



UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

**MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y
EVITAR LA INSERCIÓN DE MALWARE EN UNA RED,
BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO.**

PROYECTO DE TITULACIÓN

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTORES:

Dayannara Cindy Avila Maldonado




Joel Anthony Torres Urresto

TUTOR:

Ph.D. Franklin Ricardo Parrales Bravo

GUAYAQUIL – ECUADOR

2020

  	
REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍAS	
FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN	
TÍTULO: “ <i>Modelo De Detección De Intrusos Para Detectar Y Evitar La Inserción De Malware En Una Red, Basado En Técnicas De Aprendizaje Automático</i> ”.	
AUTORES: Dayannara Cindy Avila Maldonado Joel Anthony Torres Urresto	REVISOR: Ing. Julio Barzola Montes, MSc.
INSTITUCIÓN: Universidad de Guayaquil	FACULTAD: Ciencias Matemáticas y Físicas
CARRERA: Ingeniería en Sistemas Computacionales	
FECHA DE PUBLICACIÓN:	N° DE PAGES: 171
AREA TEMÁTICA: Seguridad Informática	
PALABRAS CLAVES: aprendizaje automático, ciberataque, conjunto de datos, filter, selección de subconjuntos de características, wrapper.	
<p>RESUMEN: Los ciberataques son uno de los principales problemas que afectan a las empresas a nivel mundial. Los causantes detrás de los ataques son conocidos como ciberdelincuentes. Estos aprovechan vulnerabilidades existentes en los sistemas informáticos para efectuar el ataque, ocasionando robo de información confidencial y pérdidas económicas para las empresas u organizaciones afectadas. Es por ello que se busca una alternativa para disminuir este problema, una opción a considerar es el aprendizaje automático como herramienta para mejorar la seguridad informática. El presente trabajo de titulación tuvo como finalidad presentar un modelo de detección de intrusos que hace uso de la técnica propuesta en el presente trabajo de titulación, la cual combina las técnicas filter y wrapper para la selección de características en la fase de preprocesamiento de datos. El conjunto de datos utilizado para el entrenamiento y prueba de los modelos fue obtenido del repositorio GitHub¹. Se utilizaron algoritmos de clasificación para el entrenamiento de los modelos. En base a la métrica de exactitud se seleccionó al mejor modelo de detección de intrusos, el cual fue entrenado mediante el algoritmo RandomForest. Este modelo consiguió una media del 99,42% de exactitud con la técnica de selección de características propuesta, mejorando en un 0.10% al resultado del modelo entrenado con el mismo algoritmo, pero sin el uso de la metodología propuesta. Con ello se evidencia que los modelos entrenados con la metodología propuesta proporcionan rendimientos similares a los modelos que no hacen uso de la misma,</p>	

¹ Disponible en: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

<i>contando con la ventaja de eliminar aquellas características redundantes del conjunto de datos. Cabe mencionar que, el tiempo de entrenamiento de los modelos con cada uno de los algoritmos para poder evaluar su desempeño y seleccionar al mejor fue de aproximadamente un minuto con diez segundos.</i>		
N° DE REGISTRO:	N° DE CLASIFICACIÓN:	
DIRECCIÓN URL: (PROYECTO DE TITULACION EN LA WEB)		
ADJUNTO PDF	SI <input checked="checked" type="checkbox"/>	NO <input type="checkbox"/>
CONTACTO CON AUTORES: Dayannara Cindy Avila Maldonado Joel Anthony Torres Urresto	Teléfono: 0968949347 0994494027	Email: dayannara.avilam@ug.edu.ec joel.torresu@ug.edu.ec
CONTACTO DE LA INSTITUCIÓN	Nombre: Ab. Juan Chávez Atocha	
	Teléfono: 2307729	
	Email: juan.chaveza@ug.edu.ec	

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación, “modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático.” elaborado por la Srta. Avila Maldonado Dayannara Cindy y el Sr. Torres Urresto Joel Anthony, **estudiantes no titulados** de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero(a) en Sistemas Computacionales, me permito declarar que luego de haber orientado, estudiado y revisado, la **apruebo** en todas sus partes.

Atentamente,

Ph.D. Franklin Ricardo Parrales Bravo

TUTOR

DEDICATORIA

A mis padres, Oscar Uribe Avila Zambrano y a Alba Elena Maldonado Solano. Gracias a ellos y a su apoyo fundamental en mi formación académica he podido cumplir con este objetivo. Sé que mi padre desde el cielo está orgulloso de mí.

A cada uno de mis familiares que me apoyaron en mi etapa universitaria.

Dayannara Cindy Avila Maldonado

Este trabajo va dedicado a mi padre Aníbal Torres y madre Aydee Urresto, quienes me inculcaron valores, me enseñaron a ser una persona correcta y que sin esfuerzo no se consigue nada en la vida.

A mis abuelitas María Inés Vargas y Pompella Espinoza, quienes intercedieron por mí, orando cada mañana y cada noche porque las cosas me vayan de la mejor manera. A mis tías, tíos, primos, primas y amigos, por cada momento que vivimos y que nos llenó de alegría y reflexión.

Joel Anthony Torres Urresto

AGRADECIMIENTO

Agradezco a Dios por estar siempre conmigo.

Agradezco a la Universidad de Guayaquil, y a los docentes que estuvieron presentes en mi proceso académico. A mis amigos y compañeros con los que compartí esta etapa universitaria.

Dayannara Cindy Avila Maldonado

Agradezco a Dios, a mis padres, abuelitas, tías, tíos, primas, primos y amigos por su apoyo incondicional que me brindaron durante todo este proceso académico.

Joel Anthony Torres Urresto

TRIBUNAL PROYECTO DE TITULACIÓN

Ing. José González Ruiz, M.Sc.
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y FÍSICAS

Ing. Lorenzo Cevallos Torres, M.Sc.
DIRECTOR DE LA CARRERA DE
INGENIERÍA EN SISTEMAS
COMPUTACIONALES

Ph.D. Franklin Ricardo Parrales Bravo
PROFESOR TUTOR DEL PROYECTO
DE TITULACIÓN

Ing. Julio Barzola Montes, M.Sc.
PROFESOR REVISOR DEL PROYECTO
DE TITULACIÓN

Ab. Juan Chávez Atocha, Esp.
SECRETARIO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Proyecto de Titulación, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL”.

Dayannara Cindy Avila Maldonado

Joel Anthony Torres Urresto



CESIÓN DE DERECHOS DE AUTOR

Ingeniero

Ing. José González Ruiz, M.Sc.

DECANO DE LA FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

Presente.

A través de este medio indico a usted que procedo a realizar la entrega de la cesión de derechos de autor en forma libre y voluntaria del trabajo de titulación “**Modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático.**”, realizado como requisito previo para la obtención del Título de Ingeniero en Sistemas Computacionales de la Universidad de Guayaquil.

Guayaquil, marzo de 2021.

Dayannara Cindy Avila Maldonado
C.I. N° 0929308898

Joel Anthony Torres Urresto
C.I. N° 0958897795



UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN
DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE
AUTOMÁTICO.**

Proyecto de Titulación que se presenta como requisito para optar por el título de
INGENIERO(A) EN SISTEMAS COMPUTACIONALES

Autores:

Dayannara Cindy Avila Maldonado

C.I. N° 0929308898

Joel Anthony Torres Urresto

C.I. N° 0958897795

Tutor: Ph.D. Franklin Ricardo Parrales Bravo

Guayaquil, marzo de 2021

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto de Titulación, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

CERTIFICO:

Que he analizado el Proyecto de Titulación presentado por los estudiantes **Dayannara Cindy Avila Maldonado, Joel Anthony Torres Urresto**, como requisito previo para optar por el Título de Ingeniero(a) en Sistemas Computacionales cuyo proyecto es:

MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO.

Considero aprobado el trabajo en su totalidad.

Presentado por:

Avila Maldonado Dayannara Cindy

0929308898

Cédula de identidad N°

0958897795

Torres Urresto Joel Anthony

Cédula de identidad N°

Tutor(a): _____

Firma

Guayaquil, marzo de 2021



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL

1. Identificación del Proyecto de Titulación

Nombre del Estudiante: Avila Maldonado Dayannara Cindy	
Dirección: Isla Trinitaria Coop. 4 de marzo Mz. 432 Sl. 8	
Teléfono: 0968949347	Email: dayannara.avilam@ug.edu.ec

Nombre del Estudiante: Torres Urresto Joel Anthony	
Dirección: Guasmo Central Coop. Carlos Castro 2 Mz. 31 Sl. 7	
Teléfono: 0994494027	Email: joel.torresu@ug.edu.ec

Facultad: Ciencias Matemáticas Y Físicas
Carrera: Ingeniería En Sistemas Computacionales
Proyecto de Titulación al que opta:
Profesor(a) Tutor(a): Ph.D. Franklin Ricardo Parrales Bravo

Título del Proyecto de Titulación: Modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático.
--

Palabras Claves: aprendizaje automático, ciberataque, conjunto de datos, filter, selección de subconjunto de características, wrapper
--

2. Autorización de Publicación de Versión Electrónica del Proyecto de Titulación

A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de este Proyecto de Titulación.

Publicación Electrónica:

Inmediata		Después de 1 año	
-----------	--	------------------	--

Firma Estudiante:

 Avila Maldonado Dayannara Cindy

 0929308898
 Cédula de identidad N°

 Torres Urresto Anthony Joel

 0958897795
 Cédula de identidad N°

3. Forma de envío:

El texto del Proyecto de Titulación debe ser enviado en formato Word, como archivo .docx, .RTF o .Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM ☐CDROM ☐

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	3
DEDICATORIA.....	4
AGRADECIMIENTO	5
TRIBUNAL PROYECTO DE TITULACIÓN	6
DECLARACIÓN EXPRESA.....	7
CESIÓN DE DERECHOS DE AUTOR	8
CERTIFICADO DE ACEPTACIÓN DEL TUTOR	10
AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL	11
ÍNDICE GENERAL	13
ÍNDICE DE TABLAS.....	18
ÍNDICE DE FIGURAS.....	20
ABREVIATURAS.....	21
RESUMEN.....	22
ABSTRACT.....	24
INTRODUCCIÓN	26
CAPÍTULO I.....	28
PLANTEAMIENTO DEL PROBLEMA	28
Descripción de la situación problemática	28

Ubicación del problema en un contexto.....	28
Situación conflicto nudos críticos	32
Delimitación del problema.....	33
Evaluación del Problema	34
Causas y consecuencias del problema	36
Formulación del problema	37
Objetivos del proyecto	38
Objetivo general.....	38
Objetivos específicos	38
Alcance del proyecto	38
Justificación e importancia	39
Limitaciones del estudio	40
CAPÍTULO II	41
MARCO TEÓRICO	41
Antecedentes del estudio.....	41
Fundamentación teórica	50
Revisiones sistemáticas	81
Meta-análisis	82
Hipótesis / Preguntas científicas a contestarse	91
Variables de la investigación.....	91

Definiciones conceptuales	91
CAPÍTULO III.....	95
METODOLOGÍA DE LA INVESTIGACIÓN	95
Tipo de investigación	95
Diseño metodológico de la investigación	96
Entregables del proyecto	98
Análisis de factibilidad	99
Propuesta	103
Lectura y limpieza de datos:	104
Selección de características:	106
División del conjunto de datos:.....	112
Entrenamiento del modelo	113
Evaluación del modelo.....	115
Validación del modelo	115
Resultados	117
Criterios de validación de la propuesta	123
Análisis de indicadores	123
CAPÍTULO IV	126
CONCLUSIONES Y RECOMENDACIONES.....	126
Conclusiones	126

Recomendaciones	127
Trabajo futuro.....	128
REFERENCIAS BIBLIOGRÁFICAS.....	129
Referencias.....	129
BIBLIOGRAFÍA.....	141
ANEXOS.....	144
Anexo 1. Planificación de actividades del proyecto	145
Anexo 2. Geo-localización del problema.....	147
Anexo 3. Fundamentación Legal	148
Anexo 4. Criterios éticos a utilizarse en el desarrollo del proyecto.....	153
Anexo 5. Formato para la validación de expertos.....	154
Anexo 6. Formato de constancia de juicio de experto	155
Anexo 7. Validación de expertos.	156
Anexo 8. Acta de entrega y recepción definitiva.....	163
Anexo 9. Artículo científico	164
INTRODUCCION	164
CONTENIDO	164
MATERIALES Y METODOS.....	166
3.1 <i>Tipo de investigación</i>	166
3.2 <i>Diseño metodológico de la investigación</i>	166

3.2.1	<i>Árbol de problemas</i>	166
3.2.2	<i>Metodología de selección de subconjunto de datos</i>	167
RESULTADOS		170
CONCLUSIONES		171
REFERENCIAS		171

ÍNDICE DE TABLAS

Tabla 1	Delimitación del problema.....	34
Tabla 2	Matriz de causas y consecuencias del problema.....	37
Tabla 3	Fortalezas y debilidades del algoritmo árbol de clasificación	66
Tabla 4	Fortalezas y debilidades del algoritmo Gradient boosting.....	68
Tabla 5	Matriz de Confusión	75
Tabla 6	Medidas de Desempeño	77
Tabla 7	Trabajos revisados	82
Tabla 8	Trabajos revisados	83
Tabla 9	Trabajos revisados	84
Tabla 10	Trabajos revisados	85
Tabla 11	Trabajos revisados	86
Tabla 12	Trabajos revisados	87
Tabla 13	Trabajos revisados	88
Tabla 14	Porcentaje de precisión y tiempo de creación de los modelos	89
Tabla 15	Porcentaje de precisión de los algoritmos.....	90
Tabla 16	Características del hardware	100
Tabla 17	Requisitos de software	100
Tabla 18	Resumen costo de inversión.....	102
Tabla 19	Costo por recurso humano	102
Tabla 20	Costo por recurso hardware y software	103
Tabla 21	Características del conjunto de datos.....	103

Tabla 22	Conjunto de datos	105
Tabla 23	Detalle del conjunto de datos	106
Tabla 24	Pseudocódigo para almacenar características	110
Tabla 25	Algoritmos y sus parámetros	113
Tabla 26	Media y desviación estándar de las métricas	118
Tabla 27	Resultados de los algoritmos	119
Tabla 28	Matriz de confusión	120

ÍNDICE DE FIGURAS

Figura 1	Sistema de Detección de Intrusos.....	57
Figura 2	Metodología CRISP-DM.....	59
Figura 3	Método f_classif.....	62
Figura 4	Método Chi-Cuadrado (Chi)	63
Figura 5	Método mutual_info_classif.....	64
Figura 6	Método de selección hacia adelante	65
Figura 7	Árbol de clasificación vs Random Forest.....	67
Figura 8	Clasificadores débiles y sus pesos aumentados.....	69
Figura 9	Clasificador final	70
Figura 10	Cross Fold Validation.....	78
Figura 11	Las 4 Etapas del Ciclo de Modelado del Aprendizaje Automático.....	78
Figura 12	Flujo de Aprendizaje	79
Figura 13	Fase de evaluación del modelo.....	80
Figura 14	Árbol de Problemas	96
Figura 15	Árbol de Objetivos	97
Figura 16	Estructura analítica del proyecto	98
Figura 17	Fases de la metodología propuesta para detección de intrusos	104
Figura 18	Fragmento de código que ejecuta la lectura y la limpieza de los datos.....	105
Figura 19	Cálculo de variable k para método chi cuadrado (chi2).....	108
Figura 20	Cálculo de variable k para método f_classif	108
Figura 21	Cálculo de variable k para método mutual_info_classif	108
Figura 22	Selección de características métodos filter.....	109

Figura 23 Metodología propuesta para la selección de subconjunto de características.....	112
Figura 24 División del conjunto de datos de entrenamiento y prueba.....	112
Figura 25 Comparación de modelos clasificadores	115
Figura 26 Pasos de la validación del modelo.....	116
Figura 27 Creación de archivos maliciosos	116
Figura 28 Prueba de predicción del modelo	121
Figura 29 Análisis y confirmación de archivos maliciosos	122

ABREVIATURAS

BBC	Corporación Británica de Radiodifusión
ESET	Compañía De Seguridad Informática
FN	Falsos negativos
FP	Falsos positivos
IA	Inteligencia Artificial
IDE	Entorno de desarrollo integrado
IDS	Sistema de detección de intrusos
Ing.	Ingeniero
M.Sc.	Máster
PyMes	Pequeñas y medianas empresas
TN	Verdaderos negativos
TP	Verdaderos positivos



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN
DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE
AUTOMÁTICO.**

Autores:

Dayannara Cindy Avila Maldonado
C.I. N° 0929308898
Joel Anthony Torres Urresto
C.I. N° 0958897795

Tutor: Ph.D. Franklin Ricardo Parrales Bravo

RESUMEN

Los ciberataques son uno de los principales problemas que afectan a las empresas a nivel mundial. Los causantes detrás de los ataques son conocidos como ciberdelincuentes. Estos aprovechan vulnerabilidades existentes en los sistemas informáticos para efectuar el ataque, ocasionando robo de información confidencial y pérdidas económicas para las empresas u organizaciones afectadas. Es por ello que se busca una alternativa para disminuir este problema, una opción a considerar es el aprendizaje automático como herramienta para mejorar la seguridad informática. El presente trabajo de titulación tuvo como finalidad presentar un modelo de detección de intrusos que hace uso de la técnica propuesta en el presente trabajo de titulación, la cual combina las técnicas filter y wrapper para la selección de características en la fase de preprocesamiento de datos. El conjunto de datos utilizado para el entrenamiento y prueba de los modelos fue obtenido del repositorio GitHub². Se utilizaron algoritmos de clasificación para el entrenamiento de los modelos. En base a la métrica de exactitud se seleccionó al mejor modelo de detección de intrusos, el cual fue entrenado mediante el algoritmo RandomForest. Este modelo consiguió una media del 99,42% de exactitud con la técnica de selección de características propuesta, mejorando en un 0.10% al resultado del modelo entrenado con el mismo algoritmo pero sin el uso de la metodología propuesta. Con ello se evidencia que los modelos entrenados con la metodología propuesta proporcionan rendimientos similares a los modelos que no hacen uso de la misma, contando con

² Disponible en: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

la ventaja de eliminar aquellas características redundantes del conjunto de datos. Cabe mencionar que, el tiempo de entrenamiento de los modelos con cada uno de los algoritmos para poder evaluar su desempeño y seleccionar al mejor fue de aproximadamente un minuto con diez segundos.

Palabras clave: aprendizaje automático, ciberataque, conjunto de datos, filter, selección de subconjunto de características, wrapper .



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**INTRUSION DETECTION MODEL TO DETECT AND PREVENT THE INSERTION OF
MALWARE IN A NETWORK, BASED ON MACHINE LEARNING TECHNIQUES.**

Authors:

Dayannara Cindy Avila Maldonado
C.I. N° 0929308898
Joel Anthony Torres Urresto
C.I. N° 0958897795

Tutor: Ph.D. Franklin Ricardo Parrales Bravo

ABSTRACT

Cyberattacks are one of the main problems that affect companies worldwide. The perpetrators behind the attacks are known as cybercriminals. These take advantage of existing vulnerabilities in computer systems to make the attacks, causing theft of confidential information and economic losses for the affected companies or organizations. That is why an alternative is being sought to reduce this problem, an option to consider is machine learning as a tool to improve computer security. The purpose of this thesis work was to present an intrusion detection model that makes use of the technique proposed in the thesis work itself, which combines the filter and wrapper techniques for the selection of characteristics in the data pre-processing phase. The data set used for the training and testing of the models was obtained from the GitHub³ repository. Classification algorithms were used to train the models. Based on the accuracy metric, the best intrusion detection model was selected, which was trained using the RandomForest algorithm. This model achieved a mean of 99.42% accuracy with the proposed feature selection technique, improving by 0.10% the result of the model trained with the same algorithm but without the use of the proposed methodology. This shows that models trained with the proposed methodology provides similar performance to the models that don't use it, with the advantage of eliminating redundant characteristics from the data set. It's worth mentioning that the training time of the models with

³ Available in: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

each one of the algorithms to be able to evaluate their performance and select the best one was approximately one minute and ten seconds.

Key words: machine learning, cyberattack, data set, filter, feature subset selection, wrapper.

INTRODUCCIÓN

La seguridad de los sistemas informáticos o redes en ocasiones ha sido vulnerada, los métodos que usan los ciberdelincuentes para irrumpir en los sistemas cada vez afectan a miles de empresas. Desde visitar una página web, abrir un correo, interactuar con anuncios, puede ser el causante de un ciberataque, provocando robo de información y daños en los equipos informáticos. Cabe mencionar que a medida que avanza la tecnología, nuevas modalidades de vulnerar un sistema aparecen, es por eso que las empresas deben implementar en su seguridad, software que cuente con técnicas que permitan la detección de actividades anormales en la red. Los sistemas de detección de intrusos (IDS) cumplen con esta función, ellos se encargan de implementar software que analiza el tráfico de las redes permitiendo así detectar un posible ciberataque y tomar acciones para reducir el impacto de este. Una alternativa para lidiar con los ciberataques es utilizar el aprendizaje automático. Este es un subcampo perteneciente a la inteligencia artificial (IA) que permite a los sistemas aprender de manera automática y otorgar la facilidad a que estos puedan predecir actividades futuras.

Existen en el mercado varios sistemas de detección de intrusos, pero esta investigación trae consigo un modelo de detección de intrusos para evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático que considera la combinación de las técnicas filter y wrapper para la selección de subconjunto de características. Dicha combinación permite garantizar la diversidad y nula redundancia entre las características del modelo, obteniendo un rendimiento similar a los modelos cuyas características fueron seleccionadas por las técnicas de selección de subconjuntos existentes.

Este trabajo investigativo está conformado por cuatro capítulos. En el capítulo uno se centra en la problemática, y en los antecedentes de este. Además, se detallan casos de ciberataques, así como las causas y consecuencias. De igual forma, se menciona el inconveniente que existe en los conjuntos de datos con base en la selección de características dentro de la etapa del preprocesamiento de estos. Por otro lado, también se presentan los objetivos tanto el general como los específicos, los cuales sirven para el desarrollo del trabajo. En este capítulo también se incluye el alcance, las limitaciones del estudio, la justificación e importancia y lo que se espera obtener de esta investigación.

En el capítulo dos se presenta el marco teórico. Este incluye material que fue extraído de investigaciones similares, las cuales aportan al entendimiento del tema planteado. Adicionalmente, se declaran las variables que permiten evaluar el desempeño del modelo. Las definiciones conceptuales también forman parte de este capítulo para proporcionar un mayor entendimiento al lector de ciertos términos usados.

En el capítulo tres se detallan las metodologías utilizadas. La investigación de tipo experimental permite detallar los pasos que conllevan el desarrollo de la propuesta. En esta sección se detalla el conjunto de datos a utilizar, al igual que los algoritmos clasificadores que serán usados para el entrenamiento del modelo. Cabe destacar que en esta sección se explica la nueva técnica de selección de subconjunto de características propuesta. Adicionalmente, se detalla los entregables del proyecto, el análisis de factibilidad, operacional, técnica, legal, y por último los resultados de esta investigación.

Finalmente, en el capítulo cuatro se encuentran las conclusiones de la investigación realizada, y cómo se cumplieron cada uno de los objetivos planteados.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

Descripción de la situación problemática

Ubicación del problema en un contexto

A medida que evoluciona la tecnología, los ciberataques se ejecutan con mayor frecuencia, ya sea en una red doméstica, PyMes o en grandes empresas. Estos ataques dirigidos por ciberdelincuentes buscan cumplir objetivos, desde dañar equipos de cómputos, colapsar las redes, hasta robo de información confidencial. Los ciberataques en los últimos años se han elevado y los ciberdelincuentes han descubierto nuevos métodos para vulnerar los sistemas informáticos, así lo indica el reporte de Cisco (2018) donde detalla que: “los adversarios son cada vez más expertos en la evasión y en usar como armas los servicios de la nube y otras tecnologías utilizadas con fines legítimos” (p.3). Los ciberdelincuentes adoptan el cifrado para evitar la detección, aunque la encriptación está ligada a mejorar la seguridad, puede ser usada como herramienta por los ciberdelincuentes para ocultar la actividad de comando y control (C2) que les permite tener tiempo para intervenir y ocasionar daños. Los ciberdelincuentes utilizan canales C2 y esta práctica hace que el tráfico de malware sea casi identificable (Cisco, 2018).

