. No obstante, eso no implica que el modelo no disminuya el problema sobre el incremento de phishing en los últimos años. Al contrario, logra una detección aceptable del ataque de suplantación de identidad

y puede ser la base para futuros estudios o mejoras en el prototipo de identificación de correos ilegítimos.

En relación a la comparación con otros algoritmos de ML de aprendizaje supervisado como Random Forest, Regresión Logística y Clasificador Ficticio. Es importante mencionar que los porcentajes de detección que se obtuvieron no superan el alcanzado con el modelo propuesto, en donde se empleó Naive Bayes y Árboles de Decisión. Sin embargo, se pueden considerar los dos algoritmos (Random Forest y Regresión Logística) que obtuvieron mayor acercamiento al porcentaje de predicción, para mejorar el modelo ya propuesto o estudiarlos con mayor detenimiento en la detección de ataques de phishing.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

La suplantación de identidad es una forma de robo de información confidencial en línea y un medio para realizar fraudes. Los phishers, por ejemplo, usan Ingeniería Social para robar datos de identidad personal de las víctimas. Por esa razón se desarrolló el presente trabajo, con el fin de reducir los ataques de suplantación de identidad mediante la detección y mitigación.

Para lograr el desarrollo se utilizó la Metodología Investigación-Acción, a su vez se emplearon los pasos para la revisión sistemática de literatura, con el propósito de identificar las técnicas de Machine Learning con mayor acogida por parte de los investigadores. Con ello, se obtuvo las características más comunes para el análisis de páginas web, correos electrónicos, análisis de imágenes o archivos adjuntos. Para el desarrollo del proyecto el enfoque principal fue la detección en base a correos electrónicos.

El desarrollo inició con el uso de la Metodología Espiral, la misma que se dividió en ciclos para la definición de cada secuencia del proyecto. Las herramientas que se utilizaron para la ejecución del proyecto fue el framework Jupyter con el lenguaje de programación Python. Se empleó Python debido a que contiene librerías propias de Machine Learning, las cuales permitieron comprender de manera rápida la estructura de un clasificador de aprendizaje supervisado.

El resultado obtenido al utilizar Naive Bayes y Árboles de Decisión fue el 96.77% de precisión en la predicción de ataques de suplantación de identidad. Se tiene en cuenta que se

empleó 1325 correos de los cuales 795 fueron legítimos y 530 correos infectados con phishing. Esto tuvo una tasa de error del 3.21%, con precisión del 96.78% y porcentaje de detección del 96.89% al emplear la matriz de confusión.

El proceso de validación del modelo al emplear Árboles de Decisión en la detección de ataques de suplantación de identidad con otros clasificadores de ML como Regresión Logística, el Clasificador Ficticio y Random Forest fue exitoso. Se realizó mediante una comparación durante el proceso de entrenamiento con todos los algoritmos mencionados. Esto dio como resultado el 61.29% al emplear el Clasificador Ficticio, el 90.32% y el 93.54% para Regresión Logística y Random Foreste respectivamente.

6.2. Recomendaciones

Debido al origen de nuevos ataques con mayor complejidad, se recomienda realizar un estudio que profundice el análisis de las características que presentan los correos electrónicos con phishing, ya que la información que se recapituló fue una breve exposición de los atributos identificados por parte de los investigadores. No obstante, cabe destacar que este proceso es importante, ya que, el modelo logra su aprendizaje a partir de datos previamente definidos.

Durante el desarrollo del presente proyecto se extrajeron enlaces infectados de PhishTank. Una fuente que contiene una base de registros sumamente grande que fueron denunciados fraudulentos. Sin embargo, es aconsejable tomar una muestra significativa de varias fuentes como servidores de correos empresariales o institucionales, Online Link Scan, UrlVoid, etc.

Para proyectos futuros se recomienda, la utilización de dos o más algoritmos de aprendizaje supervisado. Los sugeridos podrían ser Random Forest y Regresión Logística, ya que fueron los

métodos que obtuvieron un porcentaje cercano al modelo de detección desarrollado en el presente estudio. Otro algoritmo podría ser Support Vector Machine, debido a que tuvo una buena aceptación por parte de los investigadores del grupo de estudios primarios. Además, se podría realizar el desarrollo de un Plugin en base al modelo desarrollado que permita la autenticación acorde a un perfil corporativo o personal, de manera que sea compatible en cualquier navegador de Internet.