Por consiguiente, una de las falencias más comunes de las empresas es pensar que contando con antivirus y firewalls es suficiente para mantener a sus equipos de cómputo e información a salvo. “(...) hoy en día no se puede pensar en la seguridad como algo estático, como un problema que se soluciona configurando un firewall o un antivirus y dejándolo ahí para que haga su trabajo” (Pastorino, 2017), se necesita de un conjunto de medidas que se complementen entre sí para reducir el riesgo de un ciberataque, debido a que actualmente las amenazas surgen a diario, por ello la seguridad informática debe encontrarse en proceso de mejora continua, realizar análisis, auditorías, evaluando resultados y estableciendo objetivos claros (Pastorino, 2017).

Cuestiones como la existencia de fallos en la seguridad de las redes, programas informáticos desactualizados, carencia de antivirus, no establecer un firewall que permita filtrar el tráfico, da paso a que los sistemas informáticos sean vulnerables y no cumplen con la seguridad adecuada para una amenaza reciente. Las organizaciones reducirían el porcentaje de ataques si estos implementaran las mínimas medidas para protegerse (Martínez Landrove, 2019).

Además, el factor humano también representa un riesgo, por esta razón, concluye Hoyos (citado en Izaguirre Olmedo & León Gavilanez, 2018) el usuario al no tener el conocimiento suficiente, permite al ciberdelincuente acceder a información importante y personal. Por otra parte, los ciberataques pueden representar pérdidas económicas para las empresas, fallos de los servicios en línea debido a la indisponibilidad de los equipos de cómputo, causando molestias a los clientes (Freire Fajardo, 2017).

A lo largo de los años empresas alrededor del mundo han sufrido de intromisiones en sus redes informáticas, siendo estas víctimas de ciberataques. Por ejemplo, en el año 2016 se perpetró un ciberataque mientras se llevaba a cabo las elecciones para la presidencia de Estados Unidos. El partido demócrata encabezado por su candidata, la Sra. Hillary Clinton, sufrió de un ciberataque

que comprometió información importante. Según la BBC (2016) la técnica que utilizaron los hackers fue mediante el método conocido como phishing, el cual se encarga de engañar al usuario mediante técnicas de ingeniería social y a partir de ello obtener información. Los hackers tuvieron acceso a las estrategias de campaña, todo esto fue ocasionado mediante correos fraudulentos que llegaron a todos los integrantes de dicho comité.

Asimismo, en el año 2017 se produjo un ciberataque. Este afectó a 150 entidades (empresas, e instituciones) alrededor del mundo. Los ciberdelincuentes lanzaron una campaña masiva de ransomware, afectando así a los sistemas informáticos. Como consecuencia de este ciberataque las entidades suspendieron sus actividades. Para la recuperación de los sistemas intervenidos los ciberdelincuentes exigían un rescate. Entre los países afectados estuvieron: Reino Unido, Estados Unidos, China, España, Italia, Vietnam y Taiwán (BBC, 2017).

La red social Twitter fue víctima de un ciberataque en el mes de julio de 2020. De acuerdo con la BBC (2020) cuentas verificadas de figuras públicas, y empresas de Estados Unidos fueron hackeadas. Las cuentas que fueron intervenidas por ciberdelincuentes postearon un comunicado fraudulento que prometía dinero bitcoin a sus seguidores si estos realizaban una donación a sus cuentas. Esta estafa se generó mediante ingeniería social.

Según Izaguirre Olmedo y León Gavilanez (2018) en su estudio llamado “Análisis de los Ciberataques Realizados en América Latina”, la región se ha visto envuelta en ataques de espionaje, robo de información, ataques dirigidos a las redes de computadoras e infección por malware. Estos sucesos ocurrieron durante los años 2009 hasta el 2017.

Izaguirre Olmedo y León Gavilanez (2018) mencionan que el Ecuador fue uno de los países más afectados por ciberataques mediante la aplicación de Pokémon GO en el año 2016. Ellos en

su investigación detallan que los usuarios descargaron la aplicación del juego en sitios no oficiales, y el mismo no contaba con el sistema de seguridad de información adecuada.

Un caso reciente de ciberataque a nivel nacional se dio el 12 de agosto de 2020. Ese día, el Instituto Ecuatoriano De Seguridad Social (IESS) suspendió el servicio en línea por intentos de ataques informáticos para salvaguardar los datos de los usuarios. Mediante un comunicado a través de su cuenta oficial de Twitter, el IESS informó sobre el ataque y la actividad inusual en las redes de la institución (IESS, 2020).

Cabe mencionar que en las empresas ecuatorianas se percibe la cibervulnerabilidad, es por ello que en los últimos meses se han reportado ciberataques. Estos ciberataques son independientes, esto quiere indicar que los ciberataques afectan tanto a las empresas públicas como privadas (Toapanta Toapanta, Coello Ocha , Naranjo Sanchez, & Gallegos Mafla, 2019).

Además, las empresas no cuentan con medidas de seguridad robustas. De hecho, implementan encriptación de los datos para salvaguardar la información de los usuarios, pero esto es una de las medidas que se deben tener en cuenta. Las empresas también emplean software antivirus, firewalls, sistemas de detección de ataques y aun así los sistemas informáticos son vulnerables.

La mayoría de las organizaciones en Ecuador mostró, a nivel individual y durante la pandemia, limitaciones propias en cuanto a ciberseguridad. Cada empresa debe diseñar procedimientos de acuerdo con estándares internacionales: buenas prácticas, metodología de ‘hacking’ ético o pruebas de penetración a los sistemas para detectar vulnerabilidad. (Diario EL COMERCIO, 2020)

En otras palabras, las empresas ecuatorianas deben tomar en cuenta los estándares internacionales para fortalecer la ciberseguridad.

Una alternativa para lidiar con los ciberataques es utilizar el aprendizaje automático. Con este se podría elaborar un modelo predictivo el cual disminuiría el riesgo de sufrir un ciberataque. Aun así, esta alternativa tiene limitaciones. La empresa de ciberseguridad ESET, menciona que en el aprendizaje automático existen limitantes y una de ellas viene dada por parte del conjunto de datos. En el caso de desarrollar un modelo para detectar intrusos, previamente se debe tener los datos suficientes. Posterior a ello, los datos deben ser divididos en: maliciosos, no infectados y potencialmente no seguros o no deseados, pero este proceso no asegura que el modelo pueda identificar los nuevos datos que ingresen al mismo. Por lo que se necesita de supervisión humana y experiencia (ESET, 2018). Además, las características redundantes e irrelevantes en un conjunto de datos ocasionan lentitud de los algoritmos y baja tasa de detección (Shao-bo, 2017). Mungloo Dilmohamud, Marigliano, Jaufeerally Fakim, y Peña Reyes (2018) también hacen eco del problema, mencionando que los grandes conjuntos de datos suelen contener características redundantes, añadiendo complejidad a su posterior tratamiento.

Situación conflicto nudos críticos

Los ciberataques surgen por falta de medidas de seguridad en los sistemas informáticos, en ocasiones por falta de capacitación al recurso humano de las empresas. El usuario del sistema informático tiende a ser el eslabón más débil, los atacantes informáticos utilizan técnicas como la ingeniería social, para persuadir o engañar a los empleados de la empresa y así sustraer información crítica de la empresa.

Uno de los ataques basados en ingeniería social se denomina phishing, “el phishing se refiere al envío de correos electrónicos que tienen la apariencia de proceder de fuentes de confianza (como bancos, compañías de energía etc.) pero que en realidad pretenden manipular al receptor para robar información confidencial” (pandasecurity, s.f).

En conclusión, la ingeniería social es un problema existente para las empresas, permitiendo que el ciberdelincuente cumpla con su objetivo mediante esta metodología.

Delimitación del problema

La presente investigación está enfocada hacia el campo tecnológico, específicamente al área de la seguridad informática y a la detección de ciberataques mediante malware.

Actualmente, el sector público en materia de ciberseguridad es el más vulnerable. Esto se debe a que no existe política de homogeneidad para identificar riesgos, debido a esto no se toman las medidas de ciberseguridad requeridas (Alvarado Chang, 2020).

De la misma manera se menciona que en el Ecuador, “en materia de seguridad se plantea que el responsable del tratamiento de datos personales implemente prácticas de seguridad integral. Una de ellas es la encriptación, cifrado o codificación de datos” (Diario EL COMERCIO, 2019). Además, de la implementación de sistemas de prevención de ataques (Mateo & Neira Cedillo, 2017). Siendo estas metodologías las adoptadas por las empresas para protegerse de ciberataques.

Por otro lado, las técnicas del aprendizaje automático deben ser consideradas para mejorar la seguridad informática debido a que, por medio de ellas, se podrá detectar intrusos, evitando los daños ocasionados por software maliciosos en los equipos informáticos.

En la creación de un modelo predictivo se ponen en marcha varias fases, pero una de las más importantes es el preprocesamiento de datos. La fase de preprocesamiento es aplicada para preparar los datos antes del entrenamiento de un modelo, por lo cual seleccionar la mejor técnica de selección de características es importante. Como lo mencionan, Pushpalatha y Gowda Karegowda (2017) las técnicas de selección de características juegan un papel vital en este proceso. Mediante ellas se podrán seleccionar aquellas características más relevantes, reduciendo la complejidad a la hora de construir modelos predictivos.

Campo: Tecnología.

Área: Seguridad Informática.

Aspecto: Detectar y evitar la inserción de malware en una red.

Tema: Modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático En la **Tabla 1** se detalla la delimitación del problema.

Tabla 1

Delimitación del problema

Delimitador	Descripción
Campo	Tecnología
Área	Seguridad Informática
Aspecto	Detectar y evitar la inserción de malware en una red
Tema	Modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático

Nota: En esta tabla se plantean los términos de análisis aplicados para la delimitación del problema conforme al contexto en donde se desarrolla la problemática. Elaborado por Dayannara Avila y Joel Torres.

Evaluación del Problema

Delimitado: La aplicación del aprendizaje automático en la seguridad informática no ha sido tan profundizada en el Ecuador. Diario EL COMERCIO (2019) menciona en uno de sus artículos, que el Ecuador se encuentra en la escala regional como uno de los últimos países en materia de ciberseguridad, solo por delante de Venezuela y Bolivia, según el índice global de ciberseguridad (GCI). Uno de los principales problemas es el robo de información. Como mencionan Rivero Pérez, Ribeiro, y Ortiz (2016) los principales inconvenientes vienen dados por el modesto despliegue de sistemas de detección basados en algoritmos de aprendizaje bajo las restricciones impuestas por los entornos reales.

Claro: Las empresas ecuatorianas sufren las consecuencias de la inserción de malware en sus redes al no fortalecer sus medidas de seguridad. Un ejemplo de aquello se observó en el año 2019 cuando la información de miles de ecuatorianos estuvo expuesta por una falla informática en un servidor (Rueda , 2020). Por otro parte, los algoritmos usados para la detección de intrusos en su mayoría se basan en firmas y no en anomalías. Esto quiere decir que, estos solo detectan ataques previamente registrados (Rivero Pérez et al., 2016).

Evidente: La empresa de ciberseguridad ESET cada año realiza un reporte de ciberseguridad en Latinoamérica. En el reporte de 2019 el Ecuador ocupó el nivel más alto de incidentes de seguridad entre los países de la región con un 65 % (ESET, 2019). Asimismo, en el reporte de 2020 los incidentes de seguridad incrementaron dando un porcentaje de 70 % (ESET, 2020). Por otro lado, en un modelo de aprendizaje automático la selección de características puede traer consigo una limitante. Los atributos irrelevantes y/o redundantes pueden no ser eliminados, dando lugar a los problemas de correlación entre características (Maseda Tarin, 2019).

Relevante: El presente trabajo de investigación proporciona una nueva metodología para seleccionar características de un conjunto de datos. De acuerdo con lo mencionado por Venkatesh y Anuradha (2019) las técnicas de filtrado en ocasiones no descartan las características redundantes e incompletas. El objetivo de esta propuesta es obtener características de mejor calidad, que permitan obtener un buen rendimiento del modelo predictivo y que las empresas, con base a este trabajo puedan adoptar este modelo o desarrollar aplicativos que mejoren sus sistemas de detección de intrusos.

Original: En el presente trabajo se propone una nueva metodología para la selección de características que, combinando las técnicas de wrapper y filter, mejora el rendimiento de los

algoritmos de clasificación. El código del aplicativo estará disponible de forma gratuita en Github⁴ para que las empresas puedan utilizar el aplicativo para mejorar sus sistemas de detección de intrusos.

Factible: La solución que se propone es posible de implementar debido a que realiza una combinación de los métodos existentes, disponibles de forma gratuita, para mejorar el aprendizaje de los modelos predictivos. Esta técnica reducirá la existencia de características irrelevantes y redundantes en el subconjunto de datos resultante.

Causas y consecuencias del problema

Los ciberataques implican pérdidas económicas, de reputación y de clientes para una empresa u organizaciones. Los objetivos de un ataque informático es obtener la información que manejan las empresas; como datos personales, datos financieros de los clientes o de personas que laboran en el interior de esta y perjudicar el funcionamiento de los equipos informáticos.

Cabe mencionar que, en la fase de preprocesamiento de datos, suelen existir problemas, específicamente en la selección de subconjunto de características. Cai, Luo, Wang, y Yang (2018) mencionan que este tipo de técnicas suelen ser inestables, debido a que el subconjunto de características resultantes se divide al azar y no garantiza la diversidad entre los subconjuntos de características seleccionadas. Es decir, si se obtiene características redundantes o irrelevantes, el conjunto de datos será muy complejo, por lo que, el modelo no tendrá un desempeño óptimo.

Además, mantener un nivel de seguridad informático bajo, falta de capacitación del personal de las empresas, ingeniería social son las principales causas de un ciberataque. Asimismo, utilizar aplicativos que posean una debilidad, puede generar que un software malicioso se infiltre en la red.

⁴ Disponible en: <https://github.com/jatu11/Trabajo-de-titulaci-n>

En la **Tabla 2** se muestran las causas y consecuencias del problema.

Tabla 2

Matriz de causas y consecuencias del problema

Causas	Consecuencias
C1. Sistemas operativos y programas instalados no actualizados.	E1. Posible inserción de malware en los equipos informáticos.
C2. Falta de capacitación al personal sobre los sistemas informáticos	E2. Probabilidad de ser víctima de un ciberataque.
C3. Softwares deficientes para detectar amenazas.	E3. Sistemas informáticos vulnerables.
C4. Visitar páginas web no seguras.	E4. Posibilidad de permitir el ingreso de algún software malicioso.

Nota: Elaborado por Dayannara Avila y Joel Torres.

Formulación del problema

El problema de la investigación se centra en los ciberataques generados por malware. Los aplicativos que las empresas utilizan para reducir ciberataques suelen ser en ocasiones poco eficientes. El problema se genera cuando el ciberdelincuente descubre cómo funciona ese software, y crea la manera de vulnerar el sistema para buscar su objetivo. El diseño o desarrollo de nuevas técnicas para la detección de intrusos, es la clave para disminuir los ciberataques que los ciberdelinquentes pretenderán realizar más adelante.

La selección de subconjunto de características puede ser una técnica muy útil si se tiene un conjunto de datos extenso, pero como mencionan, Bolón Canedo y Alonso Betanzos (2019) la diversidad de características en los subconjuntos creados por sí solos no suele ser suficientes para mejorar el desempeño del modelo. Debido que en ocasiones aplicar estas técnicas pueden dar como resultado un subconjunto de datos con características irrelevantes y redundantes para el entrenamiento del modelo.

Objetivos del proyecto

Objetivo general

Presentar un modelo de detección de intrusos para localizar la inserción de malware en una red, considerando la combinación de las técnicas filter y wrapper para la selección de características en la fase de preprocesamiento de datos.

Objetivos específicos

1. Revisar trabajos similares para conocer las técnicas de aprendizaje automático utilizadas.
2. Seleccionar los algoritmos de aprendizaje automático a utilizar.
3. Proponer una nueva metodología de preprocesamiento que considere las técnicas filter y wrapper.
4. Realizar el preprocesamiento de datos para descartar características redundantes e irrelevantes.
5. Entrenar el modelo predictivo mediante el uso de los algoritmos de clasificación seleccionados.
6. Obtener el porcentaje de efectividad de los modelos clasificadores entrenados y verificar la mejora de los mismos gracias al uso de la metodología propuesta.

Alcance del proyecto

Para realizar esta investigación se llevará a cabo la revisión de trabajos similares, obteniendo así un panorama más amplio del aprendizaje automático y la seguridad informática. Para el cumplir con el objetivo general planteado, se seleccionará el conjunto de datos correspondiente. Además, se realizará una revisión de literatura con respecto al preprocesamiento de datos con la finalidad de conocer las técnicas más usadas para la selección de características y

proponer una metodología que, a partir de la combinación de las técnicas existentes, extraiga las características más relevantes, garantizando de esta forma su diversidad y no redundancia entre ellas.

Los algoritmos serán seleccionados mediante una consulta bibliográfica. Un ejemplo de la bibliografía revisada es el trabajo de Akinsola (2017) donde hace un análisis de los algoritmos supervisados utilizados en el aprendizaje automático. También Belavagi y Muniyal (2016) en su trabajo evalúa los algoritmos supervisados que se utilizan en la detección de intrusos. Una vez seleccionados los algoritmos se les proporcionará el conjunto de datos, previamente estos datos se dividirán en conjuntos de entrenamiento y prueba. Para entrenar los modelos se escogerá los datos de entrenamiento y estos al final generarán un modelo (entrenado). En base a estos modelos obtenidos, se evaluará el porcentaje de exactitud, especificidad, F1-score y sensibilidad de cada uno al clasificar los registros del conjunto de datos de prueba.

Para el desarrollo de los pasos mencionados en el párrafo anterior, se trabajará con el lenguaje de programación Python con su respectiva versión, en este caso, se usará la versión 3.8. El software de Anaconda se usará para el procesamiento de los datos con la ayuda del IDE Syper 3. Trabajando con estas herramientas en conjunto, se llevará a cabo el objetivo de la investigación.

Justificación e importancia

El modelo desarrollado en este proyecto le puede ser útil a las empresas que no cuenten con un sistema de defensa óptimo en la detección de intrusos. El modelo será accesible de manera gratuita para que las empresas puedan ver y comprobar que, mediante el uso de diversas técnicas del aprendizaje automático, sus compañías pueden contar con un blindaje robusto en contra de los ataques utilizando software malicioso.

También se puede tomar el presente trabajo de titulación como base para futuras investigaciones, cuyos resultados sean mejorar el modelo predictivo aplicando o combinando técnicas de aprendizaje automático adicionales a las que ya se tomaron en cuenta, o incluso, optimizando alguna de las técnicas tomadas en consideración.

Limitaciones del estudio

El conjunto de datos que se utilizará se los tomará de un repositorio de Github administrado por Tek⁵, siendo este el único conjunto de datos a utilizar en el presente trabajo.

La información que será recolectada puede ser una limitante, en el caso de no encontrar la documentación necesaria para el desarrollo del trabajo.

La búsqueda de una nueva metodología para seleccionar las características considerando de que varias técnicas han sido modificadas y mejoradas.

En cuanto a software, una limitante podría ser el tema de las versiones de Python con sus respectivas librerías y la actualización de estas.

⁵Disponible en: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

CAPÍTULO II

MARCO TEÓRICO

Antecedentes del estudio

La aplicación del aprendizaje automático y la inteligencia artificial en la seguridad informática van en crecimiento. De acuerdo con Valencia Peral (2019):

Es difícil encontrar un campo de las ciencias de la información en el que se haya producido una expansión tan rápida como reciente en el de las técnicas de aprendizaje automático. En los últimos diez años, la adopción de dichas técnicas para afrontar problemas de las más diversa índole ha tenido lugar de forma fluida y continua hasta participar hoy en día en prácticamente todos los ámbitos de nuestra interacción con sistemas de información. (p. 6)

Valencia Peral (2019) en su trabajo de investigación llamado “Técnica de aprendizaje automático para la detección de ataques en el tráfico de red” realizó un análisis de las técnicas de aprendizaje automático, aplicadas a la implementación de un sistema de detección de intrusión de red. El objetivo de esta investigación fue estudiar cómo funcionaban las técnicas de aprendizaje automático y aprendizaje profundo mediante el conjunto de datos NSL-KDD proporcionado por la Universidad de New Brunswick⁶. Para desarrollar la propuesta, Valencia Peral (2019) aplicó un análisis de los requerimientos, recolección y preparación de los datos, asimismo para el entrenamiento de los modelos utilizó el conjunto de datos NSL-KDD, el cual contiene los

⁶ Disponible en: <http://nsl.cs.unb.ca/nsl-kdd/>

siguientes ataques: denegación de servicios, sondeo de fuerza bruta, accesos no autorizados de servidores remotos, e intentos de escalación de privilegios. El conjunto de datos fue etiquetado previamente, para implementar aprendizaje supervisado.

Valencia Peral (2019) codificó su solución en Python, haciendo uso de las respectivas librerías para llevar a cabo la ejecución de las técnicas clásicas del aprendizaje automático en contraposición del aprendizaje profundo.

A fin de evaluar la efectividad de las técnicas clásicas de aprendizaje automático, Valencia Peral (2019) optó por los algoritmos: árboles de clasificación, Naive Bayes, Random Forest, entre otros. Casi todos estos algoritmos tuvieron una precisión mayor del 92 %. Por otro lado, el algoritmo de Regresión Logística obtuvo una precisión del 82 %. Los resultados generados se obtuvieron con el conjunto de datos normalizado y parámetros por defecto.

Luego de conocer los resultados de los modelos entrenados por los algoritmos de aprendizaje automático clásicos, se procedió a la construcción del modelo mediante redes neuronales a partir del conjunto de datos de entrenamiento. Para llevar a cabo esta implementación se utilizó la librería Tensor Flow haciendo uso de Keras como wrapper. En este punto se necesitó de hiperparámetros, tales como: tamaño relativo del conjunto de datos, número de epochs, número de neuronas por capa oculta, función de activación, valor de dropout, conjunto de datos normalizado. Para poner a prueba los hiperparámetros, se introdujeron algunos valores: número de epochs: 1, conjunto de datos normalizado: false, tamaño del conjunto de datos: 0.001, función de activación: lineal, valor de dropout: 1. El porcentaje de precisión obtenido con esta configuración fue de 81 %, siendo este valor inferior al que se obtuvo mediante Regresión Logística.

Para una segunda prueba, Valencia Peral (2019) utilizó dos capas, 41 neuronas y 10 epochs para un 0.6 del conjunto de datos. El resultado de esta configuración fue de 92 % de precisión, acercándose más a los resultados obtenidos con los algoritmos utilizados anteriormente.

Como punto importante, Valencia Peral (2019) menciona que, a mayor complejidad de la red, mayor será su costo computacional. Asimismo menciona que, para que la red neuronal muestre mayor precisión, se necesitarán más neuronas en la capa oculta. De igual manera se elaboró una matriz de 108 configuraciones distintas, la cual arrojó diferentes porcentajes de precisión (límite entre el 93 % de precisión y la media de 87, 127150 %) dependiendo de los valores asignados en los hiperparámetros.

Como conclusión, Valencia Peral (2019) demostró que la precisión máxima la generó el modelo de árboles de clasificación mediante la técnica Classification tree con un 92.987 % y el mejor resultado generado por el modelo de red neuronal fue de 92.397 % con su respectiva configuración: tamaño relativo del conjunto de datos con 0.03, número de epochs 10, números de neuronas por capa oculta 10, función de activación relu y el conjunto de datos normalizado.

Por otra parte, Rodríguez Rama (2018) en su trabajo de fin de master denominado “Aplicación de técnicas de Machine Learning a la detección de ataques” muestra su interés por el aprendizaje automático en la ciberseguridad. Para ello, Rodríguez consideró las siguientes etapas: 1) utilización de la plataforma llamada WEKA (entorno para el análisis del conocimiento de la Universidad de Waikato), 2) elaboración de un script codificado en Python para realizar el procesamiento de los datos, y 3) uso del modelo predictivo para la detección de conexiones maliciosas.

El conjunto de datos utilizado en su investigación fue “KDD Cup 1999”. Este conjunto de datos cuenta con intrusiones en red simuladas, el mismo se lo utilizó para el entrenamiento y prueba del modelo.

Rodríguez Rama (2018) realizó el preprocesamiento de los datos, el cual consistió en la selección de atributos mediante el framework WEKA. Ranking fue el método de selección de características utilizado y que le permitió descartar los atributos de menor relevancia al evaluar sus valores de correlación y ganancia de información. Posterior a ello, los atributos categóricos pasaron a ser atributos numéricos (0, 1) respectivamente. Esto se realizó con la ayuda de la librería Scikit-learn y la función LabelEncoder, que se encuentra incluida en sklearn.preprocessing. De igual manera, Rodríguez Rama (2018) concluyó el preprocesamiento aplicando el escalado de características y así obtuvo un conjunto de datos normalizado.

La metodología usada en este trabajo se sostuvo en 4 puntos, a saber: 1) investigación, 2) selección de herramientas, 3) desarrollo del sistema y 4) documentación del trabajo.

Rodríguez Rama (2018) utilizó diferentes algoritmos de aprendizaje supervisado (algoritmos de clasificación). Con la ayuda de WEKA se entrenaron modelos de clasificación mediante 5 algoritmos, a saber: Random Tree, Random Forest, Naive Bayes, SVM, y Regresión Logística. Adicionalmente, se utilizó una implementación en el lenguaje de programación java del algoritmo C4.5, denominado J48, a los cuales se les proporcionó el 85 % del conjunto de datos para el entrenamiento y el 15 % para la prueba. Además, se utilizó los parámetros por defecto para cada algoritmo.

En virtud de los resultados, la mayoría de los modelos obtuvieron un 99 % de precisión, excepto el modelo entrenado por el algoritmo Naive Bayes, que obtuvo un 93 %.

Posteriormente, se seleccionó al mejor modelo en términos de precisión (frecuencia con la que el modelo predice correctamente) y rendimiento (rapidez y ejecución del software) de estos. Se concluyó que los algoritmos basados en técnicas de árboles de clasificación proporcionaron los mejores resultados.

Random Tree obtuvo una precisión del 99.9271 %, con un tiempo de entrenamiento del modelo de 7.71 segundos, mientras que Random Forest obtuvo una precisión del 99.609 % con un tiempo de 3887.39 segundos. De igual manera, el resto de los modelos arrojaron distintos resultados. Regresión logística obtuvo 99.8907 % de precisión y un tiempo de 6016.51 segundos, SVM consiguió una precisión de 99.9055% y un tiempo de 486.680 segundos, Naive Bayes generó una precisión del 93.0165% y un tiempo 5.1 segundos. En conclusión, Random Tree fue el mejor algoritmo de clasificación en términos de rendimiento, mientras que con Regresión logística obtuvo mejor precisión, aunque el entrenamiento tomó mucho más tiempo (Rodríguez Rama, 2018).

Rodríguez Rama (2018) buscó mejorar los resultados en precisión y rendimiento de los modelos entrenados con árboles de clasificación. Para ello, utilizó el conjunto de datos completo en una aplicación que se diseñó, la cual permitía modificar los datos, entrenar al modelo y conocer la precisión de este. Para el desarrollo de la aplicación utilizó el lenguaje de programación Python con sus respectivas librerías (Scikit Learn, NumPy, Matplotlib, pandas) y el script que usó fue escrito en Jupyter notebook.

Como conclusión, el nuevo porcentaje de precisión fue de 99.984 % con un tiempo de entrenamiento del modelo de 49 segundos. Cabe mencionar que se evaluó la precisión con diferentes modificaciones en el conjunto de datos y estos influyeron en el porcentaje de efectividad.

Mehmood y Rais (2016) en su artículo denominado “Machine Learning Algorithms In Context Of Intrusion Detection” realizaron una comparación de algoritmos de aprendizaje supervisado para la detección de anomalías.

Para elaborar su artículo utilizaron el conjunto de datos pertenecientes al programa de evaluación DARPA’ 98 ID. Este conjunto de datos también es conocido como KDD99. Debido a que el conjunto de datos contenía demasiados registros, Mehmood y Rais (2016) hicieron uso de la versión pequeña de este llamado `kddcup.data_10_percent` (KDD99_10 %). El conjunto de datos se dividió en 22 ataques para el conjunto de entrenamiento (492021 instancias) y 14 ataques para el conjunto de prueba (311029 instancias).

El conjunto de datos contenía 4 tipos de ataques: Root-to-Local (R2L), Denegación de Servicio (DoS), Sonda, Usuario a Raíz (U2R), y una clase de datos legítima llamada Normal. Las instancias de cada clase estaban conformadas de 41 características.

Mehmood y Rais (2016) eligieron los algoritmos máquina de soporte (SVM), Naive Bayes y C4.5. Adicionalmente, utilizaron una tabla de decisión para la detección de anomalías. Cabe mencionar que la tabla de decisión la usaron para tabular condiciones y acciones. Los resultados que proporciona la tabla de decisión se las conocen como regla. Los algoritmos seleccionados se implementaron en la plataforma WEKA 3.7.

Mehmood y Rais (2016) compararon los modelos entrenados, pero decidieron hacerlo por tipos de ataques. Las métricas de evaluación seleccionadas fueron: tasa de verdaderos positivos, falsos negativos, y precisión del algoritmo.

Mehmood y Rais (2016) iniciaron con el proceso de comparación entre modelos. El primer ataque seleccionado para evaluar la precisión de los mismos fue el ataque DoS. De manera general, Mehmood y Rais concluyeron que todos los modelos tuvieron un buen rendimiento. Además,

detallaron que la tasa de verdaderos positivos (TPR) de los modelos estuvo entre el 85 % y 100 %, la tasa de falsos negativos (FPR) fue baja, estos no superaron ni el 5 % y en cuanto a precisión los algoritmos superaron el 95 %. En esta primera comparación, la tabla de decisión y el modelo entrenado por el algoritmo C4.5 fueron los mejores. Ambos obtuvieron un TPR alto y la mayor precisión, superando el 98 %.

A continuación, Mehmood y Rais (2016) utilizaron la clase normal para evaluar la precisión de los modelos. En esta segunda comparación la tabla de decisión y el algoritmo SVM obtuvieron el mismo TPR (99%). Sin embargo, estos obtuvieron el mayor FPR en comparación con los demás. Mehmood y Rais concluyeron que el mejor modelo fue el entrenado por el algoritmo C4.5 con una precisión mayor del 95 %.

De la misma manera, se utilizó el ataque Sonda para evaluar a los modelos. En esta clase, el modelo entrenado por el algoritmo SVM consiguió un TPR muy bajo. Mientras que el resto obtuvo un TPR muy alto, superando el 90 %. En cuanto al FRP de los modelos, el modelo entrenado por el algoritmo Naive Bayes obtuvo un porcentaje alto, aproximándose al 10 %. Y con respecto a la precisión C4.5 fue el que más se aproximó al 100 %. Mehmood y Rais como punto importante mencionaron que el modelo Naive Bayes fue el clasificador con más baja precisión.

Finalmente, Mehmood y Rais (2016) compararon los modelos con los dos ataques restantes (R2L, U2R) y concluyeron que, en el ataque R2L todos los modelos no obtuvieron un resultado convincente, debido que para esta clase los datos de entrenamiento fueron muy pocos. Pero seleccionaron la tabla de decisión como la mejor por su proximidad al 99 % en precisión. Por otra parte, en la clase U2R todos los modelos obtuvieron un 90 % de TPR. Naive Bayes obtuvo un porcentaje elevado en FPR mientras que los demás generaron un 0 %. Mehmood y Rais para esta clase seleccionaron a C4.5 como el mejor, con un porcentaje mayor del 97 % en precisión.

Luego de obtener los resultados de todos los modelos con las diferentes clases, Mehmood y Rais (2016) concluyeron que, la precisión de C4.5 fue muy alta con respecto a los demás, e igualmente su tasa de clasificación errónea fue baja. Por otro lado, Naive Bayes fue el modelo con más clasificación errónea, así como una precisión baja.

En esta primera parte de antecedentes del estudio se evidenció como funciona de manera general un modelo que ha sido entrenado previamente mediante un algoritmo y un conjunto de datos. Los trabajos presentados utilizaron conjunto de datos distintos, pero todos ellos llegaron a la misma conclusión, a saber: los algoritmos de árboles de clasificación fueron los que generaron los mejores resultados.

Dentro la creación de un modelo hay un punto importante de vital consideración como lo es la selección de características y es por eso que a continuación se presentan dos trabajos con respecto a este tema.

Mungloo Dilmohamud et al. (2018) En su trabajo titulado “A Comparative Study of Feature Selection Methods for Biomarker Discovery” compararon diferentes técnicas de selección de características. Como punto importante mencionaron que utilizar un determinado conjunto de datos con diferentes técnicas de selección no siempre proporcionan los mismos resultados. En esta comparativa usaron 10 técnicas las cuales fueron: SAM, LIMMA, Rank Product, ReliefF, Fisher Score, MRMR, ExtraTrees, SVM-RFE, F y SVM. Estos métodos se implementaron en R y se ejecutaron en WEKA. Para la primera parte de este trabajo utilizaron el conjunto de datos Golub y MILE. Estos datos se pusieron a prueba con las 10 técnicas seleccionadas. Para que estos métodos tengan consistencia en el subconjunto resultante, se procedió a ejecutar estos métodos varias veces. Cabe mencionar, que el número de características seleccionadas variaron dependiendo de la configuración y métodos aplicados. Con respecto a esto, se promedió la lista de

características seleccionadas donde 15 de ellas se mantuvieron para medir la similitud. La similitud fue representada por medio de matrices, las cuales mostraron el porcentaje de biomarcadores comunes entre los distintos métodos aplicados, sobre los conjuntos de datos MILE y Golub (Mungloo Dilmohamud et al., 2018).

En la segunda parte de este trabajo Mungloo Dilmohamud et al. (2018) utilizaron 2 métodos con 8 distintos conjuntos de datos. Aquí, Mungloo Dilmohamud et al. (2018) se percataron de que los dos métodos seleccionados (SAM Y LIMMA) arrojaron resultados muy similares. Sin embargo, los resultados obtenidos mostraron que la similitud no es tan alta para todos los conjuntos. Esto se debió a que las clases de los conjuntos de datos (multiclase (consta de dos o más respuestas) o binario (consta de una sola respuesta)) van a interferir en los resultados. Los problemas multiclase favorecen a una mayor similitud que los problemas binarios. El 75% de los conjuntos de datos multiclase conducen a correlaciones ($> 50\%$) mientras que solo el 20% de los conjuntos de datos binarios. Mungloo Dilmohamud et al. (2018) mencionaron que la razón detrás de esto debe explorarse más a fondo. Además, los autores mencionaron que el porcentaje de similitud fluctúa a medida que aumenta el número de características consideradas.

En la literatura existen trabajos que implementan nuevas técnicas para seleccionar características. Un ejemplo de esto es el trabajo de Bataghva Shahbaz, Wang, Behnad, y Samarabandu (2016) ellos propusieron mejorar la selección de características basada en la correlación. Esto lo consiguieron considerando la correlación entre subconjuntos de características y etiquetas de clase (este trabajo seleccionó características para un sistema de detección de intrusos). Las métricas consideradas fueron selección de características basada en la correlación (CFS) y la incertidumbre simétrica (SU) estas dos métricas permitieron medir el nivel de dependencia entre las características. Cada una de estas métricas tuvo un determinado trabajo. La

primera, CFS se utilizó para eliminar las características redundantes y mantener las relevantes. Mientras que SU se utilizó en el subconjunto seleccionado para la eliminación de características que no colaboran con otras características del mismo subconjunto. Para la selección de características se hizo uso del conjunto de datos NSL-KDD.

Para evaluar el rendimiento de la propuesta Bataghva Shahbaz et al. (2016) utilizaron otras técnicas de selección de características basadas técnicas de filtros (Information gain, la ratio de ganancia, el chi-cuadrado). Además, se utilizaron diferentes algoritmos de clasificación. El mejor resultado con base a la tasa de detección, lo tuvo el clasificador C4.5 mediante la técnica propuesta, obteniendo un porcentaje de 86.1 %. Mientras que utilizando Information gain con el mismo clasificador se obtuvo un 80.1 %. Asimismo, el rendimiento en cuanto a términos de falsos positivos, la técnica propuesta fue la mejor y obtuvo un 12.5 % con el clasificador C4.5. Por otro lado, la técnica de Information gain para el mismo clasificador fue de 19.1 %. En términos generales, la técnica propuesta por el autor consiguió superar a las demás.

En síntesis, estos trabajos al igual que otros similares que existen en la literatura tienen un mismo objetivo y es seleccionar las mejores características para sus modelos y así mejorar el rendimiento de estos. La elección de las técnicas de selección de características dependerá de cada autor con base a su necesidad, problema a solucionar, criterio y experiencia.

Fundamentación teórica

Esta investigación abordó el tema de aprendizaje automático como alternativa para mejorar la seguridad informática. Entiéndase por seguridad informática a

la disciplina que, con base en políticas y normas internas y externas de la empresa, se encarga de proteger la integridad y privacidad de la información que se encuentra

almacenada en un sistema informático, contra cualquier tipo de amenazas, minimizando los riesgos tanto físicos como lógicos, a los que está expuesta. (Baca Urbina, 2016, p. 12)

Las personas que se desempeñan en el campo de la seguridad informática se encargan de levantar la seguridad de las redes. Estas implementan normas que impiden o dan acceso a las solicitudes que realizan los diferentes depósitos en la red.

En la seguridad informática existen 3 conceptos principales, los cuales son:

Riesgo:

Se lo define como “el análisis de vulnerabilidades en un sistema informático. El riesgo permite tomar decisiones para proteger mejor el sistema” (Avenía, 2017, p. 11). Es decir, todos los dispositivos que se encuentran conectados a internet corren el riesgo de sufrir un ciberataque explotando sus vulnerabilidades. Siendo el riesgo, una posible amenaza de ataque.

Amenazas:

Se las define como “una circunstancia que tiene el potencial de causar daños o pérdidas” (Avenía, 2017, p. 12). En otras palabras, las amenazas son los posibles ciberataques que pueden sufrir los sistemas informáticos de una red.

Por otra parte, se tiene a las

Vulnerabilidades:

Estas se caracterizan por ser “una debilidad del sistema informático que puede ser utilizada para causar un daño. Las debilidades pueden aparecer en cualquiera de los elementos de una computadora, tanto en el hardware, el sistema operativo, cómo en el software” (Avenía, 2017, p. 12).

En resumen, si no se cuenta con software de seguridad o incluso se utilizan de forma ilegal los dispositivos serán vulnerables a ataques que se encuentran en el exterior. También puede

ocurrir que, los dispositivos que se conecten a una red cuenten con programas de seguridad licenciados y actualizados, pero las políticas de seguridad de la red no son eficientes o robustas. Por lo cual, la información que viaja a través de la red enviada por los dispositivos puede verse comprometida en caso de un ciberataque.

Por consiguiente, estos conceptos dan lugar a un término muy conocido en la seguridad informática, denominado ciberataque.

Ciberataque:

Se denomina ciberataque al daño o modificación de un sistema informático. Este tipo de ataque está dirigido por personas o grupos organizados que aprovechan vulnerabilidades existentes en los sistemas, para cumplir con ciertos objetivos. Entre estos objetivos está el de robo de información confidencial como también obtener algún beneficio económico (Alvear Reinoso, 2019).

Entre los ciberataques más conocidos se encuentran:

Ingeniería social:

Conjunto de técnicas psicológicas y también habilidades sociales que se utilizan de forma consciente para lograr la obtención de información de terceras personas o para lograr que una persona realice las acciones que permita al ingeniero social lograr su objetivo.

(Sebastián y Nelson, citado en Vallejo de la Torre et al., 2018, p. 54)

Todo esto aprovechándose de los errores o vulnerabilidades humanas, debido que el recurso humano de las organizaciones no cuenta con el conocimiento suficiente para identificar y mitigar esta amenaza.

Phishing:

Se refiere a un método que consiste en “estafar y obtener información confidencial de forma fraudulenta como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria de la víctima” (InfoSpyware, citado en Vallejo de la Torre et al., 2018, p. 54). Este tipo de ataque en su mayoría se da mediante la suplantación de identidad de entidades financieras engañando al usuario para que le brinde su información y poder hacer uso a conveniencia de los datos sustraídos.

Spoofing:

Consiste en crear tramas TCP/IP utilizando direcciones IP falsas. La idea de esto es suplantar la identidad de una máquina para conseguir acceso a recursos de un tercer sistema. Por lo que un ordenador que no cuente con las credenciales de acceso a una red puede adoptar la identidad de otra máquina que sí posea estas credenciales para infiltrarse en la red, causar estragos en ella, en los sistemas informáticos o robar información que circula en la red (Zona Virus, citado en Vallejo de la Torre et al., 2018).

Ataques de inyección de código SQL:

“Es un ataque en el cual se inserta código malicioso en las cadenas que posteriormente se pasan a una instancia de SQL Server para su análisis y ejecución” (Segu Info, citado en Vallejo de la Torre et al., 2018 p. 60). Todas las organizaciones o la gran mayoría cuentan con sitios web y la información que registran o presentan en ellas reposan en bases de datos. Este tipo de ataque informático altera el procedimiento de obtención o registro de información para obtener los datos que ellos requieran y poder sacar provecho de esta sustracción ilegal de información.

Ataques de autenticación:

“Este tipo de ataque tiene como objetivo engañar al sistema de la víctima para ingresar al mismo” (EcuRed, citado en Vallejo de la Torre et al., 2018, p. 62).

Exploits:

“Es un programa o código que "explota" una vulnerabilidad del sistema o de parte de él para aprovechar esta deficiencia en beneficio del creador del mismo” (Segu Info, citado en Vallejo de la Torre et al., 2018, p. 63).

Malware:

El término malware (también conocido como software malicioso o software malintencionado) hace referencia a todo tipo de programas diseñados específicamente para dañar un ordenador o una red o para obtener algún tipo de beneficio o hacer mal uso del mismo. (Universidad de Jaén, 2018, p. 3)

Todos estos programas denominados malware actúan de manera distinta, pero siempre el objetivo es el mismo. Desde perjudicar equipos de cómputos hasta obtener información de los usuarios del sistema afectado.

Para tener una idea de cómo actúan los malware, la Universidad de Jaén menciona que:

Suelen registrar pulsaciones de teclas (keylogging) o controlan la actividad de nuestro equipo, provocando ingresos involuntarios a páginas web, envío de correos electrónicos o cualquier otra actividad sin nuestro conocimiento. Los efectos del malware pueden ser inofensivos como una molestia que no pase a mayores o algo tan grave con el robo de identidad o datos bancarios. (Universidad de Jaén, 2018, p. 3)

Virus:

Son un tipo de software malicioso cuyo objetivo es el de alterar el funcionamiento regular de un ordenador, sin contar con el permiso del usuario. Normalmente estos programas reemplazan archivos ejecutables por otros programas infectados y pueden dañar la información almacenada en el ordenador (Universidad de Jaén, 2018).

Gusanos:

Son un tipo de malware similar al virus, se replican automáticamente usando una red informática para enviar copias de sí mismo a los ordenadores conectados a esa red (Universidad de Jaén, 2018).

Troyanos:

Son programas destructivos que se hacen pasar por una aplicación legítima. Este tipo de software malicioso actúa de manera que el usuario espera que lo haga, pero en segundo plano está realizando otros procesos sin que el usuario tenga conocimiento de esta actividad (Universidad de Jaén, 2018).

Spyware o software espía:

Es un programa que extrae información sobre los usuarios sin su permiso. Su principal objetivo es enviar información del sistema donde están instalado o abrir una puerta trasera para que otro ordenador pueda tener acceso al ordenador donde se encuentra instalado (Universidad de Jaén, 2018).

Ransomware:

Es un malware que encripta los archivos que se encuentra en los equipos de cómputo. Este exige al usuario una cantidad de dinero a cambio de recuperar el acceso a la información. Existen dos tipos de ransomware: lockscreen y cryptolockers (ESET, 2017).

Ransomware de tipo lockscreen:

Niega el acceso y el uso del equipo, mostrando una pantalla de bloqueo impidiendo así que el usuario ejecute acciones sobre el equipo infectado. Este tipo de ransomware no encripta los archivos por lo que la información puede ser recuperada (ESET, 2017).

Ransomware criptográfico:

Utiliza algoritmos para cifrar archivos que se encuentran en el equipo infectado. Este cambia la estructura de estos impidiendo así su lectura, para recuperar los archivos a su estado inicial debe usarse una clave que solo el ciberdelincuente conoce. En la pantalla del equipo infectado se muestra un mensaje que indica que sus archivos se encuentran cifrados. Además, el ciberdelincuente exige dinero a cambio de los archivos descifrados (ESET, 2017).

“Un método típico de infección de ransomware es a través de un correo electrónico falso, que habitualmente asegura provenir de una empresa conocida, una entidad bancaria o una agencia gubernamental” (ESET , 2017, p. 7).

Para mitigar el impacto de los ataques mencionados anteriormente se utiliza un sistema de detección de intrusos (IDS). Este “es un componente más dentro del modelo de seguridad de una organización. Consiste en detectar actividades inapropiadas, incorrectas o anómalas desde el exterior-interior de un sistema informático” (Segu Info, citado en Vallejo de la Torre et al., 2018, p. 46). Es decir, los sistemas de detección de intrusos se encargan de mantener un constante monitoreo de la red. Este permite detectar e impedir algún ciberataque.

Cualquier sistema de detección de intrusos, sea cual sea el mecanismo en el que esté basado, debería contar con las siguientes características:

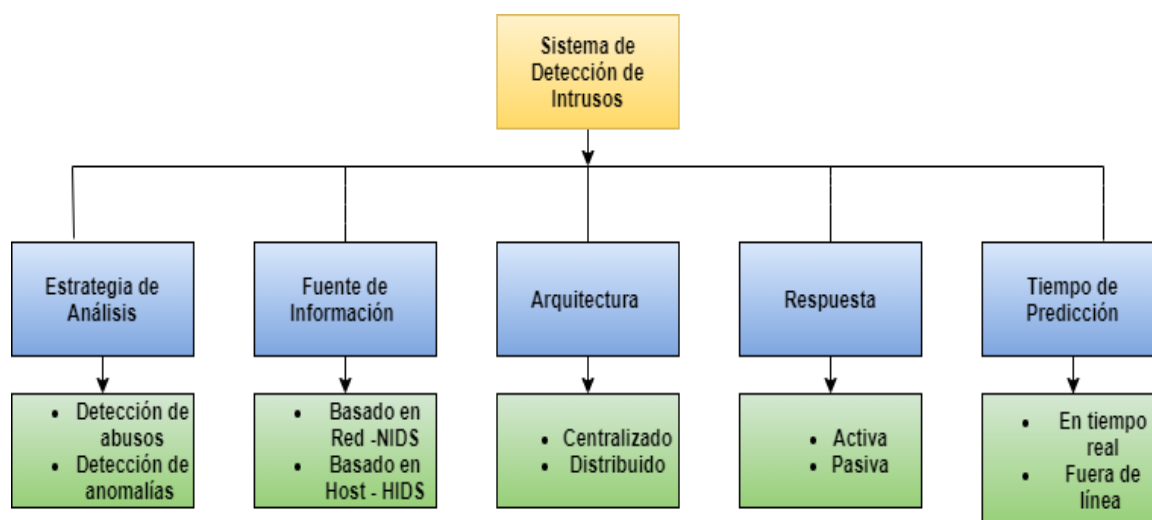
Debe funcionar continuamente sin supervisión humana. El sistema debe ser lo suficientemente fiable para poder ser ejecutado en background dentro del equipo que está siendo observado. Sin embargo, no debe ser una "caja negra" (debe ser examinable desde el exterior) (Segu Info, citado en Vallejo de la Torre et al., 2018, p. 46).