REFERENCIAS BIBLIOGRÁFICAS

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A Comparison of Machine Learning Techniques for Phishing Detection. En Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit (pp. 60–69). New York, NY, USA: ACM. https://doi.org/10.1145/1299015.1299021.

Aburrous, M., Hossain, M. A., Dahal, K., & Thabtah, F. (2010). Predicting Phishing Websites Using Classification Mining Techniques with Experimental Case Studies. En Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations (pp. 176–181). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ITNG.2010.117

Al-Janabi, M., Quincey, E. de, & Andras, P. (2017). Using Supervised Machine Learning Algorithms to Detect Suspicious URLs in Online Social Networks. En Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017 (pp. 1104–1111). New York, NY, USA: ACM. https://doi.org/10.1145/3110025.3116201.

Anandita, Yadav, D. P., Paliwal, P., Kumar, D., & Tripathi, R. (2017). A Novel Ensemble Based Identification of Phishing E-Mails. En Proceedings of the 9th International Conference on Machine Learning and Computing (pp. 447–451). New York, NY, USA: ACM. https://doi.org/10.1145/3055635.3056583

APWG. (2016). Phishing Activity Trends Report. Recuperado 27 de mayo de 2018, a partir de http://docs.apwg.org/reports/apwg_trends_report_q3_2016.pdf

Awad, M., & Khanna, R. (2015). Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers (1st 2015). Apress OPEN.

Awan, M. S. K., & Dahabiyeh, L. (2018). Corporate attractiveness index: A measure for assessing the potential of a cyber-attack. 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 1-6). https://doi.org/10.1109/IACS.2018.8355432

Basnet, R. B., & Sung, A. H. (2012). Mining Web to Detect Phishing URLs. En Proceedings of the 2012 11th International Conference on Machine Learning and Applications - Volume 01 (pp. 568–573). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ICMLA.2012.104

Basnet, R. B., Sung, A. H., & Liu, Q. (2012). Feature Selection for Improved Phishing Detection. En H. Jiang, W. Ding, M. Ali, & X. Wu (Eds.), Advanced Research in Applied Artificial Intelligence (pp. 252-261). Springer Berlin Heidelberg.

Belabed, A., Aïmeur, E., & Chikh, A. (2012). A Personalized Whitelist Approach for Phishing Webpage Detection. En Proceedings of the 2012 Seventh International Conference on Availability, Reliability and Security (pp. 249–254). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/ARES.2012.54.

Belani, R., & Higbee, A. (2018). The State of Phishing Defense 2018 (Cybersecurity). Recuperado de https://cofense.com/whitepaper/state-of-phishing-defense-2018/

Boehm, B. W. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72. https://doi.org/10.1109/2.59

Borghello, C. (2009). El arma infalible: la Ingeniería Social [Seguridad Informática]. Recuperado 18 de noviembre de 2018, de https://docplayer.es/8473894-El-arma-infalible-la-ingenieria-social.html

Budnik, C. (2012). Software Testing, Software Quality and Trust in Software-Based Systems (pp. 253-253). IEEE. https://doi.org/10.1109/COMPSAC.2012.107

Bullée, J.-W. H., Montoya, L., Pieters, W., Junger, M., & Hartel, P. (2017). On the anatomy of social engineering attacks—A literature-based dissection of successful attacks. Journal of Investigative Psychology and Offender Profiling, 15(1), 20-45. https://doi.org/10.1002/jip.1482

CERT-UK. (2015). Introduction to Social Engineering, 10.

Cestnik, B., Kononenko, I., & Bratko, I. (1987). ASSISTANT 86: A Knowledge-elicitation Tool for Sophisticated Users. Proceedings of the 2Nd European Conference on European Working Session on Learning, 31–45. Recuperado de http://dl.acm.org/citation.cfm?id=3108739.3108742

Chandra, B., Gupta, M., & Gupta, M. P. (2007). Robust Approach for Estimating Probabilities in Naive-Bayes Classifier. En A. Ghosh, R. K. De, & S. K. Pal (Eds.), Pattern Recognition and Machine Intelligence (pp. 11-16). Springer Berlin Heidelberg.