“Debe ser tolerante a fallos en el sentido de que debe ser capaz de sobrevivir a una caída del sistema” (Segu Info, citado en Vallejo de la Torre et al., 2018, p. 46). Los sistemas de detección de intrusos (IDS) se clasifican de acuerdo a diferentes criterios, a saber: 1) Estrategia de análisis,

2) Fuente de información, 3) Arquitectura, 4) Respuesta y 5) Tiempo de espera. Cada una de ellas se compone de diferentes tipos. La estrategia de análisis se divide en detección de anomalías, y detección de abusos. La fuente de información está compuesta por IDS basados en red y basados en host. Si se menciona al criterio de arquitectura, este se divide en centralizado y distribuido. Asimismo, por el tipo de respuesta se divide en activa y pasiva. Y de acuerdo al tiempo de predicción se divide en real o en línea. En la **Figura 1** se muestra el esquema de clasificación de un IDS.

Figura 1

Sistema de Detección de Intrusos



Nota: En esta figura se muestra la clasificación de un IDS. Adaptada de (Sanna Morales & Londoño Castaño, 2018).

Para entender mejor la clasificación o tipos de IDS, Llopis Polvoreda (2017) detalla algunos de los tipos que existen:

La clasificación por situación o también conocida como fuente de información se divide en dos: IDS basado en host (HIDS) e IDS basado en red (NIDS).

Un HIDS solo procesa datos asociados a un recurso. Lo que significa que, el IDS basado en host solo evaluará al ordenador que reciba el ataque. Por otro lado, un NIDS procesa los datos

asociados a varios recursos por lo que analizará los datos de toda la red o subredes, dependiendo de la configuración que se asigne (Llopis Polvoreda, 2017).

Por otra parte, la clasificación según la técnica de análisis se divide en dos: detección de usos anómalos y detección de usos indebidos.

La detección de usos anómalos genera alertas sobre actividades fuera de lo normal, debido que esta se apoya en comprender cual es el tráfico “normal” de una red. En la detección de usos indebidos, a diferencia con la técnica anterior, no reconoce lo que sería “normal” en el tráfico de la red. Este tipo de detección conoce los ataques o actividades “anormales” que se han producido y con base a ello, lo detecte y envía una alerta (Llopis Polvoreda, 2017).

Las respuestas activas y pasivas corresponden a la clasificación según su naturaleza o tipo de respuesta. En la respuesta pasiva se detecta un posible ataque o violación a la seguridad, se registra la información que fue detectada del ataque y se genera una alerta. El actuar de una respuesta activa varía en que, si se detecta alguna actividad ilegal, este tomará medidas inmediatas, como puede ser el bloqueo de actividades o intercambio de información (Llopis Polvoreda, 2017).

Para reforzar los sistemas de detección de intrusos se utiliza el aprendizaje automático.

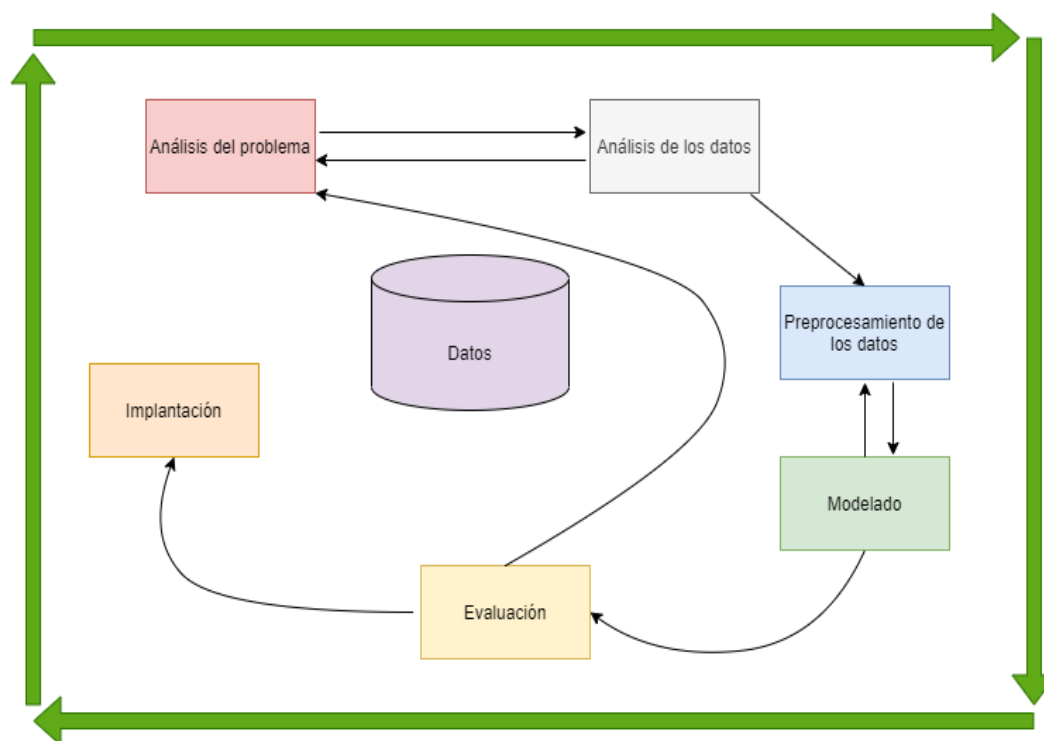
Aprendizaje automático:

El aprendizaje automático es una rama de la Inteligencia Artificial. Esta les permite a las máquinas aprender de sí mismas a través del análisis de datos, los cuales llevan el nombre de datos de entrenamiento. El modelo de aprendizaje automático forma patrones, estos patrones son usados para que el modelo pueda aprender y le permita realizar predicciones (Valdez Alvarado A. R., 2017). En el aprendizaje automático es común utilizar la metodología CRISP-DM. Dicha metodología cuenta con 6 etapas. La primera etapa se enfoca en el análisis del problema, la segunda en el análisis de los datos, la tercera en el preprocesamiento de datos, la cuarta en el modelado, la

quinta en la evaluación y la sexta en la implementación (Espinosa Zuñiga, 2020). A continuación, la **Figura 2** muestra la metodología CRISP-DM.

Figura 2

Metodología CRISP-DM



Nota: Fases de la Metodología CRISP-DM. Elaborado por Dayannara Avila y Joel Torres

Dependiendo del problema a abarcar se puede emplear un aprendizaje supervisado o un aprendizaje no supervisado.

Aprendizaje supervisado:

En el aprendizaje supervisado los datos son previamente etiquetados por un experto o de manera semi automática. En este aprendizaje se conoce de antemano la salida de los datos, debido a que desde un principio se conocen los datos de entrada. Es decir, los datos de entrada proporcionan la salida esperada (Valdez Alvarado A. , 2019).

Aprendizaje no supervisado:

En este tipo de aprendizaje los datos no son etiquetados previamente, se presentan tal como se los recibe. Su objetivo es encontrar patrones de entrada, los cuales determinan las relaciones de diferencia, similitud o asociación. En este caso, no se conoce de antemano la salida de los datos de entrada (Valdez Alvarado A. , 2019).

Luego de haber definido los tipos de aprendizaje y conocer sobre ellos, se seleccionó al aprendizaje supervisado para llevar a cabo el desarrollo de la propuesta del presente trabajo de titulación. Adicionalmente , es necesario conocer un punto importante para la creación de modelos como lo es el preprocesamiento de los datos.

Preprocesamiento de Datos

Consiste en la limpieza, integración, normalización, y transformación de los datos. El objetivo principal del preprocesamiento de datos es obtener datos correctos y útiles para los algoritmos (García , Luengo, & Herrera, 2016).

Para reducir la complejidad de los datos, eliminar el ruido y eliminar datos irrelevantes existen ciertos procesos (García et al., 2016). Uno de ellos es la selección de características.

Selección de características

Consiste en seleccionar un subconjunto de características del conjunto original de modo que estas sean las más aptas para la tarea a realizar (Spasova Dimitrova, 2017). En este paso se busca seleccionar las características más importantes en el conjunto de datos. A saber, que en el conjunto de datos existen características relevantes, irrelevantes, y redundantes. Se dice que una característica es irrelevante “cuando el conocimiento de su valor no aporta nada para predecir la clase objetivo” (Spasova Dimitrova, 2017, p. 10). La clase objetivo es la característica que permite medir el grado de severidad del problema que se estudia .

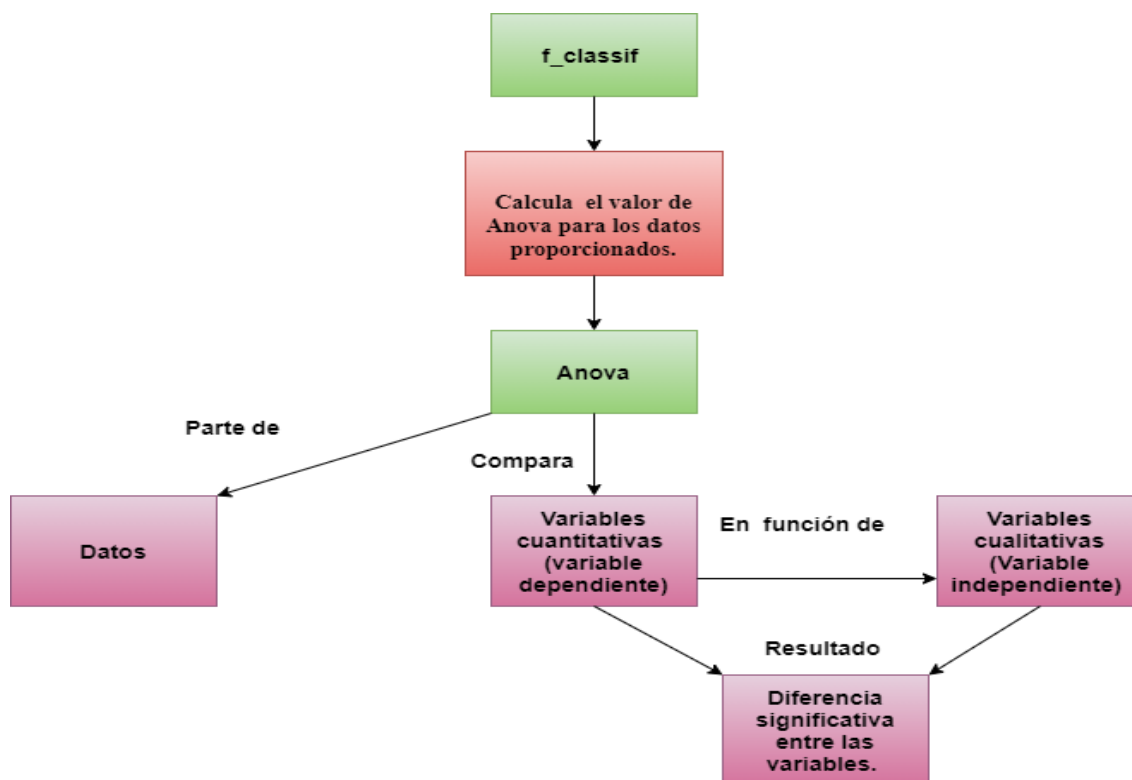
Las características redundantes son aquellas que no aportan la suficiente información en el conjunto de datos y pueden ser eliminadas (Spasova Dimitrova, 2017). Mientras que las características relevantes son lo opuesto a las redundantes (Spasova Dimitrova, 2017). La eliminación de una característica relevante lleva al modelo a disminuir su precisión y la información que esta posee no puede ser proporcionada por las otras características (Spasova Dimitrova, 2017).

Para seleccionar las mejores características se deben aplicar técnicas que ayuden a este proceso. Entre las técnicas de selección de características se encuentran: las técnicas de filter, wrapper, embedded.

Técnica filter (filtro)

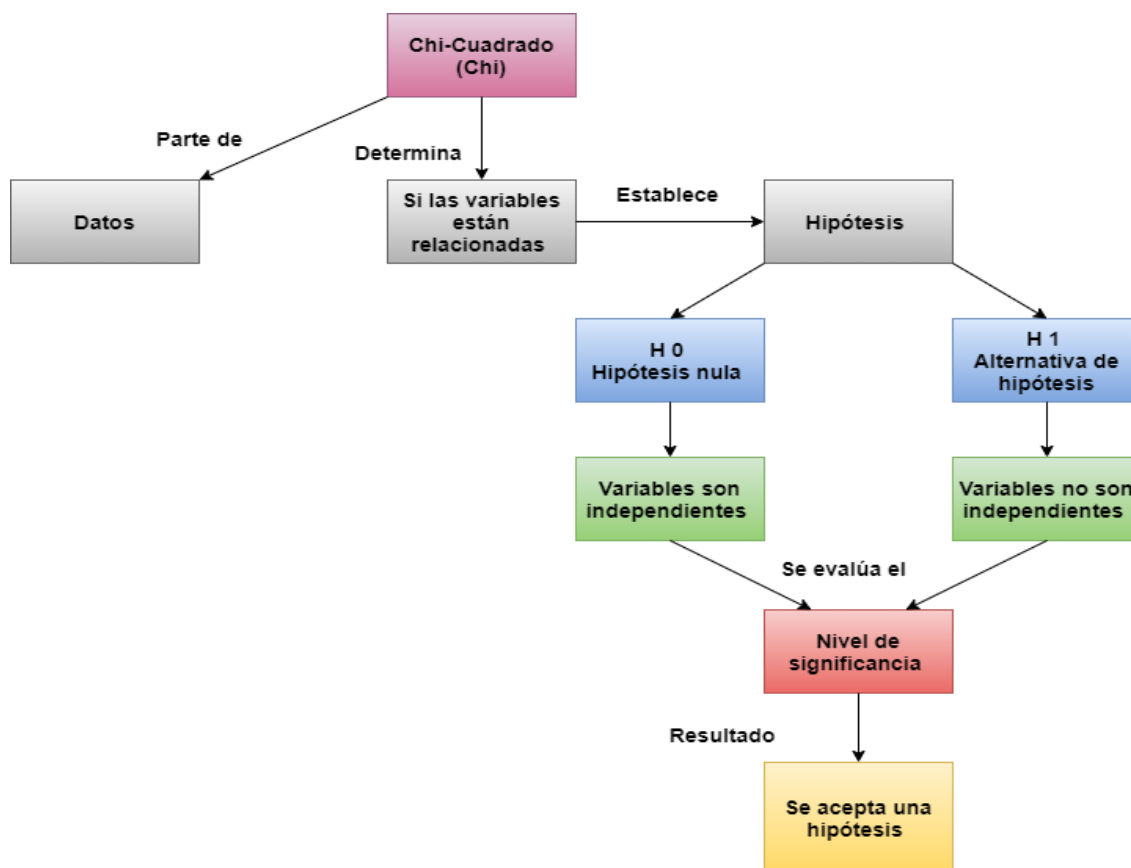
La técnica filter asigna valores para cada una de las características por medio de una función, esta las clasifica y las ordena de mayor a menor. Las características con mayor puntuación son seleccionadas para ser usadas en el método de aprendizaje a utilizar. Este método es de bajo coste computacional a diferencia de los demás (Maseda Tarin, 2019). Cabe mencionar, que el proceso de selección de características mediante esta técnica es independiente del algoritmo a utilizar.

Posterior a ello, se seleccionó los métodos filter que serán usados en el desarrollo de la propuesta del presente trabajo de titulación. Entre los métodos elegidos se encuentran: f_classif, Chi-cuadrado (Chi2) y mutual_info_classif. En la **Figura 3** se detalla en un esquema el método f_classif. Este método calcula el valor de Anova para los datos proporcionados (scikit-learn, s.f).

Figura 3*Método f_classif*

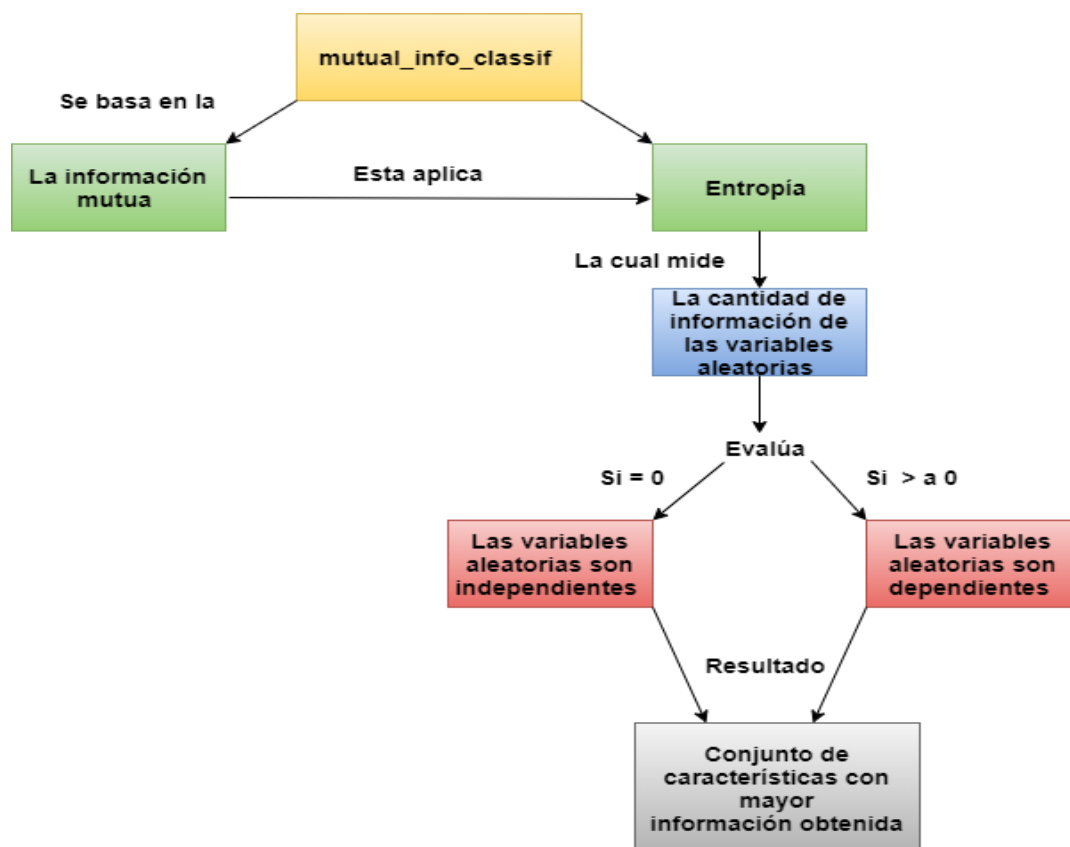
Nota: Esquema del método f_classif. Fuente: Datos de la investigación

El segundo método seleccionado fue Chi-cuadrado (Chi) en la **Figura 4** se presenta su esquema. La finalidad de este método es conocer si las variables son independientes de la clase o no, aceptando una hipótesis (scikit-learn, s.f). Es decir, La hipótesis nula para la prueba chi2 es que dos variables categóricas son independientes. Entonces, un valor más alto de la estadística chi2 significa que dos variables categóricas son dependientes y más útiles para la clasificación.

Figura 4*Método Chi-Cuadrado (Chi)*

Nota: Esquema del método Chi- Cuadrado (Chi). Fuente: Datos de la investigación

De igual manera en la **Figura 6** se muestra el esquema del método `mutual_info_classif`. Este método calcula el valor de la información mutua para cada una de las variables independientes con respecto a la variable dependiente y selecciona las que tienen más información obtenida. Tiene como finalidad medir la dependencia entre dos variables aleatorias con el valor objetivo (Yinghua, Yongkang, & Liping, 2019).

Figura 5*Método mutual_info_classif*

Nota: Esquema del método mutual_info_classif. Fuente: Datos de la investigación

Luego de conocer la técnica filter con sus respectivos métodos lo siguiente es conocer sobre la técnica wrapper.

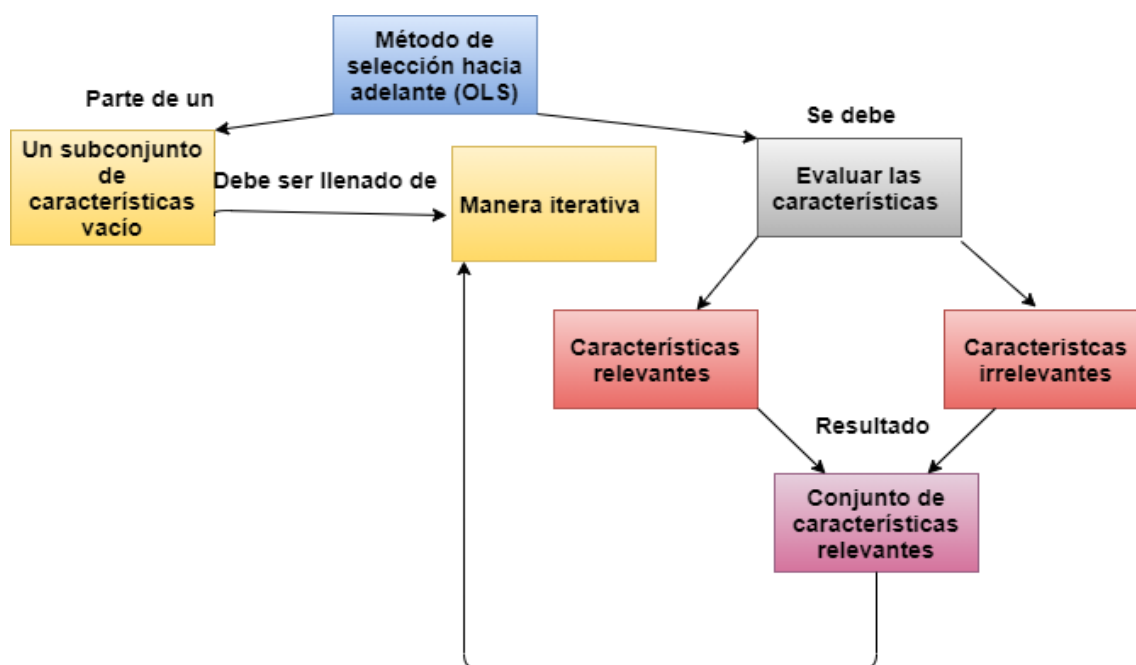
Técnica wrapper (envoltura)

Utiliza algoritmos de aprendizaje para conocer la eficacia de las características basándose en el nivel de predicción del algoritmo. Esta técnica elige subconjuntos de características y les asigna un valor. Como resultado se obtiene al mejor conjunto de características con base a la evaluación del algoritmo aplicado (Maseda Tarin, 2019). El método seleccionado para aplicar la técnica wrapper en el desarrollo de la propuesta fue el método de selección hacia adelante (OLS).

En la **Figura 6** se muestra el esquema de dicho método. El método selección hacia adelante comienza con un subconjunto de características vacío, que secuencialmente irá introduciendo características relevantes, hasta llegar al número total de características que se haya indicado previamente (Maseda Tarin, 2019).

Figura 6

Método de selección hacia adelante



Nota: Esquema del método de selección hacia adelante. Fuente: Datos de la investigación

Técnica embedded (embebidas)

El algoritmo de aprendizaje del clasificador incluye la búsqueda del subconjunto óptimo de características. Además, el algoritmo de aprendizaje sabe que se está realizando la selección. Tiene un menor coste computacional, a diferencia de las técnicas de wrappers (Spasova Dimitrova, 2017).

La aplicación de las técnicas anteriormente descritas deja a los datos listos para el siguiente paso. El cuál es la fase de entrenamiento, pero antes es necesario escoger el tipo de algoritmo a utilizar.

Algoritmos de clasificación

Los algoritmos de clasificación se encargan de buscar patrones en los datos proporcionados, con el fin de clasificarlos. Posterior a ello, comparan los nuevos datos y los ubican en los grupos correspondientes (Sandoval, 2018).

Entre los algoritmos de clasificación se encuentran:

Árbol de clasificación:

“Son clasificadores potentes, que utilizan una estructura de árbol para modelar la relación entre las características del modelo y los potenciales resultados” (Gago Utrera, 2017, p. 25).

En la siguiente **Tabla 3** se enumeran algunas fortalezas y debilidades con las que cuenta este algoritmo.

Tabla 3

Fortalezas y debilidades del algoritmo árbol de clasificación

Fortalezas	Debilidades
Simple de entender e interpretar.	Se puede generar árboles muy complejos.
No se necesitan preparar muchos datos.	Pequeñas variaciones en los datos producen un cambio considerable en la estructura generada.
Se puede manejar datos numéricos y categóricos.	El problema de encontrar el árbol óptimo está clasificado como NP-completo, por lo que se usan métodos heurísticos
El coste de utilizar el árbol para realizar predicciones es logarítmico.	

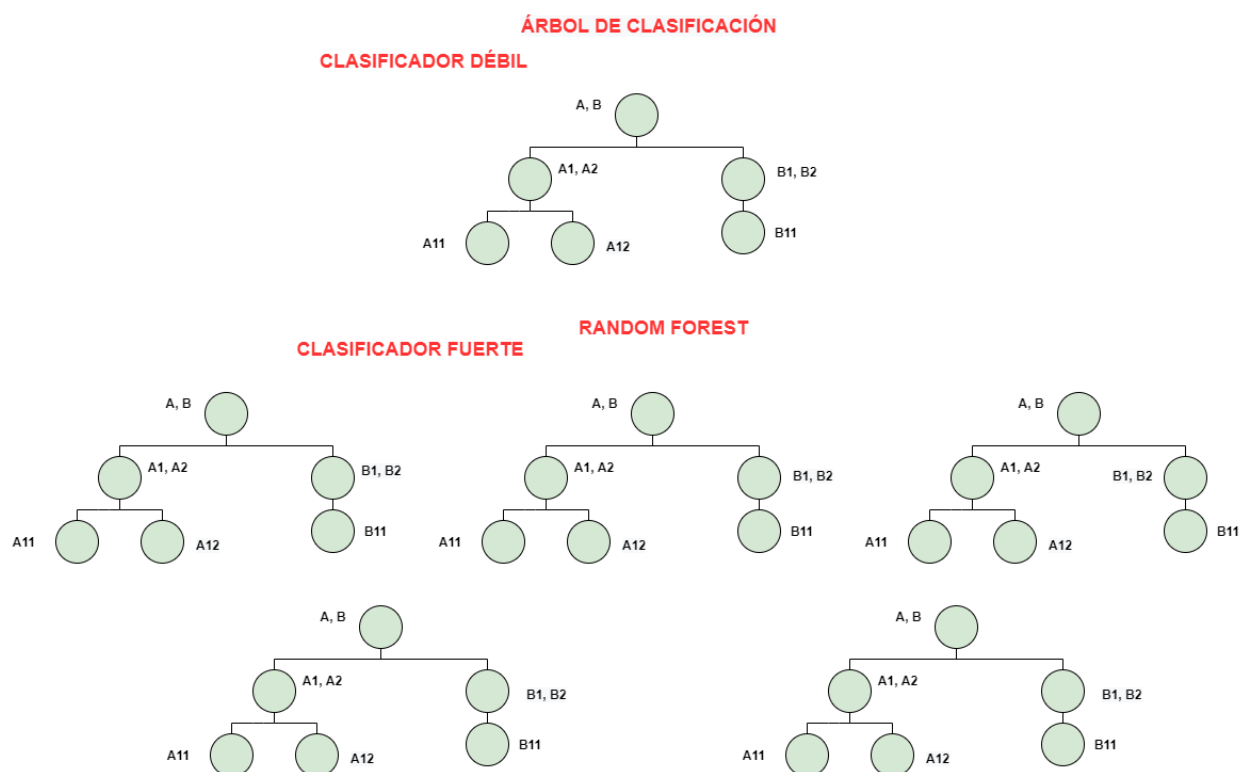
Nota: Detalla las debilidades y fortalezas del algoritmo. Tomada de (Gago Utrera, 2017).

Random Forest:

Es una combinación de árboles predictivos. Este algoritmo trabaja con un conjunto de árboles no correlacionados y los promedia (Hastie et al., citado en Medina Medrano & Ñique Chacón , 2017). “En el cual se tiene que cada árbol depende de los valores de un vector aleatorio de la muestra de manera independiente y con la misma distribución de todos los árboles en el bosque” (Medina Medrano & Ñique Chacón , 2017, p. 170). En **la Figura 7** se muestra la estructura del algoritmo de árbol clasificación y el algoritmo Random Forest.

Figura 7

Árbol de clasificación vs Random Forest



Nota: Estructura de un Árbol de clasificación y Random Forest. Adaptada de: (Medina Medrano & Ñique Chacón , 2017)

Gradient boosting (GB):

El algoritmo Gradiente Boosting está conformado por un conjunto de árboles de clasificación individuales. Estos árboles se entrenan de manera secuencial, por lo que los nuevos árboles tratan de mejorar los errores de los árboles anteriores (Amat Rodrigo, 2020). “La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo” (Amat Rodrigo, 2020). En la **Tabla 4** se muestran las fortalezas y debilidades del algoritmo

Tabla 4

Fortalezas y debilidades del algoritmo Gradient boosting

Fortalezas	Debilidades
Puede aplicarse para problemas de regresión y clasificación	Al combinar múltiples árboles, se pierde la interpretabilidad que tienen los modelos basados en un único árbol.
A comparación de otros algoritmos no necesita de mucha limpieza y preprocesamiento de datos	Cuando tratan con predictores continuos, pierden parte de su información al categorizarlas en el momento de la división de los nodos
Identifica de forma rápida las características más importantes	No son capaces de extrapolar fuera del rango de los predictores observados en los datos de entrenamiento.
Puede aplicarse a conjuntos de datos con un elevado número de observaciones	

Nota: Detalla las debilidades y fortalezas del algoritmo. Tomada de (Amat Rodrigo, 2020).

AdaBoost:

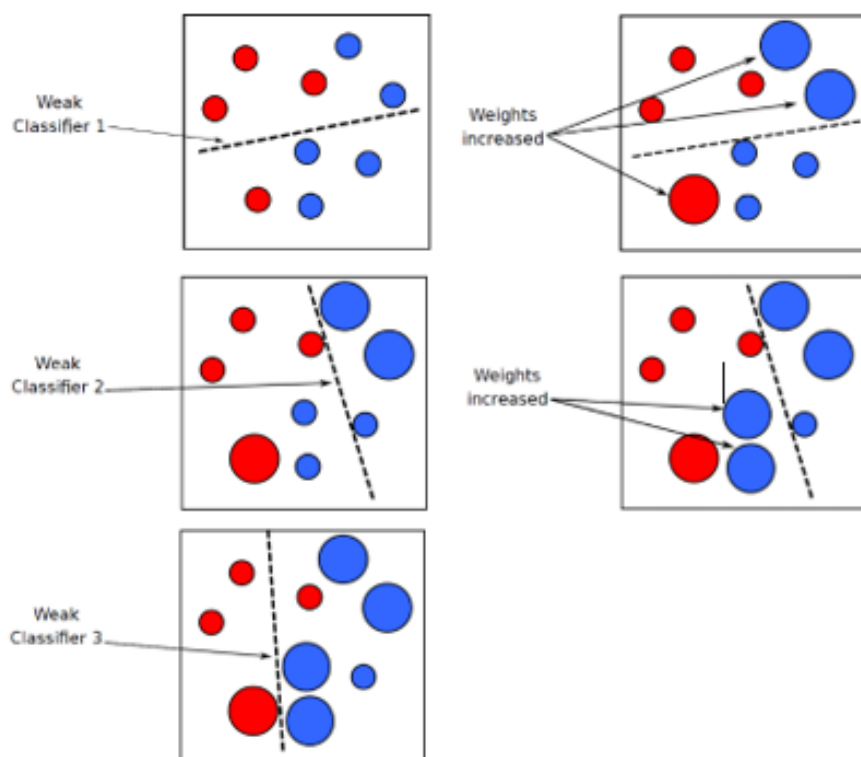
El meta-algoritmo AdaBoost tiene como objetivo combinar distintos clasificadores débiles ($h_i(x)$) a la vez para mejorar el rendimiento de la clasificación, donde $h_i(x)$ es un solo clasificador. Cada clasificador débil se capacita mediante un conjunto de muestras de capacitación. A su vez cada muestra tiene un peso y los pesos de todas las muestras son ajustados iterativamente.

AdaBoost entrena iterativamente a los clasificadores débiles y calcula un peso para cada uno y este peso representa la robustez de los clasificadores (Tharwat, 2018).

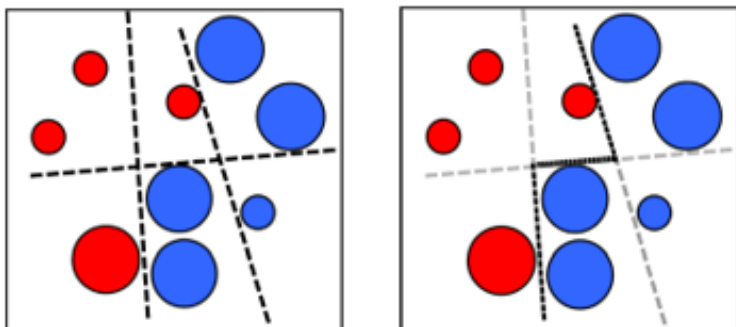
En la **Figura 8**, se muestra los clasificadores débiles y sus respectivos pesos aumentados y en la **Figura 9**, se presentan la clasificación final del algoritmo, la cual es una combinación de todos los clasificadores débiles.

Figura 8

Clasificadores débiles y sus pesos aumentados



Nota: Tomada de (Tharwat, 2018).

Figura 9*Clasificador final*

Nota: Tomada de (Tharwat, 2018).

En el algoritmo AdaBoost, existen las siguientes reglas:

- AdaBoost debe tener algunos de los estudiantes débiles, por ejemplo, árboles de decisión o clasificadores de redes neuronales (Tharwat, 2018).
- El error de cada alumno débil debe ser mejor que un error aleatorio clasificador, es decir, la tasa de error debe ser inferior al 50% (Tharwat, 2018).

AdaBoost tiene los siguientes pasos:

1. Paso de muestreo: En este paso, se seleccionan algunas muestras (D_t) del conjunto de entrenamiento, donde (D_t) es el conjunto de muestras en la iteración T (Tharwat, 2018).
2. Paso de entrenamiento: En este paso, se entrenan diferentes clasificadores usando (D_t) y se calculan las tasas de error (i) para cada clasificador (Tharwat, 2018).
3. Paso de combinación: Aquí se combinan todos los modelos entrenados (Tharwat, 2018).

Posterior a la selección de los algoritmos clasificadores se procede con el entrenamiento del modelo.

Entrenamiento

En esta fase se necesita del conjunto de datos, es por ello por lo que una parte de estos se divide en el conjunto de entrenamiento. Este va a permitir entrenar al modelo y brindarle la información para que encuentre patrones y poder así realizar futuras predicciones (Sandoval, 2018). Pero en este punto es importante destacar un problema que trae consigo el entrenamiento, A este problema se lo conoce como:

Overfitting y undertitting

Para aclarar lo anterior, en su artículo Ying (2019) indica que en el aprendizaje supervisado un modelo que no es capaz de generalizar los datos observados (datos de entrenamiento) de los datos que no han sido observados (datos de prueba) ocasiona un inconveniente que comúnmente recibe el nombre de sobreajuste u “overfitting”. Debido a la existencia del sobreajuste el modelo funciona correctamente en el set de entrenamiento, mientras que en el set de prueba el modelo tiene dificultades para adaptarse a la información de dicho conjunto.

Los modelos sobreajustados tienden a memorizar todos los datos, incluido el ruido inevitable en el conjunto de entrenamiento, en lugar de aprender la disciplina que se esconde detrás de los datos (Ying, 2019).

Las causas de este fenómeno se clasifican en 3 tipos:

- Aprendizaje por ruido en el conjunto de entrenamiento: Cuando el conjunto de datos es muy pequeño, o tiene datos menos representativos o demasiados ruidosos (Ying, 2019).
- Complejidad de la hipótesis: Cuando los algoritmos tienen demasiadas hipótesis (demasiadas entradas). El modelo se vuelve más preciso en promedio con menor

consistencia. Puede haber una diferencia drástica en diferentes conjuntos de datos (Ying, 2019).

- Procedimientos de comparaciones múltiples que son omnipresentes en los algoritmos de inducción, así como otros algoritmos de inteligencia artificial (Ying, 2019).

Por otra parte, el subajuste o “underfitting” es lo contrario al sobreajuste. Esto significa que el entrenamiento es insuficiente y la precisión del aprendizaje es baja (Zhang, Zhang, & Jiang, 2019).

Para solucionar el problema con respecto al sobreajuste, Ying (2019) indica 4 métodos:

Parada anticipada

Es una estrategia que se utiliza para evitar el fenómeno de “ralentización de la velocidad de aprendizaje”. Este problema significa que la precisión de los algoritmos deja de mejorar de algún punto, o incluso llega a empeorar debido al aprendizaje por ruido (Ying, 2019).

Si el modelo continúa aprendiendo después del punto, el error de validación aumentará mientras que el error de entrenamiento continuará disminuyendo. Si dejamos que nuestro modelo aprenda antes del punto, es insuficiente. Si lo detenemos después del punto, sobreajustamos el modelo. El objetivo es encontrar el punto exacto para dejar de entrenar (Ying, 2019).

Para encontrar el punto exacto, se debe realizar un seguimiento en la precisión de los datos de prueba mientras la red se entrena. Es decir, se calcula la precisión al final de cada época y se deja de entrenar cuando la precisión de los datos de prueba deje de mejorar (Ying, 2019).

Reducción de la red

La reducción de ruido se convierte en una alternativa de investigación para verificar la inhibición. En base a esto, se propone la poda para educar el tamaño de los clasificadores finales en el aprendizaje relacional, especialmente en el aprendizaje del árbol de decisión (Ying, 2019).

La poda consiste en reducir la complejidad de la clasificación eliminando datos irrelevantes y finalmente, evitar el sobreajuste y mejorar la precisión de la clasificación (Ying, 2019).

Existen dos enfoques para tratar el ruido: Prepoda y Postpoda.

Prepoda

Los algoritmos de prepoda funcionan durante el proceso de aprendizaje, utilizan criterios de detección para determinar cuándo agregar condiciones a una regla o agregar reglas a la descripción del modelo (Ying, 2019).

Postpoda

Divide el conjunto de entrenamiento en dos subconjuntos: conjunto de crecimiento y conjunto de poda. Los algoritmos postpoda ignoran los problemas de sobreajuste durante el proceso de aprendizaje en el conjunto de cultivo. Ellos ayudan a evitar el sobreajuste mediante la eliminación de condiciones y reglas del modelo generado durante el aprendizaje (Ying, 2019).

Expansión de los datos de entrenamiento

En el aprendizaje automático, el algoritmo no es la única clave que afecta la precisión de la clasificación final.

Su desempeño puede verse afectado significativamente por la cantidad y calidad del conjunto de datos de entrenamiento en muchos casos, especialmente en el área del aprendizaje supervisado (Ying, 2019).

El entrenamiento de modelos es en realidad un proceso de ajuste de sus hiperparámetros. Los parámetros bien ajustados logran un buen equilibrio entre la precisión y la regularidad del

entrenamiento y luego inhiben el efecto del sobreajuste, así como el del mal ajuste. Para ajustar estos parámetros, el modelo necesita muestras suficientes para el aprendizaje. Es decir, un conjunto de datos amplio puede mejorar la precisión de la predicción, especialmente si el modelo es complejo (Ying, 2019).

Regularización

Generalmente, la salida de un modelo puede verse afectada por múltiples características. Cuando la cantidad de funciones aumenta, el modelo se vuelve complicado. Un modelo de sobreajuste tiende a incorporar todas las características consideradas, aunque algunos de ellos tienen un efecto muy limitado en el resultado final. O peor aún, algunos de ellos son ruidos que no tienen sentido para la salida (Ying, 2019).

Para limitar estos casos, tenemos dos tipos de soluciones:

1. Seleccionar solo las funciones útiles y eliminar las funciones inútiles de nuestro modelo (Ying, 2019).
2. Minimizar los pesos de las características que tienen poca influencia en la clasificación final (Ying, 2019).

En otras palabras, se debe limitar el efecto de esas características inútiles, aunque no siempre se tiene conocimiento de cuáles son las funciones de menor significancia, por lo que se intenta limitar todas minimizando la función de costo del modelo (Ying, 2019).

Considerando los procedimientos que se deben realizar para un buen entrenamiento, el siguiente paso es la evaluación del modelo.

Evaluación del modelo

En esta etapa el modelo ya ha sido entrenado. Posterior a ello, será evaluado con nuevos datos. La tarea será clasificar estos nuevos datos y por medio de este proceso conocer que tan

efectivo es el modelo. Para ello se utilizan algunas métricas que permiten conocer su precisión y desempeño.

Urcuqui López (2016) menciona que, existen cuatro tipos de medidas (Falsos positivos (FP), Falsos negativos (FN), Verdaderos positivos (TP) y Verdaderos negativos (TN)). Estas conforman la matriz de confusión y se aplica a los problemas de clasificación. Su objetivo es calcular la precisión y evaluación del modelo. En la **Tabla 5** se presenta la matriz de confusión.

Tabla 5

Matriz de Confusión

Valor Predicción	Fraude Legítimo	<i>Valor Real</i>	
		<i>Fraude</i> TP FN	<i>Legítimo</i> FP TN

Nota: Matriz de confusión. Adaptada de (Urcuqui López, 2016).

Esta matriz contiene términos que serán detallados a continuación:

Verdaderos positivos: “Resultados positivos que el modelo ha detectado como positivos” (Quevedo Muñoz, 2020, p. 30).

Verdaderos negativos: “Resultados negativos que el modelo ha detectado como negativos” (Quevedo Muñoz, 2020, p. 30).

Falsos positivos: “Resultados negativos que el modelo ha detectado como positivos” (Quevedo Muñoz, 2020, p. 30).

Falsos negativos: “Resultados positivos que el modelo ha detectado como negativo” (Quevedo Muñoz, 2020, p. 30).

Asimismo, existen otras métricas que permiten evaluar el desempeño del modelo. A saber:
1) Precisión, 2) Sensibilidad/Exhaustividad (Recall) 3) Exactitud 4) Especificidad. 5) F1-score.

Precisión: “Mide el número de clasificación positivas correctas realizadas” (Maseda Tarin, 2019, P. 16). Se debe mencionar que esta métrica se encuentra en la librería Scikit-learn como *precision_score ()* y los parámetros que recibe este método son los datos reales de prueba y los datos obtenidos del modelo clasificador.

Exhaustividad/ Sensibilidad (Recall): “Mide la proporción del número de clasificaciones positivas identificadas correctamente” (Maseda Tarin, 2019, P. 16). En otras palabras, esta métrica encuentra todas las observaciones positivas (Zamorano Ruiz, 2018). Se encuentra en la librería Scikit-learn como *recall_score ()*.y los parámetros que recibe este método son los datos reales de prueba y los datos predichos por el modelo clasificador.

Exactitud: “Mide las predicciones que el modelo ha calificado correctamente” (Maseda Tarin, 2019, P. 16). El resultado de esta métrica es proporcionar que tan efectivo es el modelo clasificador. Se encuentra en la librería Scikit-learn como *acurracy_score ()* y recibe como parámetros los datos reales de prueba y los datos predichos por el modelo clasificador.

Especificidad: Esta métrica determina las predicciones negativas correctas que el modelo de predicción ha acertado con base al número total de condiciones negativas reales (Kok, Azween, & Jhanjhi, 2020).

F1-score: “Es la combinación de las métricas de precisión y exhaustividad y sirve de compromiso entre ellas” (Balparda, 2020, p. 52). Es decir, que se combinan las dos métricas para conocer su rendimiento. Se encuentra en la librería Scikit-learn como *f1_score ()* y recibe como parámetros los datos reales de prueba y los datos predichos por el modelo clasificador.

Adicionalmente, se debe considerar la diferencia que existe entre precisión y exactitud. “La exactitud es la cercanía de una medida al valor real, mientras que la precisión es el grado de cercanía de los valores de varias medidas en un punto” (Zita, s.f).

En la **Tabla 6** se presentan las métricas mencionadas anteriormente.

Tabla 6

Medidas de Desempeño

Nombre	Fórmula
Precisión	$\frac{TP}{TP + FP}$
Sensibilidad (Recall)	$\frac{TP}{TP + FN}$
Exactitud	$\frac{TP + TN}{TP + FP + TN + FN}$
F1-score	$2 * \frac{TP + TN}{TP + FP + TN + FN}$
Especificidad	$\frac{TN}{TN + FP}$

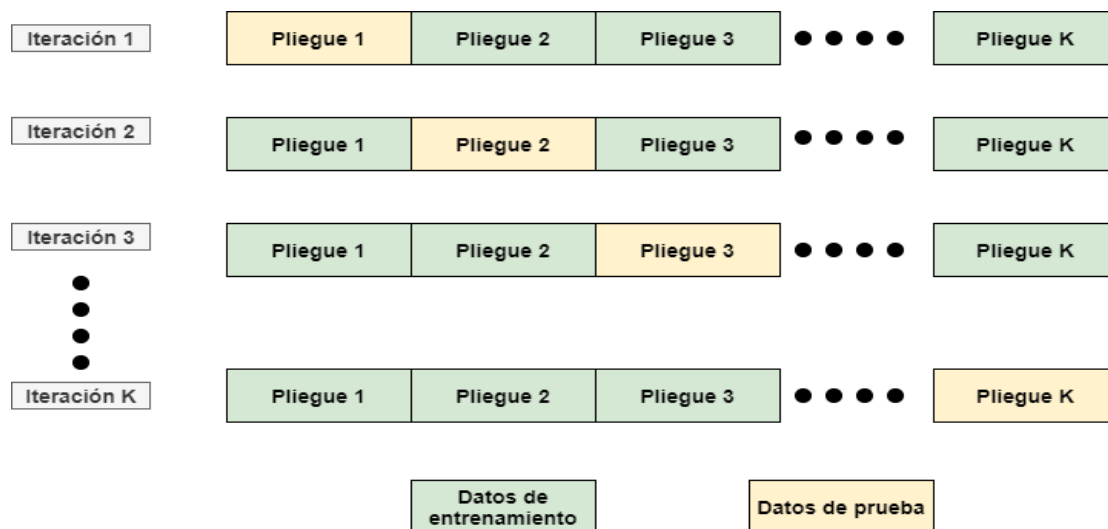
Nota: Métricas para evaluar el desempeño de un modelo. Adaptada de (Maseda Tarin, 2019).