Chang, M. (2017, noviembre 30). 4 Stages of the Machine Learning (ML) Modeling Cycle | LinkedIn. Recuperado 13 de abril de 2019, de Linkedin website: https://www.linkedin.com/pulse/4-stages-machine-learning-ml-modeling-cycle-maurice-chang/

Chen, T.-C., Stepan, T., Dick, S., & Miller, J. (2014). An Anti-Phishing System Employing Diffused Information. ACM Trans. Inf. Syst. Secur., 16(4), 16:1–16:31. https://doi.org/10.1145/2584680.

Chowdhury, M., Rahman, A., & Islam, R. (2017). Protecting data from malware threats using aprendizaje automático technique. En 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 1691-1694). https://doi.org/10.1109/ICIEA.2017.8283111

Chung, H., Chen, R., Han, S. C., & Kang, B. H. (2016). Combining RDR-Based Machine Learning Approach and Human Expert Knowledge for Phishing Prediction. En R. Booth & M.-L. Zhang (Eds.), PRICAI 2016: Trends in Artificial Intelligence (pp. 80-92). Springer International Publishing.

Cloudflare. (2019). What Is a Distributed Denial-of-Service (DDoS) Attack? Recuperado 29 de marzo de 2019, de Cloudflare website: https://www.cloudflare.com/en-au/learning/ddos/what-is-a-ddos-attack/

CONGRESO NACIONAL. (2002). LEY DE COMERCIO ELECTRÓNICO, FIRMAS ELECTRÓNICAS Y MENSAJES DE DATOS (Ley No. 2002-67). Recuperado a partir de http://www.oas.org/juridico/spanish/cyb_ecu_ley_comelectronico.pdf

Delers, A. (2016). El principio de Pareto: Optimice su negocio con la regla del 80/20. Recuperado de https://books.google.com.ec/books?id=3WDyCwAAQBAJ&printsec=frontcover&hl=es& source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Digiware. (2016). Contexto de seguridad en Tecnología Operacional. Recuperado 27 de mayo de 2018, a partir de http://www.digiware.net/sites/default/files/doc_digiware/ Digiware-OTOscarCifuentes.pdf

Domingos, P. M., & Pazzani, M. J. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, 105–112. Recuperado de http://dl.acm.org/citation.cfm?id=3091696.3091710

Douzi, S., Amar, M., & Ouahidi, B. E. (2017). Advanced Phishing Filter Using Autoencoder and Denoising Autoencoder. En BDIOT. https://doi.org/10.1145/3175684.3175690.

Fiuza Pérez, Ma. D., & Rodríguez Pérez, J. C. (2000, diciembre). La regresión logística: una herramienta versátil. Nefrología, 20(6), 477-565.

Gómez Vieites, Á. (2014). Enciclopedia de la Seguridad Informática (2da.). México: Alfaomega Grupo Editor S.A.

Gowtham, R., & Krishnamurthi, I. (2014). A comprehensive and efficacious architecture for detecting phishing webpages. Computers & Security, 40, 23-37. https://doi.org/10.1016/j.cose.2013.10.004.

Gyawali, B., Solorio, T., Montes-y-Gómez, M., Wardman, B., & Warner, G. (2011). Evaluating a Semisupervised Approach to Phishing Url Identification in a Realistic Scenario. En Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference (pp. 176–183). New York, NY, USA: ACM. https://doi.org/10.1145/2030376.2030397.

Hadnagy, C. (2014, abril 28). The Social Engineering Infographic. Recuperado 13 de febrero de 2019, de https://www.social-engineer.org/social-engineering/social-engineering-infographic/

Hamid, I. R. A., Abawajy, J., & Kim, T. (2013). Using Feature Selection and Classification Scheme for Automating Phishing Email Detection. Studies in Informatics and Control, 22(1). https://doi.org/10.24846/v22i2y101307.

Huang, H., Wang, Y., & Liu, L. (2011). Detection technology of phishing. En 2011 International Conference on Electrical and Control Engineering (pp. 3890-3893). https://doi.org/10.1109/ICECENG.2011.6057587

Huber, M., Kowalski, S., Nohlberg, M., & Tjoa, S. (2009). Towards automating social engineering using social networking sites. Computational Science and Engineering, IEEE International Conference on.

Islam, R., & Abawajy, J. (2013). A Multi-tier Phishing Detection and Filtering Approach. J. Netw. Comput. Appl., 36(1), 324–335. https://doi.org/10.1016/j.jnca.2012.05.009

Ismail, I., Marsono, M. N., & Nor, S. M. (2014). Malware detection using augmented naive Bayes with domain knowledge and under presence of class noise. International Journal of Information and Computer Security, 6(2), 179-197. https://doi.org/10.1504/IJICS.2014.065173.