Adicionalmente, existen técnicas honestas de medición del rendimiento del modelo. Una de ellas se denomina k-fold cross-validation.

k-fold cross-validation (validación cruzada k-fold)

Esta técnica separa de manera aleatoria el conjunto de datos en k secciones o pliegues de tamaño aproximadamente iguales. Para entrenar al modelo clasificador, se utiliza los k-1 secciones y posteriormente se evalúa la precisión del modelo logrado en la sección restante. Este procedimiento se repite k veces (Parrales Bravo, 2020).

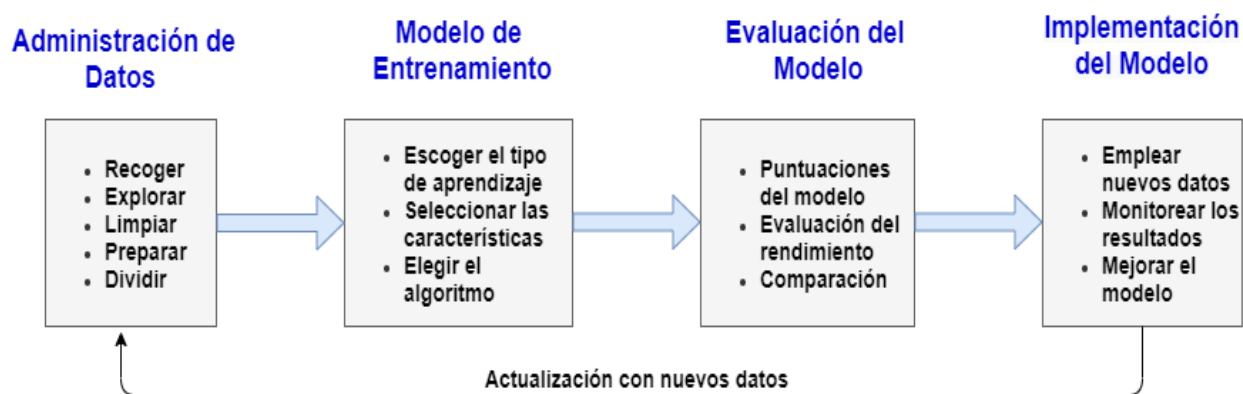
En la **Figura 10** se muestra un gráfico de dicha técnica.

Figura 10*Cross Fold Validation*

Nota: Cross Fold Validation. Adaptada de (Qiubing , Mingchao , & Shuai , 2019).

Otro aspecto importante que considerar es el ciclo de modelado del aprendizaje automático.

En la **Figura 11** se detalla el ciclo de modelado del aprendizaje automático. Este ciclo se aplica tanto para el aprendizaje supervisado y no supervisado.

Figura 11*Las 4 Etapas del Ciclo de Modelado del Aprendizaje Automático*

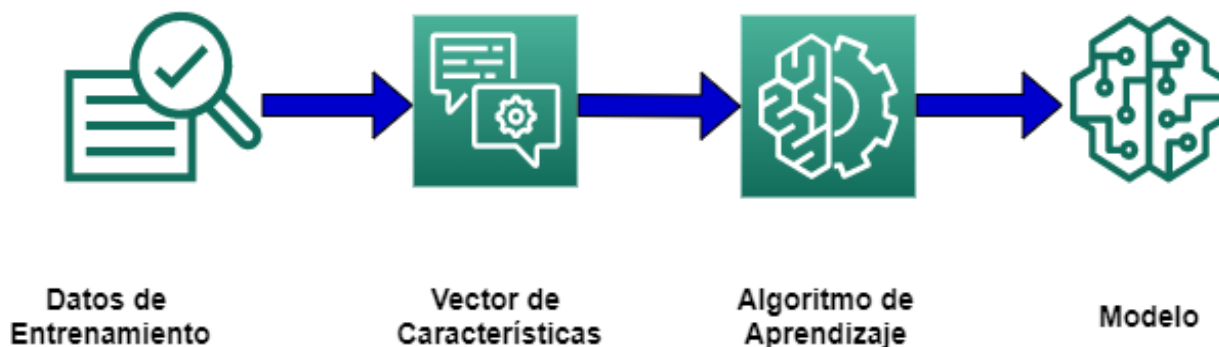
Nota: Las 4 Etapas del Ciclo de Modelado del Aprendizaje Automático Adaptada de (Simba Amores & Espinoza Padilla, 2019).

El ciclo de modelado inicia en la administración de los datos. En esta parte la información es recopilada para su respectivo uso. Posterior a ello, el conjunto de datos es explorado permitiendo así conocer la estructura y significado del mismo. Estos datos deben ser limpiados, lo que implica convertirlos en construcciones matemáticas que el modelo pueda entender. Terminado el proceso de limpieza de datos, estos son cargados en un entorno de programación y se dividen en subconjuntos de entrenamiento y prueba (Simba Amores & Espinoza Padilla, 2019).

En segunda instancia el ciclo avanza a la etapa de modelo de entrenamiento. Aquí se procede a elegir el tipo de aprendizaje a utilizar para la creación del modelo, ya sea por predicción, agrupación, etc. Una vez elegido el tipo de aprendizaje a utilizar, se deben estudiar las características pertenecientes al conjunto de datos de entrenamiento. Posterior a ello, se elige el algoritmo con el cual se entrenará el modelo (Simba Amores & Espinoza Padilla, 2019). En La **Figura 12** se muestra el flujo de aprendizaje y se detalla el proceso.

Figura 12

Flujo de Aprendizaje



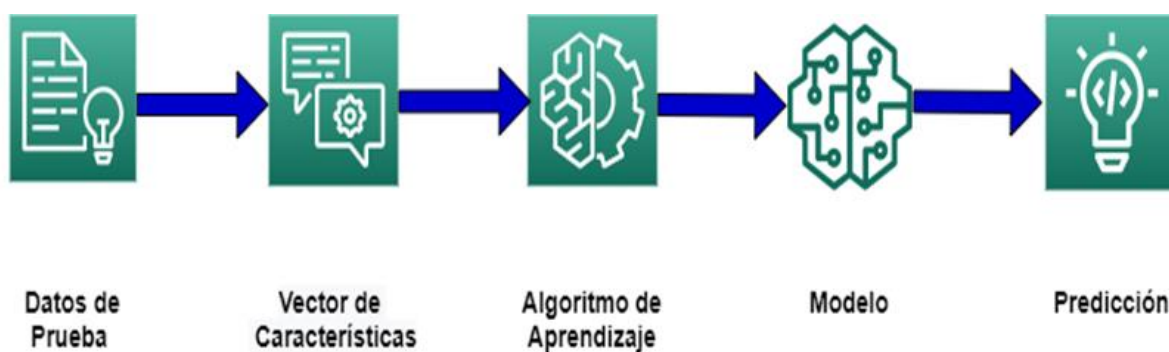
Nota: Flujo de aprendizaje del modelo. Adaptada de (Simba Amores & Espinoza Padilla, 2019).

El flujo de aprendizaje empieza con el entrenamiento del conjunto de datos, luego de este paso, los datos se transforman en una representación de vectores. En la penúltima instancia del proceso la lista de vectores se procesa con un algoritmo de aprendizaje y se obtiene un nuevo modelo con los parámetros de dicho algoritmo (Simba Amores & Espinoza Padilla, 2019).

Por otro lado, en la etapa de evaluación del modelo, este se somete a una evaluación con distintos algoritmos, con el objetivo de encontrar al que proporcione la mejor precisión con base a los parámetros de cada algoritmo. Para conocer cuál de los modelos generó la mayor precisión en algunos casos se utiliza la matriz de confusión (Simba Amores & Espinoza Padilla, 2019). En la **Figura 13** se muestra el flujo de evaluación del modelo.

Figura 13

Fase de evaluación del modelo



Nota: Fase de evaluación del modelo Adaptada de (Sukla, citado en Simba Amores & Espinoza Padilla, 2019).

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. (Simba Amores & Espinoza Padilla, 2019, p. 20)

Revisiones sistemáticas

La revisión sistemática de este trabajo se basó en cómo puede ayudar un modelo de detección de intrusos basado en aprendizaje automático en la seguridad de una red. En apartados anteriores se mencionó la problemática que existe en la actualidad con la seguridad informática, por lo que lograr mitigar los ciberataques puede ser de ayuda para las empresas. Se revisó que tan eficiente es el aprendizaje automático para detectar posibles ciberataques. Además, se incluyó trabajos sobre la selección de características en la fase de preprocesamiento de datos.

De este análisis se deriva una tabla de información con los artículos científicos, tesis más destacadas y pertinentes en el tema de estudio.

Búsqueda

Se realizó una búsqueda de trabajos científicos o trabajos de titulación acerca de modelos de detección de intrusos basados en aprendizaje automático, alojados en repositorios de distintas universidades y en revistas científicas. Luego de la exploración de documentos, se adquirió información de los trabajos previamente seleccionados. Se revisó parte conceptual, problemáticas, procesos e ideas.

Criterios

Se realizó una búsqueda seleccionando áreas, tales como: seguridad informática, e inteligencia artificial. De igual manera, se realizó búsquedas mediante palabras claves como: ciberseguridad, aprendizaje automático, detección de intrusos, y selección de características.

Resultados

Se revisaron 14 documentos entre trabajos de tesis y artículos científicos, de los cuales se descartaron 2 debido a la fecha de publicación (2009 y 2014). De los 12 documentos restantes se descartaron 6 debido a que los modelos tenían un enfoque diferente a la detección de software

malicioso. De los 6 trabajos restantes, todos ellos aportaron información importante, pero se eligieron 3. Estos artículos fueron considerados debido a que coinciden con lo propuesta realizada y en ciertos procesos se asemejaron al presente trabajo. Adicionalmente, se revisaron 8 trabajos relacionados con la selección de características, de los cuales se consideraron 3.

Meta-análisis

A continuación, los detalles de los trabajos revisados se muestran en la **Tabla 7**.

Tabla 7

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
<ul style="list-style-type: none"> • Jorge Luis Rivero Pérez • Bernardete Ribeiro • Kadir Héctor Ortiz 	Comparación de algoritmos para detección de intrusos en entornos estacionarios y de flujo de datos	Revista Universidad y Sociedad	Diciembre 2016	<ul style="list-style-type: none"> • Aprendizaje automático • detección de intrusos en redes • flujos de datos • KDD99
Johan Mardini	Modelo de detección de intrusiones en sistemas computacionales, realizando selección de características con Chi Square, entrenamiento y clasificación con Ghsom	Investigación e Innovación en Ingenierías	18 de julio de 2017	<ul style="list-style-type: none"> • DATASET KDDNSL DARPA • IDS (sistema de detección de intrusiones) • GHSOM (mapas auto organizativos jerárquicos) Reconocimie

				nto de patrones
Rolando Salazar Hernández	Sistemas de detección intrusos mediante modelado de URI	de Universidad de Granada	de 02 de febrero del 2016	

Nota: Elaborado por: Dayannara Avila y Joel Torres con los datos de la investigación

Tabla 8

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
Jorge Luis Rivero Pérez	Técnicas de aprendizaje automático para la detección de intrusos en redes de computadora	Revista Cubana de Ciencias Informáticas	1 de diciembre de 2014	<ul style="list-style-type: none"> • Aprendizaje automático. • Detección de intrusos, Inteligencia de enjambre • KDD cup 99.
Fernando Javier Villar Freire	Detección de anomalías de red mediante técnicas de machine learning	Universidade Da Coruña	noviembre 2019	<ul style="list-style-type: none"> • Machine learning • Clasificación • Ingeniería de características • Regresión logística, • Random forest • Svm • Anomalía • Ids • Computación distribuida • Flujo

José Manuel Rodríguez Rama	Aplicación de técnicas de Machine Learning a la detección de ataque	Universitat Oberta de Catalunya	4 de junio del 2018	<ul style="list-style-type: none"> Machine Learning Scikit-Learn
Andrés Valencia Peral	Técnica de aprendizaje automático para la detección de ataques en el tráfico de red	Universitat Oberta de Catalunya	04 de junio de 2019	<ul style="list-style-type: none"> Machine Learning, Intrusion Detection System, Deep

Nota: Las filas con datos de color azul fueron los trabajos considerados. Elaborado por: Dayannara Avila y Joel Torres.

Tabla 9

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
<ul style="list-style-type: none"> Claudia Ximena Sanna Morales Sebastian Alberto Londoño Castaño 	Modelo de detección de intrusos usando técnicas de aprendizaje de máquina	Tecnológico de Antioquia	13 de diciembre del 2018	
Carlos Jiménez Galindo	Diseño y Optimización de un Sistema de Detección de Intrusos Híbrido	Universidad de Almería	2009	
Jonathan Nabor López Espinosa	Uso de técnicas de machine learning para la detección de fraudes en	OLACEFS	2019	

		los contratos de obras públicas		
César Guevara Maldonado	Byron	Desarrollo de algoritmos eficientes para identificación de usuarios en accesos informáticos	Universidad Complutense De Madrid	2018

Nota: Elaborado por: Dayannara Avila y Joel Torres con los datos de la investigación.

Tabla 10

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
José María Dueñas Quesada	Aprendizaje supervisado para la detección de amenazas web mediante clasificación basada en árboles de decisión	Universitat Oberta de Catalunya	2 de junio de 2020	<ul style="list-style-type: none"> • Aprendizaje automático • Machine Learning • Seguridad informática • Ciberseguridad • Análisis de datos • Aprendizaje Supervisado • Árboles de decisión • Python • Scikit-learn
<ul style="list-style-type: none"> • Mehmood Tahir • Helmi B Md 	Machine learning algorithms in context of intrusion detection	IEEE Xplore	2016	<ul style="list-style-type: none"> • Intrusion detection system • Machine learning • Network-based

				intrusion detection system <ul style="list-style-type: none"> • Anomaly detection • Supervised algorithms
Urcuqui López Christian	Módulo de machine learning para detección de malware en Android	Universidad Ices	2016	<ul style="list-style-type: none"> • Aprendizaje automático • Seguridad • Teléfono inteligente • Google • Software malicioso • Inteligencia artificial

Nota: Las filas con datos de color azul fueron los trabajos considerados. Elaborado por: Dayannara Avila y Joel Torres.

Tabla 11

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
<ul style="list-style-type: none"> • Mahsa Bataghva Shahbaz • Xianbin Wang • Aydin Behnad • Jagath Samarabandu 	On efficiency enhancement of the correlation-based feature selection for intrusion detection systems	IEEE Xplore	2016	<ul style="list-style-type: none"> • Correlation-based Feature Selection (CFS), feature selection • Intrusion Detection System (IDS) • NSL-KDD • Symmetric Uncertainty (SU).

Nota: Las filas con datos de color azul fueron los trabajos considerados. Elaborado por: Dayannara Avila y Joel Torres.

Tabla 12*Trabajos revisados*

AUTORES	RESUMEN	UNIVERSIDAD/ REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
<ul style="list-style-type: none"> • Zahra Mungloo Dilmohamud • Gary Marigliano • Yasmina Jaufeerally Fakim • Carlos Peña Reyes 	<p>A Comparative Study of Feature Selection Methods for Biomarker Discovery</p>	IEEE Xplore	2018	<ul style="list-style-type: none"> • Feature selection • Robustness • Machine learning • Gene expression data • Biomarker discovery
<ul style="list-style-type: none"> • K.R.Pushpalatha • Asha Gowda Karegowda 	<p>CFS Based Feature Subset Selection for Enhancing Classification of Similar Looking Food Grains- A Filter Approach</p>	IEEE Xplore	2017	<ul style="list-style-type: none"> • Feature selection • Filter approach • CFS • Classifiers • Similar looking food grains, CGLCM
DU Shao-bo	<p>Intrusion Feature Selection Method based on Neighborhood Distance</p>	IEEE Xplore	2017	<ul style="list-style-type: none"> • Feature selection • Neighborhood distance • Attribute significance • Detection rate
<ul style="list-style-type: none"> • C. Kalimuthan, • J. Arokia Renjit, 	<p>Review on intrusion detection using feature selection with</p>	Elsevier	2020	<ul style="list-style-type: none"> • Intrusion detection systems • Machine learning • Deep learning

machine learning techniques	<ul style="list-style-type: none"> • Feature selection • Classifier • Network computing security • Benchmark dataset
-----------------------------	--

Nota: Las filas con datos de color azul fueron los trabajos considerados. Elaborado por: Dayannara Avila y Joel Torres.

Tabla 13

Trabajos revisados

AUTORES	RESUMEN	UNIVERSIDAD / REVISTA	FECHA PUBLICACIÓN	PALABRAS CLAVES
<ul style="list-style-type: none"> • Bachu Venkatesh • J. Anuradha 	A Review of Feature Selection and Its Methods	Cybernetics and Information Technologies	19 de marzo de 2020	<ul style="list-style-type: none"> • Dimensionality Reduction (DR) • Feature Selection (FS) • Feature Extraction (FE)
<ul style="list-style-type: none"> • Jie Cai • Jiawei Luo • Shulin Wang • Sheng Yang 	Feature selection in machine learning: A new perspective	Neurocomputing	2018	<ul style="list-style-type: none"> • Feature selection • Dimensionality reduction • Machine learning • Data mining
<ul style="list-style-type: none"> • Verónica Bolón Canedo • Amparo Alonso Betanzos 	Ensembles for feature selection: A review and future trends	Information Fusion	2019	<ul style="list-style-type: none"> • Ensemble learning • Feature selection

Nota: Las filas con datos de color azul fueron los trabajos considerados. Elaborado por: Dayannara Avila y Joel Torres.

Los trabajos consultados muestran la ayuda que brinda el aprendizaje automático a la seguridad informática, específicamente a la detección de intrusos. Dan a conocer que los sistemas

de detección de intrusos ordinarios realizan su trabajo de detección basándose en firmas y se conoce que esta metodología no es eficiente, debido a que existen sistemas polimórficos que pueden eludir fácilmente esta metodología de detección.

“El uso de herramientas cuyo objetivo es la detección de diversos tipos de amenazas se basan cada vez más en sistemas de aprendizaje automático” (Dueñas Quesada, 2020, p. 6).

A continuación, se analizó a partir de los resultados de dos de los trabajos considerados, la precisión de los modelos creados para clasificar ataques y cuál de ellos fue el mejor.

En el trabajo de Rodríguez Rama (2018) se utilizó un conjunto de datos con 4.900.000 vectores de conexiones y cada uno con 41 atributos. Donde cada uno de ellos era etiquetado como “normal” o como un ataque determinado. El conjunto de datos cuenta con un total de 22 ataques, agrupados en 4 categorías: Dos, Probing, R2L y U2R. En la **Tabla 14**, se puede evidenciar los algoritmos que se utilizaron para este trabajo. Se observó cuál fue el tiempo de creación de modelo y el porcentaje de precisión.

Tabla 14

Porcentaje de precisión y tiempo de creación de los modelos

Nombre del algoritmo	Porcentaje de precisión	Tiempo de creación de modelo
Random Tree	99.9271 %	7.71 segundos
Random Forest	99.9609 %	387.39 segundos
C4.5	99.9541 %	64.68 segundos
Naïve Bayes	93.0165 %	5.1 segundos
SVM	99.9055 %	486.68 segundos
Regresión Logística	99.8907 %	6016.51 segundos

Nota: Tomada de (Rodríguez Rama, 2018).

Como se puede observar, el algoritmo de árboles (Random Tree) fue el que mejor resultados obtuvo.

En el trabajo de Mehmood y Rais (2016) se utilizó el conjunto de datos perteneciente al programa de evaluación DARPA' 98 ID (KDD99). Este conjunto de datos contiene 4 tipos de ataques y una clase denominada normal. Se aplicó aprendizaje supervisado, para comparar la tasa de verdaderos positivos, falsos negativos y precisión por parte de los algoritmos utilizados en este artículo.

La **Tabla 15**, detalla los algoritmos que se utilizaron para la clasificación de ataques con su respectivo porcentaje de precisión.

Tabla 15

Porcentaje de precisión de los algoritmos

Nombre del algoritmo	Porcentaje de precisión
Naïve Bayes	84 %
SVM	98 %
C4.5	99.5 %
Tabla de decisión	99.5 %

Nota: Adaptada de (Mehmood & Rais, 2016).

Mehmood y Rais (2016) concluyeron que, la precisión de C4.5 fue muy alta con respecto a los demás, e igualmente su tasa de clasificación errónea fue baja. Por otro lado, Naive Bayes fue el peor resultado de precisión.

Hipótesis / Preguntas científicas a contestarse

¿Qué impacto tendrá el proveer a las empresas ecuatorianas de un aplicativo gratuito que les permita detectar intrusos en su red?

¿Qué impacto tendrá la técnica de selección de características propuesta en el rendimiento de los modelos clasificadores?

Variables de la investigación

Se determinaron las variables dependientes e independientes de la investigación y son las siguientes:

Variable dependiente: Es el modelo clasificador entrenado que ayudará a realizar predicciones.

Variable independiente: Son los diferentes algoritmos de aprendizaje supervisado y métodos de preprocesamiento que se utilizan para construir el modelo clasificador. Adicionalmente, las métricas que servirán para evaluar la efectividad del modelo. A saber: 1) exactitud, 2) especificidad, 3) exhaustividad / sensibilidad (recall), 4) F1-score.

Definiciones conceptuales

Algoritmo:

“Conjunto ordenado y finito que permite hallar la solución de un problema” (Real Academia Española, s.f).

Aprendizaje automático:

Es un subcampo de las ciencias de la computación y una rama de la inteligencia artificial cuya finalidad es desarrollar técnicas que permitan a las computadoras aprender,

convirtiéndose en un pilar fundamental para el trato de datos a gran escala. (Ramírez Hinestroza, 2018, p. 2)

Anova:

Es una prueba de hipótesis, cuyo objetivo es encontrar la diferencia estadística entre los resultados de más de dos algoritmos. Esta prueba asegura si en los resultados la diferencia es significativa o se debe al azar (Binbusayyis & Vaiyapuri, 2020).

chi-cuadrado (Chi):

Función proporcionada por la librería scikit-learn. Mide la dependencia entre variables estocásticas y descarta las características que tienen la mayor probabilidad de ser independientes de la clase (scikit-learn, s.f).

Ciberataque:

“Cualquier práctica realizada por una persona u organización con la finalidad de infiltrar, atacar y ocasionar daños a los sistemas de información” (Freire López, 2017, p. 5).

Exactitud:

“Sirve para sacar el porcentaje de observaciones que ha clasificado correctamente, cuanto más cercano sea su valor a 1 mejor será”. (Zamorano Ruiz, 2018, p. 34).

Exhaustividad /Sensibilidad (Recall):

“Mide la habilidad del clasificador de encontrar todas las observaciones positivas, buscamos que este valor sea lo mayor en una escala del 0 al 100” (Zamorano Ruiz, 2018, p. 34).

f_classif:

Función proporcionada por la librería scikit-learn. Mediante el cual se calcula el valor de F Anova para los datos proporcionados (scikit-learn, s.f).

F1 -score:

“Es la combinación de las métricas de precisión y exhaustividad y sirve de compromiso entre ellas. La mejor puntuación F1 es igual a 1 y la peor a 0” (Balparda, 2020, p. 52).

Malware:

Programa informático que tiene efecto no deseado o malicioso, (...) busca en su mayoría robar información personal que puede ser utilizada por los atacantes para cometer fechorías. (Symantec, citado en Llamas Covarrubias & Llamas Covarrubias, 2018, p.378)

Matriz de confusión:

“Se utiliza para evaluar la precisión de un clasificador (TP, FN, FP, TN)” (Balparda, 2020, p. 52).

mutual_info_classif;

Función proporcionada por la librería scikit-learn. Proporciona una medida general de dependencia entre dos variables (Yinghua, Yongkang, & Liping, 2019). Indica si las variables comparten información entre sí.

Precisión:

“Mide la habilidad de un clasificador de no etiquetar como positivo una observación que se debe considerar como negativa. Cuanto mayor sea este valor mejor será” (Zamorano Ruiz, 2018, p. 34).

Selección hacia adelante (OLS):

Algoritmo que comienza con un subconjunto de características vacío, que secuencialmente irá introduciendo características relevantes, hasta llegar al número total de características que se haya indicado previamente (Maseda Tarin, 2019).

F1-score:

“Es la combinación de las métricas de precisión y exhaustividad y sirve de compromiso entre ellas. La mejor puntuación F1 es igual a 1 y la peor a 0” (Balparda, 2020, p. 52).

train_to_split:

Método perteneciente a la librería Scikit-learn. Este método permite dividir en matrices en subconjuntos de entrenamiento y prueba (scikit-learn, s.f).

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se detalla la metodología de investigación utilizada, la cual permitió el análisis de los datos y selección de características. Además, mediante esta metodología se explicó los algoritmos utilizados y el entrenamiento del modelo.

La revisión documental se utilizó para la elaboración de la parte teórica del proyecto. La cual permitió conocer los conceptos bases e información relevante del tema propuesto.

La revisión documental es aquella que se realiza a través de la consulta de documentos (libros, revistas, periódicos, etc.) (Universidad de Jaén, s.f).

Tipo de investigación

Para el presente trabajo de titulación se seleccionó el tipo de investigación experimental. La investigación experimental “es aquella donde se realiza la manipulación de una o varias variables no verificadas, en condiciones de riguroso control, con la finalidad de detallar el modo y las causas por las cuales se ha producido un determinado hecho o fenómeno” (Alan Neill & Cortez Suárez, 2018). Se escogió este tipo de investigación debido a que permite identificar un problema, desarrollar diferentes soluciones y probar cada una de ellas.

Diseño metodológico de la investigación

Para el presente trabajo de titulación se optó por utilizar la metodología experimental. En la sección llamada “Propuesta” se detallan los pasos a seguir para el desarrollo de esta.

Definición del problema central

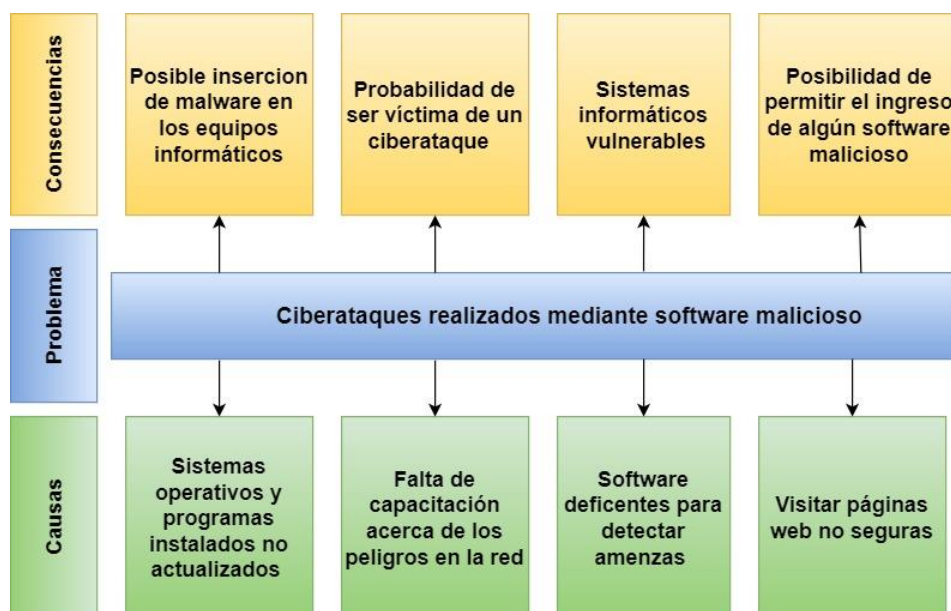
Uno de los problemas más comunes que surgen en las compañías son los ciberataques, los cuales pueden ocasionar problemas críticos en las empresas. Estos problemas o inconvenientes pueden mitigarse adoptando o implementando el aprendizaje automático en la seguridad informática de las empresas.

Análisis de problemas:

Los ciberataques que sufren las empresas dan como resultado el robo de información de esta y afectan el correcto funcionamiento de los equipos informáticos. Estos ataques se pueden dar por: 1) vulnerabilidades en los sistemas informáticos, 2) falta de capacitación de los usuarios, 3) bajo nivel de ciberseguridad y 4) protocolos frágiles que manejan las empresas. En la **Figura 14** se observa el árbol de problemas.

Figura 14

Árbol de Problemas



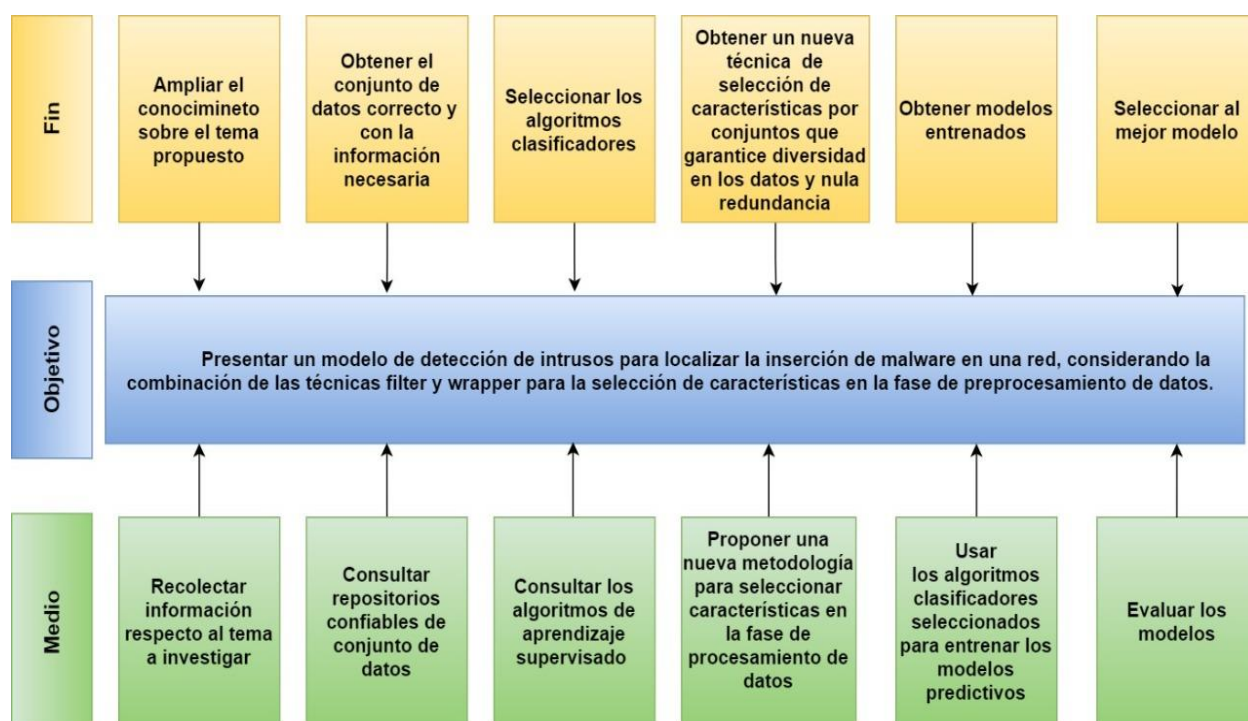
Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente datos de la investigación

Análisis de objetivos:

Una vez entendido las causas y consecuencias que tienen los ciberataques en las empresas, se ha optado por plantear medios que ayuden con la solución del problema. Los cuales se reflejan en la **Figura 15** mediante el árbol de objetivos.

Figura 15

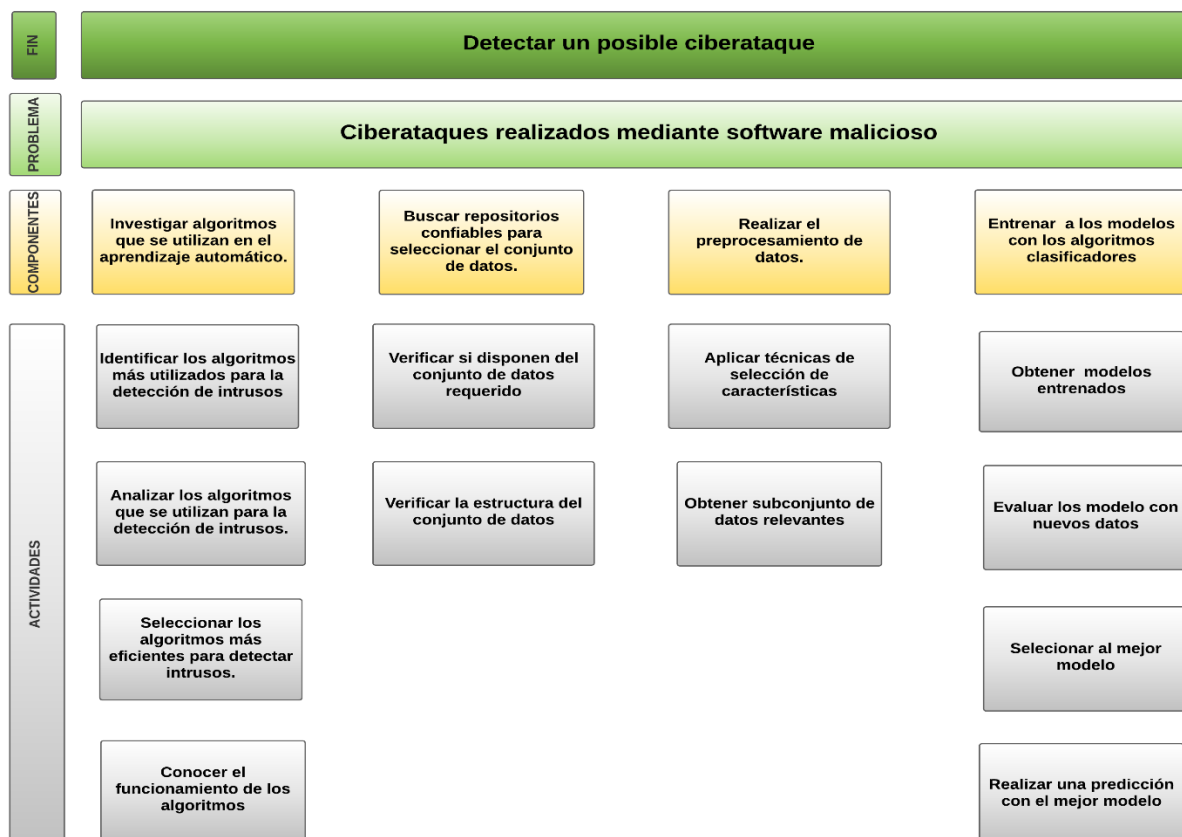
Árbol de Objetivos



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente datos de la investigación.

Diseño de estrategias:

En la **Figura 16**, se muestra la estructura analítica del proyecto, donde se expone cada una de las actividades de acuerdo con la alternativa seleccionada que llevarán a resolver la problemática del presente trabajo.

Figura 16*Estructura analítica del proyecto*

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente Datos de la investigación.

Entregables del proyecto

El entregable de este trabajo de investigación será un artículo científico, el cual explicará la metodología propuesta, cómo se usaron los algoritmos de clasificación y el entrenamiento del modelo. Este artículo científico se encuentra en el apartado de anexos, específicamente en el anexo 9. Además, se proporcionará el código, los resultados y las conclusiones que se obtuvieron de la investigación realizada.

Análisis de factibilidad

La innovación en el proceso de detección de intrusos va aumentando a medida que los días pasan. Utilizando técnicas de aprendizaje automático se puede lograr la obtención de un modelo que: 1) combine técnicas de selección de características, y 2) realice un entrenamiento con algoritmos de clasificación; todo ello con el propósito de predecir si un archivo es malicioso o legítimo.

El objetivo principal de este proyecto es realizar un modelo de detección de intrusos utilizando técnicas de aprendizaje automático, lo que ayudará a mejorar el proceso de detección de intrusos. Así pues, en esta sección se llevará a cabo un análisis para determinar si la investigación propuesta es factible, considerando para ello: 1) las herramientas de las que disponemos, 2) el tiempo permitido para el trabajo de titulación, y 3) el cumplimiento de cada uno de los objetivos planteados. Para ello, presentamos a continuación el respectivo análisis de factibilidad describiendo cada uno de los siguientes criterios:

- Factibilidad Operacional.
- Factibilidad Técnica.
- Factibilidad Legal.
- Factibilidad Económica.

Factibilidad operacional

El aplicativo desarrollado en el presente trabajo de investigación puede ser manejado y utilizado por el personal del área de sistemas de la empresa que, al menos, tenga conocimientos sobre: 1) lenguaje de programación Python, 2) manejo de consola de Windows, 3) instalación y actualización de programas. Si el usuario desea realizar modificaciones en alguna sección del

modelo presentado, deberá tener conocimientos avanzado acerca del aprendizaje automático y las librerías que se utilizan para efectuar dichas modificaciones.

Factibilidad técnica

Para desarrollar el modelo del presente trabajo de investigación, se hará uso del lenguaje de programación Python, utilizando las librerías pandas, numpy, pickle y sklearn. Este modelo propuesto en el presente trabajo cuenta con un algoritmo de selección de subconjunto de características que, mediante la combinación de técnicas filter y wrapper, busca que las características seleccionadas sean las más relevantes del conjunto de datos y que no exista redundancia entre ellas.

Requisitos de hardware

Para el desarrollo del modelo se utilizó un equipo de recursos medios, sus características se pueden evidenciar en la **Tabla 16**.

Tabla 16

Características del hardware

Características	Especificaciones
Procesador	Core i3 5th generación de 2.00 GHz
Memoria Ram	6 GB RAM
Disco duro	1 TB

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

Requisitos de software

La **Tabla 17** contiene las tecnologías utilizadas para la realización del proyecto.

Tabla 17

Requisitos de software

Tecnología	Versión
Lenguaje de programación	Python 3.8
Entorno de programación	Spyder 3

Librerías de programación	Numpy 1.19.2 Pandas 1.1.3 Pickle 0.7.5 Sklearn 0.23.2 Joblib 0.17.0
Sistema Operativo	Windows10 Pro, Kali Linux 2020

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

Factibilidad legal

El proyecto es desarrollado haciendo uso del software libre. Al respecto, la ley gubernamental decretada por Rafael Correa Delgado en el Decreto Ejecutivo 1014, emitido el 10 abril del 2008, recomienda el uso de software libre como política pública para las entidades de la Administración Pública Central para sus sistemas y equipamientos informáticos.

El sistema operativo Windows 10 que se utilizó para el desarrollo del proyecto fue adquirido y, por lo tanto, se cuenta con la respectiva licencia que proporciona la empresa Microsoft.

Por lo consiguiente, no se está infringiendo la ley de propiedad intelectual, la cual indica que no se puede usar total o parcial, software que haya sido patentado por alguien sin previo su consentimiento. Con todo ello, se puede concluir que el presente trabajo de titulación es factible a nivel legal.

Factibilidad económica

A continuación, se detalla los costos de los recursos a utilizarse en el desarrollo del presente proyecto. La **Tabla 18**. detalla el costo de inversión del proyecto.

Tabla 18*Resumen costo de inversión*

Tecnología	Versión
Recurso Humano	\$ 600,00
Hardware	\$ 600,00
Software	\$ 30,00
Total	\$ 1.230,00

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

El costo de inversión está dividido en 3 tipos de recursos. La **Tabla 19**, nos muestra los costos por recurso humano, en el cual se obtuvo un total de \$ 600,00 para cubrir dicho recurso.

Tabla 19*Costo por recurso humano*

Cargo	Costo	Cantidad	Total
Investigador1	\$ 300,00	1	\$ 300,00
Investigador2	\$ 300,00	1	\$ 300,00
Total			\$ 600,00

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

La **Tabla 20** nos muestra el detalle de los costos de hardware y software, sumando entre los dos \$ 630.00.

Tabla 20*Costo por recurso hardware y software*

Cargo	Costo	Cantidad	Total
Laptop DELL	\$ 600,00	1	\$ 600,00
Licencia Windows Pro	\$ 30,00	1	\$ 30,00
Total			\$ 630,00

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Propuesta

La propuesta de este proyecto es un modelo de detección de intrusos utilizando técnicas de aprendizaje automático. Para llevar a cabo el desarrollo y prueba del modelo se hizo uso del conjunto de datos que se encuentra alojado en el repositorio digital GitHub⁷. Dicho conjunto de datos está conformado por características tanto de archivos maliciosos como archivos legítimos. En la **Tabla 21** se pueden observar las 57 características del conjunto de datos correspondientes a los archivos maliciosos y legítimos. Cabe mencionar, que estas características están basadas en los parámetros de los archivos ejecutable portátil (PE), formato utilizado por ejecutables del entorno Windows.

Tabla 21*Características del conjunto de datos*

Características			
Name	md5	Machine	SizeOfOptionalHeader
Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode
SizeOfInitializedData	SizeOfUninitializedData	AddressOfEntryPoint	BaseOfCode
BaseOfData	ImageBase	SectionAlignment	FileAlignment
MajorOperatingSystemVersion	MinorOperatingSystemVersion	MajorImageVersion	MinorImageVersion
MajorSubsystemVersion	MinorSubsystemVersion	SizeOfImage	SizeOfHeaders

⁷ Disponible en: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

Checksum	Subsystem	DllCharacteristics	SizeOfStackReserve
SizeOfStackCommit	SizeOfHeapReserve	SizeOfHeapCommit	LoaderFlags
NumberOfRvaAndSizes	SectionsNb	SectionsMeanEntropy	SectionsMinEntropy
SectionsMaxEntropy	SectionsMeanRawsize	SectionsMinRawsize	SectionMaxRawsize
SectionsMeanVirtualsize	SectionsMinVirtualsize	SectionMaxVirtualsize	ImportsNbDLL
ImportsNb	ImportsNbOrdinal	ExportNb	ResourcesNb
ResourcesMeanEntropy	ResourcesMinEntropy	ResourcesMaxEntropy	ResourcesMeanSize
ResourcesMinSize	ResourcesMaxSize	LoadConfigurationSize	VersionInformationSize
Legitimate			

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

Luego de conocer las características del conjunto de datos seleccionado, se procedió a describir las fases en las que fue segmentado el modelo desarrollado. A saber: 1) lectura y limpieza de datos, 2) selección de características, 3) división en conjunto de entrenamiento y prueba, 4) entrenamiento del modelo por los algoritmos de clasificación y 5) validación del modelo. En la **Figura 17** se presenta las fases de la metodología propuesta.

Figura 17

Fases de la metodología propuesta para detección de intrusos



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Lectura y limpieza de datos:

Para leer los datos se utilizó la librería “pandas”, donde se abrió el archivo (.csv) que contiene los datos, en la **Tabla 22** se puede visualizar un poco sobre la estructura del contenido

del conjunto de dato. Además, para el conjunto “X” se eliminaron las columnas que no son significativas, tales como: el nombre de la maquina “*name*”, su encriptación “*md5*” y su característica que la identifica como archivo malicioso o legítimo, ver en **Figura 18**.

Al conjunto “Y” se asigna únicamente la columna “*legitimate*” que es una columna que contiene datos numéricos binarios e indica si un archivo es legítimo (1) o malicioso (0). A este conjunto “Y” se le denomina variable objetivo.

Figura 18

Fragmento de código que ejecuta la lectura y la limpieza de los datos

```
def leerDatos():
    data = pd.read_csv('data.csv', sep='/')

    X = data.drop(['Name', 'md5', 'legitimate'], axis=1).values
    y = data['legitimate'].values
    return X, y, data
```

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Tabla 22

Conjunto de datos

Name	Md5	Machine	SizeOfCode	...	Legitimate
memtest.exe	631ea355665f28d470 748e442fbf5b8	332	361984	...	1
outlook.exe	ca6db5cb169e09209d 0380e398d87b	332	13552640	...	1
⋮	⋮	⋮	⋮	⋮	⋮
VirusShare_4a400b74 7afe6547e09ce0b02da e7f1c	4a400b747afe6547e0 9ce0b02	332	1818586738	...	0

VirusShare_1165700c 4920b4599ffc9b9f785 3f959	1165700c4920b4599f fcb9f7853f959	332	4136960	...	0
---	-------------------------------------	-----	---------	-----	---

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

El detalle del conjunto de datos utilizado puede ser observado en la **Tabla 23**. En ella se observa que el conjunto de datos contiene 57 columnas y 138.047 filas, el 70,07% de estos registros o filas corresponden a los datos de archivos maliciosos y el 29,93% restante corresponde a los datos de archivos legítimos.

Tabla 23

Detalle del conjunto de datos

Detalle	Cantidad
Número de filas con registros	138.047
Número de columnas con registros	57
Filas con características de datos Legítimos	41.323
Filas con características de datos Maliciosos	96.724

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Selección de características:

Se propuso una nueva metodología de selección de subconjunto de características, la cual consiste en combinar dos tipos de métodos de selección, métodos filter y wrapper. Estos métodos filter y wrapper fueron seleccionados porque son los métodos que mejor rendimiento pueden dar en el entrenamiento de modelos utilizando algoritmos de clasificación (scikit-learn, s.f) (Salazar Casares, 2019).

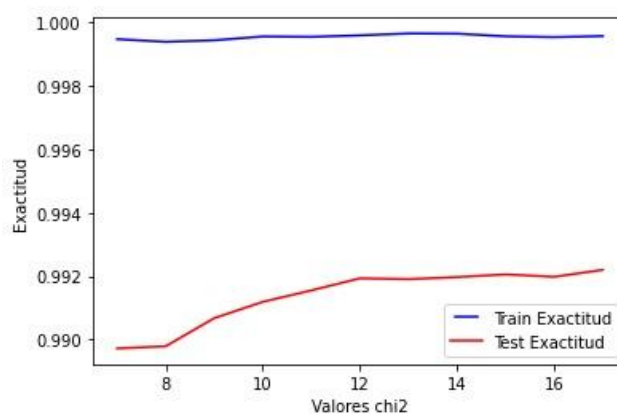
Los métodos filter utilizados fueron chi cuadrado (chi2), f_classif y mutual_info_classif. Estos métodos filter necesitan recibir el valor del parámetro k que indica la cantidad de características a seleccionar. Con la finalidad de encontrar el valor óptimo de k , se evaluó cada uno de los métodos de selección filter, entrenando modelos con los algoritmos clasificadores e identificando el valor que brinde el mejor rendimiento.

“*RandomForestClassifier*” fue el algoritmo clasificador utilizado para evaluar modelos para encontrar los valores k para los métodos filter (chi2, f_classif y mutual_info_classif). Este algoritmo mencionado fue seleccionado en base a una lista de algoritmos clasificadores utilizados en la detección de intrusos descritos en el trabajo de (Ucci, Aniello, & Baldoni, 2019).

En la **Figura 19**, **Figura 20** y **Figura 21**, se presentan los diagramas que muestran los valores de k para los métodos chi cuadrado (chi2), f_classif y mutual_info_classif respectivamente.