ISOTools. (2017, marzo 24). Aspectos clave de la Seguridad de la Información según ISO 27001. Recuperado de https://www.isotools.cl/aspectos-clave-de-la-seguridad-de-la-informacion-segun-iso-27001/

ITU. (2007). Guía de ciberseguridad para los países en desarrollo. Recuperado 4 de marzo de 2019, de https://www.itu.int:443/es/publications/ITU-D/Pages/publications.aspx

ITU. (2016, agosto 22). Cyberbullying: How ITU is countering the global trend. Recuperado 31 de marzo de 2019, de ITU News website: https://news.itu.int/cyberbullying-how-itu-is-countering-the-global-trend/

Jain, A. K., & Gupta, B. B. (2018). Towards Detection of Phishing Websites on Client-side Using Machine Learning Based Approach. Telecommun. Syst., 68(4), 687–700. https://doi.org/10.1007/s11235-017-0414-0

Jypiter. (2019, mayo 6). How IPython and Jupyter Notebook work - Jupyter Documentation 4.1.1 alpha documentation. Recuperado 15 de mayo de 2019, de Jypiter Documentation website: https://jupyter.readthedocs.io/en/latest/architecture/how_jupyter_ipython_work.html

Kaspersky Lab. (2018, enero 9). MailSploit: se descubren vulnerabilidades que permiten falsificar los encabezados de correos electrónicos [Noticias Informáticas]. Recuperado 17 de febrero de 2019, de https://securelist.lat/mailsploit-se-descubren-vulnerabilidades-que-permiten-falsificar-los-encabezados-de-correos-electronicos-2/85898/

Ke, L., Li, B., & Vorobeychik, Y. (2016). Behavioral Experiments in Email Filter Evasion. En Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 827–833). Phoenix, Arizona: AAAI Press. Recuperado de http://dl.acm.org/citation.cfm?id=3015812.3015935

Kitchenham, B., & Charters, S. (2007, julio 9). Guidelines for performing Systematic Literature Reviews in Software Engineering. Recuperado de https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Kohavi, R., & Provost, F. (1998). On Applied Research in Machine Learning (Vol. 30). New York: Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process.

L'Huillier, G., Weber, R., & Figueroa, N. (2010). Online Phishing Classification Using Adversarial Data Mining and Signaling Games. SIGKDD Explor. Newsl., 11(2), 92–99. https://doi.org/10.1145/1809400.1809421.

Labaca Castro, R. (2011, julio 21). Envenenamiento de DNS: ataque de phishing a banco en Brasil. Recuperado 17 de febrero de 2019, de https://www.welivesecurity.com/la-es/2011/07/21/envenenamiento-dns-phishing-banco-brasil/

Liu, Y., Zhang, J., Sarabi, A., Liu, M., Karir, M., & Bailey, M. (2015). Predicting Cyber Security Incidents Using Feature-Based Characterization of Network-Level Malicious Activities. En Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics (pp. 3–9). New York, NY, USA: ACM. https://doi.org/10.1145/2713579.2713582.

Ma, L., Ofoghi, B., Watters, P., & Brown, S. (2009). Detecting Phishing Emails Using Hybrid Features. En Proceedings of the 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (pp. 493–497). Washington, DC, USA: IEEE Computer Society. https://doi.org/10.1109/UIC-ATC.2009.103

MathWorks. (2019). Machine Learning. Recuperado 7 de abril de 2019, de MathWorks website: https://es.mathworks.com/discovery/machine-learning.html

Minguillón, J., & Pujol, J. (2002). Árboles de decisión. Terceras Jornadas de Matemática Discreta y Algorítmica, 3.

Mohammed, M., Khan, M. B., & Mohammed, E. (2016). Machine Learning: Algorithms and Applications (ed. 2016). CRC Press.

Mourtaji, Y., Bouhorma, M., & Alghazzawi. (2017). Perception of a New Framework for Detecting Phishing Web Pages. En Proceedings of the Mediterranean Symposium on Smart City Application (pp. 11:1–11:6). New York, NY, USA: ACM. https://doi.org/10.1145/3175628.3175633.

Ndibwile, J. D., Kadobayashi, Y., & Fall, D. (2017). UnPhishMe: Phishing Attack Detection by Deceptive Login Simulation through an Android Mobile App. En 2017 12th Asia Joint Conference on Information Security (AsiaJCIS) (pp. 38-47). https://doi.org/10.1109/AsiaJCIS.2017.19

NIST. (2015). malware - Glossary [Seguridad Informática]. Recuperado 29 de marzo de 2019, de Computer Security Resource Center website: https://csrc.nist.gov/glossary/term/malware

OEA, Symantec, & AMERIPOL. (2014). Tendencias de Seguridad Cibernética en América Latina y el Caribe. Recuperado 28 de mayo de 2018, a partir de https://www.symantec.com/content/es/mx/enterprise/other_resources/b-cyber-security-trends-report-lamc.pdf

Onieva, D. (2019, enero 26). Netflix ya es la segunda empresa del mundo con más ataques phishing [Informática y noticias]. Recuperado 9 de febrero de 2019, de https://www.adslzone.net/2019/01/26/netflix-empresa-ataques-phishing/

OpenDNS. (2005). PhishTank | Join the fight against phishing. Recuperado 2 de junio de 2019, de PhishTank website: https://www.phishtank.com/

Osuagwu, E. U., Chukwudebe, G. A., Salihu, T., & Chukwudebe, V. N. (2015). Mitigating social engineering for improved cybersecurity. En 2015 International Conference on Cyberspace (CYBER-Abuja) (pp. 91-100). https://doi.org/10.1109/CYBER-Abuja.2015.7360515

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011, enero 2). Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research, 12, 2825-2830.

Pierdant, M. (2013). Ciberbullying, Grooming and Sexting, los niños y adolescentes ante el internet. ¿Dónde estamos los padres de familia y los pediatras? Recuperado de https://www.medigraphic.com/pdfs/conapeme/pm-2013/pm133a.pdf

Pressman, R. (2010). Ingeniería del software. Un enfoque práctico (7ma ed.). Mexico: McGraw Hill.

Ramos, M. del P., & Hurtado, A. (2011). Seguridad Informática (11.ª ed.). Editorial Paraninfo.

Ramos, P. (2011, noviembre 2). El ciclo de un ataque de malware. Recuperado 5 de marzo de 2019, de https://www.welivesecurity.com/la-es/2011/11/02/el-ciclo-de-un-ataque-de-malware/

Ravula, R., Liszka, K., & Chan, C. (2011). Dynamic Analysis of Malware Using Decision Trees. Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, 74-83. https://doi.org/10.5220/0003660200740083.

Salgado, L. L. F. (2014). Derecho Informático. Grupo Editorial Patria.

Sandoval Castellanos, E. J. (2018). Ingeniería Social: Corrompiendo la mente humana [Seguridad Informática]. Recuperado 31 de marzo de 2019, de Universidad Nacional Autónoma de México website: https://revista.seguridad.unam.mx/numero-10/ingenier%C3%AD-social-corrompiendo-la-mente-humana

Sanglerdsinlapachai, N., & Rungsawang, A. (2010). Web Phishing Detection Using Classifier Ensemble. En Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services (pp. 210–215). New York, NY, USA: ACM. https://doi.org/10.1145/1967486.1967521.

Sanou, B. (2015). ICT Facts and Figures - The world in 2015. Recuperado 18 de marzo de 2018, a partir de https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf

Shukla, N. (2017). Machine Learning with TensorFlow (10.ª ed.). Recuperado de https://livebook.manning.com/#!/book/machine-learning-with-tensorflow/about-this-book/

Smola, A., & Vishwanathan, S. V. N. (2008). Introduction to Machine Learning (ed. 2008). Reino Unido: Cambridge University Press.

Sommerville, I. (2011). Ingeniería del software (9na ed.). Mexico: Pearson.

Toolan, F., & Carthy, J. (2010). Feature selection for Spam and Phishing detection. En 2010 eCrime Researchers Summit (pp. 1-12). https://doi.org/10.1109/ecrime.2010.5706696.

UNESCO. (2019, febrero 28). Workshop on «Machine learning, human development, and the future of Futures Literacy». Recuperado 5 de abril de 2019, de UNESCO website: https://en.unesco.org/events/workshop-machine-learning-human-development-and-future-futures-literacy

UNICEF. (2016). Guía de sensibilización sobre Convivencia Digital (1.ª ed.). Recuperado de https://www.unicef.org/argentina/media/1601/file

United Nations Office on Drugs and Crime (UNODC). (2012). The use of the Internet for terrorist purposes. Recuperado de https://www.unodc.org/documents/frontpage/Use_of_ Internet_for_ Terrorist_Purposes.pdf

Valle, J. C. (2013). EL DELITO INFORMÁTICO DE PHISHING (Maestría). Universidad Regional Autónoma de los Andres (UNIANDES), Quevedo-Ecuador. Recuperado de http://dspace.uniandes.edu.ec/bitstream/123456789/2819/1/TUQMDPC005-2013.pdf

Valle, M. (2018, enero 25). Panda Security advierte de una campaña de phishing que roba cuentas de Spotify. Recuperado 13 de febrero de 2019, de https://bitlifemedia.com/2018/01/alerta-campana-phishing-roba-cuentas-spotify/

Verma, R., & Dyer, K. (2015). On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. En Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (pp. 111–122). New York, NY, USA: ACM. https://doi.org/10.1145/2699026.2699115.

Verma, R., & Rai, N. (2015). Phish-IDetector: Message-ID based automatic phishing detection. En 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE) (Vol. 04, pp. 427-434).

Wu, S., Tong, X., Wang, W., Xin, G., Wang, B., & Zhou, Q. (2018). Website Defacements Detection Based on Support Vector Machine Classification Method. En Proceedings of the 2018 International Conference on Computing and Data Engineering (pp. 62–66). New York, NY, USA: ACM. https://doi.org/10.1145/3219788.3219804.

Wu, T., Liu, S., Zhang, J., & Xiang, Y. (2017). Twitter Spam Detection Based on Deep Learning. En Proceedings of the Australasian Computer Science Week Multiconference (pp. 3:1–3:8). New York, NY, USA: ACM. https://doi.org/10.1145/3014812.3014815.

Yamak, Z., Saunier, J., & Vercouter, L. (2016). Detection of Multiple Identity Manipulation in Collaborative Projects. En Proceedings of the 25th International Conference Companion on World Wide Web (pp. 955–960). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2872518.2890586

Yusoff, M. N., & Jantan, A. (2011). A Framework for Optimizing Malware Classification by Using Genetic Algorithm. En J. M. Zain, W. M. bt Wan Mohd, & E. El-Qawasmeh (Eds.), Software Engineering and Computer Systems (pp. 58-72). Springer Berlin Heidelberg.

Zaforas, M. (2016, septiembre 19). Jupyter: Data Science aplicada. Recuperado 15 de mayo de 2019, de Paradigma website: https://www.paradigmadigital.com/dev/jupyter-data-science-aplicada/

Zhan, J., & Thomas, L. (2011). Phishing detection using stochastic learning-based weak estimators. En 2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS) (pp. 55-59). https://doi.org/10.1109/CICYBS.2011.5949409.

Zhang, H., Liu, G., Chow, T. W. S., & Liu, W. (2011). Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach. IEEE Transactions on Neural Networks, 22(10), 1532-1546. https://doi.org/10.1109/TNN.2011.2161999.

Zhang, W., Jiang, Q., Chen, L., & Li, C. (2017). Two-stage ELM for phishing Web pages detection using hybrid features. World Wide Web, 20(4), 797-813. https://doi.org/10.1007/s11280-016-0418-9.

Meneses, F., Fuertes, W., Sancho, J., Salvador, S., Flores, D., Aules, H., ... & Nuela, D. (2016). RSA encryption algorithm optimization to improve performance and security level of network messages. Int. J. Comput. Sci. Netw. Secur., 16(8), 55-62.

Bustamante, F., Fuertes, W., Tulkeredis, T., & Ron, M. (2018, April). Situational Status of Global Cybersecurity and Cyber Defense According to Global Indicators. Adaptation of a Model for Ecuador. In International Conference of Research Applied to Defense and Security (pp. 12-26). Springer, Cham.

Ron, M., Fuertes, W., Bonilla, M., Toulkeridis, T., & Diaz, J. (2018, June). Cybercrime in Ecuador, an exploration, which allows to define national cybersecurity policies. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-7). IEEE.

Ron, M., Rivera, O., Fuertes, W., Toulkeridis, T., & Díaz, J. (2019, February). Cybersecurity Baseline, An Exploration, Which Permits to Delineate National Cybersecurity Strategy in

Ecuador. In International Conference on Information Technology & Systems (pp. 847-857). Springer, Cham.