Figura 19

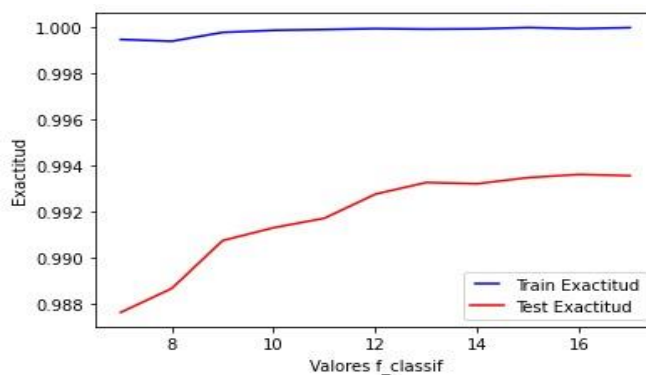
Cálculo de variable k para método chi cuadrado



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Figura 20

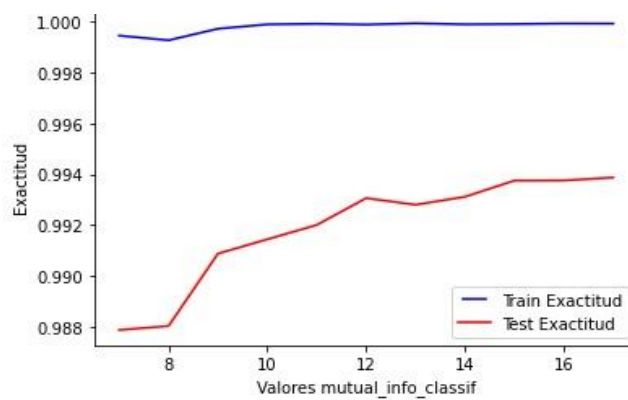
Cálculo de variable k para método $f_classif$



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Figura 21

Cálculo de variable k para método $mutual_info_classif$



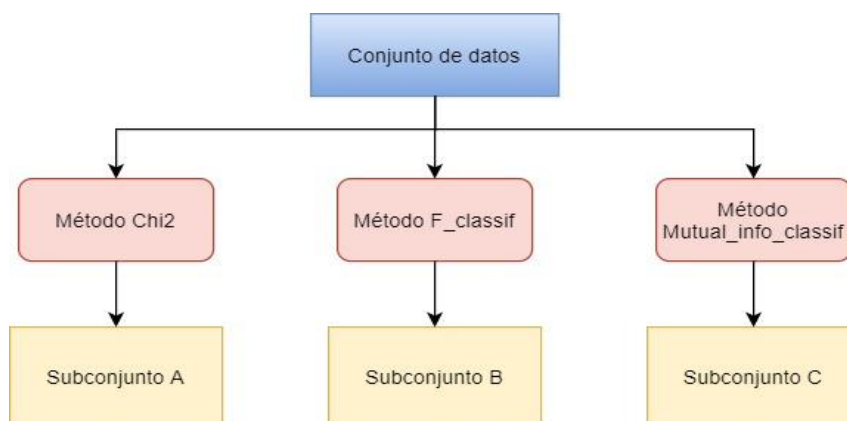
Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Se puede observar en la **Figura 19**, que el valor k que brinda un mejor rendimiento usando el método de selección χ^2 es 14 ($k=14$), en la **Figura 20**, se observa que el valor k óptimo para el método de selección $f_classif$ es 15 ($k=15$) y, por otro lado, para el método $mutual_info_classif$ en la **Figura 21**, se evidencia que el valor de k que permite un mejor rendimiento en el modelo es 15 ($k=15$).

Luego, cada uno de los métodos filter reciben: 1) las matrices de características “X”, y 2) la matriz objetivo “Y”. Con los tres métodos filter considerados en el presente trabajo de titulación, se procede a extraer k características (los valores de k de cada uno de los métodos han sido definidos en el párrafo anterior), obteniéndose un subconjunto de k características por cada método. Este procedimiento se puede observar en el diagrama de la **Figura 22**.

Figura 22

Selección de características métodos filter



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Con el fin de evitar que existan muchas características relevantes en un mismo subconjunto, ya sea este el subconjunto A, B o C (ilustrados en la **Figura 22**), se procede a distribuir uniformemente las características almacenadas en estos subconjuntos (A, B y C) en 3

grupos diferentes. En la **Tabla 24**, se observa el pseudocódigo del algoritmo que permite almacenar uniformemente las características en 3 grupos diferentes.

Tabla 24

Pseudocódigo para almacenar características

Pseudocódigo del algoritmo para almacenar las características uniformemente en 3 grupos

Algoritmo Distribución Características

```

index  $\leftarrow$  0
bandera  $\leftarrow$  false
contador  $\leftarrow$  false
n_subconjunto  $\leftarrow$  3
Para i  $\leftarrow$  0 Hasta tamaño (listaClasificadores) Con Paso 1 Hacer
    n  $\leftarrow$  tamaño (listaClasificadores[i])
    Si contador = false
        contador  $\leftarrow$  true
        Para k  $\leftarrow$  0 Hasta n Con Paso 1 Hacer
            Si index < n_subconjunto Y bandera = false
                lista[index]  $\leftarrow$  listaClasificadores[i][k]
                index  $\leftarrow$  index+1
            SiNo index = n_subconjunto O bandera = True
                Si bandera = False
                    index  $\leftarrow$  index-1
                FinSi
                lista[index]  $\leftarrow$  listaClasificadores[i][k]
                index  $\leftarrow$  index-1
                Si index < 0
                    bandera  $\leftarrow$  False
                    index  $\leftarrow$  index+1
                SiNo
                    bandera  $\leftarrow$  True
                FinSi
            SiNo index < 0 Y bandera = False
                lista[index]  $\leftarrow$  listaClasificadores[i][k]
                index  $\leftarrow$  index+1
            FinSi
        FinPara
    SiNo contador = True
        contador  $\leftarrow$  False
    Para k  $\leftarrow$  n Hasta 0 Con Paso -1 Hacer
        Si index < n_subconjunto Y bandera = False
            lista[index]  $\leftarrow$  listaClasificadores[i][k]
            index  $\leftarrow$  index+1

```

```

SiNo index = n_subconjunto O bandera = True
    Si bandera = False
        index  $\leftarrow$  index-1
    FinSi
    lista[index]  $\leftarrow$  listaClasificadores[i][k]
    index  $\leftarrow$  index-1
    Si index < 0
        bandera  $\leftarrow$  False
        index  $\leftarrow$  index+1
    SiNo
        bandera  $\leftarrow$  True
    FinSi
SiNo index < 0 Y bandera = False
    lista[index]  $\leftarrow$  listaClasificadores[i][k]
    index  $\leftarrow$  index+1
FinSi
FinPara
FinSi
index  $\leftarrow$  0
bandera  $\leftarrow$  False
FinPara
FinAlgoritmo

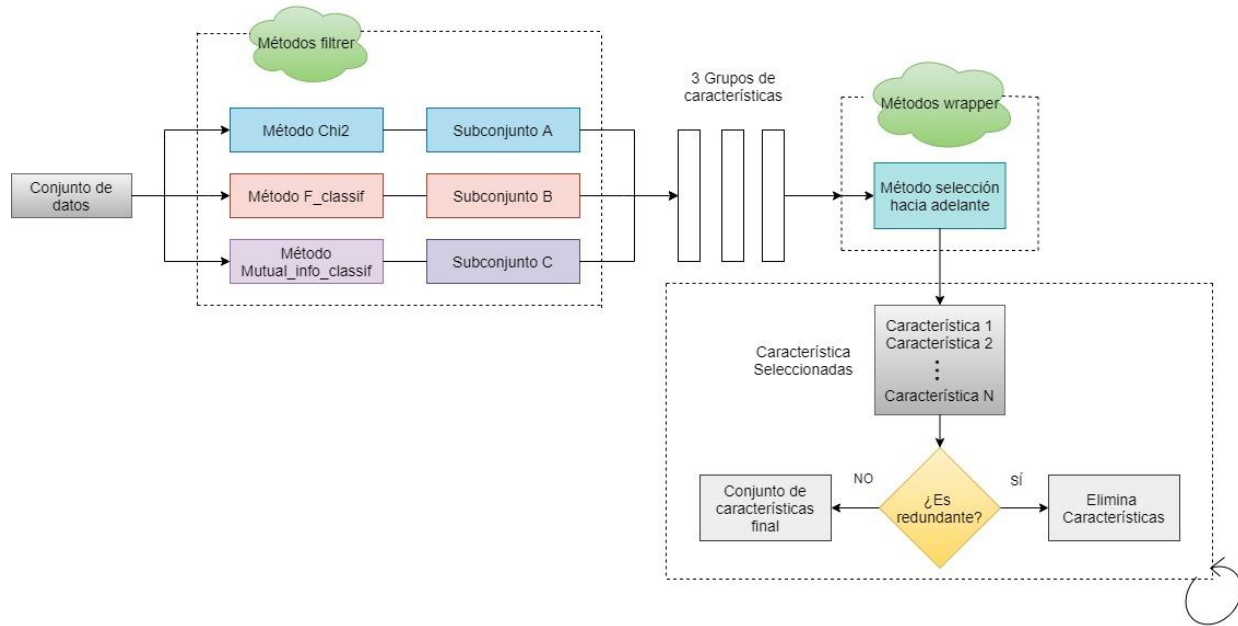
```

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Luego de haber almacenado las características uniformemente entre los 3 grupos, se procede a extraer las mejores características de cada uno de estos mediante el método de selección OLS (Ordinary Least Squares o en español Mínimos cuadrados ordinarios), también llamado método de selección hacia adelante (técnica wrapper descrita en la sección “definiciones conceptuales” situada en el **Capítulo II**). Este proceso se puede ver ilustrado en la **Figura 23**, con esto se obtendrá un conjunto final de características. En dicho conjunto se observará las características redundantes y serán eliminadas del mismo.

Figura 23

Metodología propuesta para la selección de subconjunto de características



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

División del conjunto de datos:

El conjunto de datos que contiene las características seleccionadas anteriormente es dividido en dos grupos: 75% del conjunto de datos está destinado al grupo de entrenamiento y el 25% restante pertenecerá al grupo de prueba. La división se la realiza con la herramienta “*train_test_split*” proporcionada por la librería “*sklearn*” En la **Figura 24** se muestran los parámetros de la división del conjunto de datos.

Figura 24

División del conjunto de datos de entrenamiento y prueba

```
X_train, X_test, y_train, y_test = train_test_split(matriz, y ,test_size=0.25, random_state=1)
```

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Entrenamiento del modelo

Entrenamiento de los modelos por los algoritmos de clasificación:

Ucci, Aniello, y Baldoni (2019) mencionan una lista de algoritmos clasificadores y sus utilidades, una de ellas es la detección de intrusos. En base a esto, se escogió estos algoritmos porque, como indican en la literatura consultada, son los algoritmos que más se utilizan y que mejor rendimiento brindan en la detección de intrusos.

Los algoritmos utilizados son los siguientes: “*DecisionTreeClassifier*”, “*RandomForestClassifier*”, “*GradientBoostingClassifier*”, “*AdaBoostClassifier*”. Los parámetros que se utilizaron en cada uno de los algoritmos se detallan en la **Tabla 25**, los valores de dichos parámetros fueron escogidos mediante la evaluación de los modelos entrenados por los algoritmos mencionados anteriormente, en donde se ingresó a cada uno de estos parámetros diferentes valores y se seleccionaron los valores que produjeron el mejor rendimiento.

Tabla 25

Algoritmos y sus parámetros

Algoritmo	Parámetro	Descripción
DecisionTreeClassifier	criterion="gini"	La función para medir la calidad de una división (impureza y ganancia de información).
	splitter="best"	La estrategia utilizada para elegir la división en cada nodo.
	max_depth=25	La profundidad máxima del árbol.
	min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.
	min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
	min_weight_fraction_leaf=0.0	La fracción ponderada mínima de la suma total de pesos (de todas las

RandomForestClassifier		muestras de entrada) que se requiere para estar en un nodo hoja.
	max_features=12	La cantidad de características a considerar al buscar la mejor división.
	n_estimators=40	Cantidad de árboles en el bosque.
	criterion="gini"	La función para medir la calidad de una división (impureza y ganancia de información).
	max_depth=25	La profundidad máxima del árbol.
	min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.
	min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
GradientBoostingClassifier	max_features=6	La cantidad de características a considerar al buscar la mejor división.
	loss="deviance"	La función de pérdida a optimizar.
	learning_rate=1	La tasa de aprendizaje reduce la contribución de cada árbol.
	n_estimators=40	El número de etapas de impulso a realizar.
	max_depth=4	La profundidad máxima de los estimadores de regresión individuales. La profundidad máxima limita el número de nodos en el árbol.
	min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.
	min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
AdaBoostClassifier	max_features=6	La cantidad de características que se deben considerar al buscar la mejor división.
	n_estimators=85	Número máximo de estimadores en los que finaliza el refuerzo
	learning_rate=1	La tasa de aprendizaje reduce la contribución de cada clasificador.

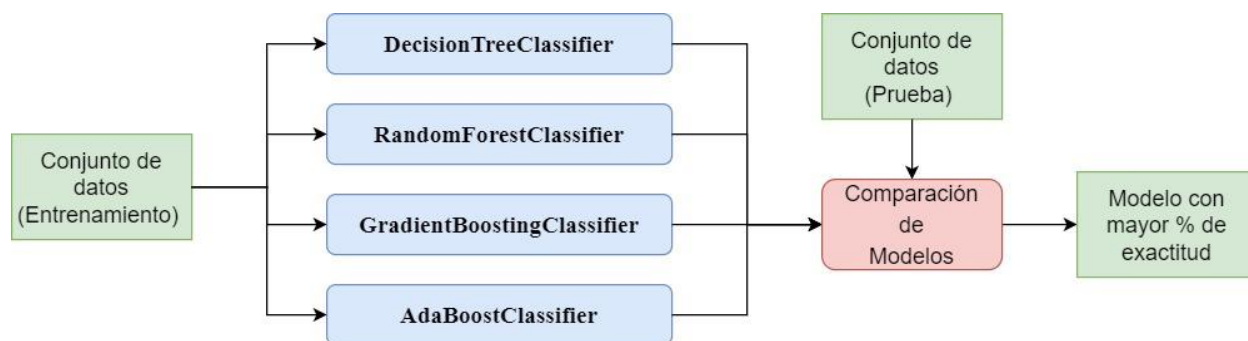
Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Evaluación del modelo

Una vez entrenados los modelos de detección de intrusos con cada uno de los algoritmos presentados anteriormente, se procede a comparar el rendimiento de los modelos con la finalidad de seleccionar a aquel que presente la mayor exactitud. Este proceso se lo puede observar en la **Figura 25**.

Figura 25

Comparación de modelos clasificadores



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

Validación del modelo

El paso final ilustrado en la **Figura 26**, es realizar la validación del modelo entrenado. Para llevar a cabo esta operación, se han desarrollado dos archivos maliciosos y que no han sido utilizados ni para la etapa de entrenamiento ni para la de evaluación del modelo clasificador. El primer archivo consiste en una puerta trasera (BackDoor) “PrimerVirus” y el segundo se trata de un virus espía (KeyLogger) “SegundoVirus”. Estos archivos maliciosos fueron creados con la herramienta msfvenom en el sistema operativo Kali Linux. En la **Figura 27** se puede observar cómo se generaron estos archivos maliciosos.

También se le ha proporcionado, al modelo entrenado, dos archivos legítimos propios de Windows con la finalidad de validar si el modelo puede reconocer tanto los archivos maliciosos como los archivos legítimos.

Figura 26

Pasos de la validación del modelo



Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación

Figura 27

Creación de archivos maliciosos

```

root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.0.102
  LPORT=4444 -f exe -o PrimerVirus.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: PrimerVirus.exe
root@kali:~# msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHO
ST=192.168.0.112 LPORT=1234 -b "\x00" -e x86/shikata_ga_nai -f exe -o SegundoVir
us.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: SegundoVirus.exe
  
```

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Resultados

El producto resultante del presente trabajo de titulación es un modelo de detección de intrusos que ha sido construido a partir de las diversas técnicas de aprendizaje automático. Adicionalmente, se ha propuesto una metodología de selección de subconjunto de características que busca obtener diversidad y nula redundancia entre características. Dicha metodología combina los métodos filter y wrapper al distribuir uniformemente las características obtenidas por los métodos filter en 3 grupos diferentes, y posterior a esto, seleccionar las mejores características de cada uno de estos 3 grupos utilizando un método de selección wrapper.

Con el nuevo conjunto de datos obtenido, estos se dividen en: 1) datos de entrenamiento y 2) datos de prueba. Una vez obtenida esta división, se entrena el modelo con cada uno de los algoritmos de clasificación mencionados en apartados anteriores.

Posteriormente, se ha evaluado cada uno de los modelos obtenidos mediante diversas métricas con la finalidad de seleccionar aquel que obtenga la mejor exactitud.

El modelo desarrollado fue entrenado en diez ocasiones, tanto con la metodología de selección de subconjunto de características propuesta y sin la metodología propuesta, esta se diferencia, porque la selección de características se la realiza aleatoriamente utilizando la herramienta de sklearn llamada “ExtraTreesClassifier”.

Por cada una de estas iteraciones se obtuvieron las métricas de: exactitud, sensibilidad (+), especificidad (-) y F1-score (puntaje). En la **Tabla 26**, se evidencia el cálculo de la media y desviación estándar de las métricas mencionadas.

Tabla 26*Media y desviación estándar de las métricas*

Con la metodología propuesta				
	Exactitud	Sensibilidad (+)	Especificidad (-)	F1-score
DecisionTree	99.15 \pm 0.040	98.71 \pm 0.1128	99.35 \pm 0.0290	98.67 \pm 0.0699
RandomForest	99.43 \pm 0.018	99.17 \pm 0.040	99.54 \pm 0.012	99.07 \pm 0.0185
GradientBoosting	99.07 \pm 0.046	98.58 \pm 0.103	99.28 \pm 0.039	98.52 \pm 0.055
AdaBoost	98.74 \pm 0.002	97.87 \pm 0.003	99.12 \pm 0.000	97.92 \pm 0.00
Sin la metodología propuesta				
	Exactitud	Sensibilidad (+)	Especificidad (-)	F1-score
DecisionTree	99.10 \pm 0.025	98.65 \pm 0.051	99.29 \pm 0.027	98.54 \pm 0.1934
RandomForest	99.33 \pm 0.012	99.02 \pm 0.047	90.46 \pm 28.46	99.00 \pm 0.029
GradientBoosting	98.91 \pm 0.076	98.24 \pm 0.123	99.19 \pm 0.058	98.40 \pm 0.0629
AdaBoost	98.56 \pm 0.004	97.4107 \pm 0.000	99.06 \pm 0.001	97.58 \pm 0.00

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Como se evidencia en la **Tabla 26**, mediante la aplicación de la metodología propuesta se ha obtenido una ligera mejoría en el rendimiento de los modelos entrenados por los algoritmos de clasificación. Esto se puede observar al comparar los resultados de las métricas de: exactitud, sensibilidad, especificidad y F1-score (puntaje). Así, por ejemplo, el resultado del modelo entrenado con mejor puntuación de exactitud, es decir, aquel modelo entrenado con RandomForest, tuvo una mejoría aproximadamente del 0.10% en la media de exactitud con respecto al modelo entrenado sin la metodología propuesta. Con ello, se está aportando con una

nueva metodología de selección de características cuyo rendimiento es similar a los métodos existentes en la literatura.

Por otra parte, el no considerar la metodología propuesta tiene como punto débil el no garantizar la exclusión de características redundantes. En cambio, la metodología propuesta permite separar características irrelevantes y redundantes para luego puntuar dichas características y elegir sólo aquellas que poseen mayor puntuación. Para ello, se hace uso de una combinación de técnicas filter y wrapper para mitigar la poca diversidad de características en los subconjuntos y la redundancia que puede existir entre ellas, tal como lo mencionan (Cai, Luo, Wang, & Yang, 2018).

Los resultados de los algoritmos clasificadores se obtuvieron en un tiempo aproximado de un minuto, como se realizaron 10 pruebas, se escogió una de ellas aleatoriamente y los resultados de esta prueba se muestra en la **Tabla 27**, donde se muestran las siguientes métricas:

- Falsos Negativos: FN
- Falsos Positivos: FP
- Exactitud: E
- Sensibilidad: S+
- Especificidad: S-
- F1-score: F

Tabla 27

Resultados de los algoritmos

Algoritmo	FN	FP	E	S+	S-	F
DecisionTreeClassifier	1.40	0.55	99.18	98.59	99.44	98.65
RandomForestClassifier	0.86	0.44	99.42	99.13	99.55	99.05
GradientBoostingClassifier	1.45	0.64	99.11	98.54	99.35	98.53

AdaBoostClassifier	2.12	0.87	98.74	97.87	99.12	97.92
---------------------------	------	------	-------	-------	-------	-------

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

El modelo entrenado por el algoritmo con mejor porcentaje de exactitud fue “RandomForestClassifier”. El cual arrojó un 99,42% de exactitud clasificando correctamente 23942 datos como casos negativos, lo que representa un 99.55% (especificidad) y 10375 datos como casos positivos, que representa un 99.13% (sensibilidad).

La matriz de confusión, con las predicciones correctas e incorrectas de la clase positiva y negativa, se reflejan en la **Tabla 28**, donde 23942 es el número de predicciones correctas de clase negativa (Negativos reales), 108 es el número de predicciones incorrectas de clase positiva (Falsos positivos), 90 es el número de predicciones incorrectas de clase negativa (Falsos negativos) y 10372 es el número de predicciones correcta de clase positiva (Positivos reales).

Tabla 28

Matriz de confusión

		Predicción	
		Negativo	Positivo
Reales	Negativo	23942	108
	Positivo	90	10372

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

Con el modelo entrenado se procede a realizar las pruebas de predicción del modelo. En este proceso se usaron archivos ejecutables explicados en el apartado anterior, donde el modelo determinará si estos archivos son maliciosos o legítimos. Con esto se prueba que el modelo de detección de intrusos resultante de la investigación realizada es capaz de reconocer si un archivo ejecutable es de clase maliciosa o legítima, esto se puede evidenciar en la **Figura 28**. Cabe destacar, que los archivos legítimos son nativos de Windows y los archivos maliciosos fueron

creados para el sistema operativo Windows. Se lo hizo así dado que el conjunto de datos utilizado contiene las características de un archivo ejecutable (PE) de Windows.

Figura 28

Prueba de predicción del modelo

```
(base) C:\Users\Jatu11\Desktop\desarrollo modelo prevención intrusos>python checkpe.py PrimerVirus.exe
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.ensemble.forest module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.ensemble. Anything that cannot be imported from sklearn.ensemble is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.tree.tree module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.tree. Anything that cannot be imported from sklearn.tree is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
The file PrimerVirus.exe is malicious
```

```
(base) C:\Users\Jatu11\Desktop\desarrollo modelo prevención intrusos>python checkpe.py SegundoVirus.exe
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.ensemble.forest module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.ensemble. Anything that cannot be imported from sklearn.ensemble is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.tree.tree module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.tree. Anything that cannot be imported from sklearn.tree is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
The file SegundoVirus.exe is malicious
```

```
(base) C:\Users\Jatu11\Desktop\desarrollo modelo prevención intrusos>python checkpe.py conda-verify.exe
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.ensemble.forest module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.ensemble. Anything that cannot be imported from sklearn.ensemble is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.tree.tree module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.tree. Anything that cannot be imported from sklearn.tree is now part of the private API.
  warnings.warn(message, FutureWarning)
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\Jatu11\anaconda3\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estimator RandomForestClassifier from version 0.20.3 when using version 0.23.2. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(
The file conda-verify.exe is legitimate
```

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

En el apartado de la propuesta del proyecto, se detalla la creación de los archivos maliciosos. En la **Figura 28** se observó que, en efecto, los archivos fueron detectados como malicioso por el modelo desarrollado en este trabajo de investigación.

En la **Figura 29** se evidencia, que los archivos “PrimerVirus.exe” y “SegundoVirus.exe” son detectados como archivos maliciosos, por medio de una exploración realizada con el antivirus “ESET”.

Figura 29

Análisis y confirmación de archivos maliciosos

 INTERNET SECURITY
Exploración del equipo
Registro de la exploración
Versión del motor de detección: 21844 (20200818)
Fecha: 16/1/2021 Hora: 14:28:12
Discos, carpetas y archivos explorados: C:\Users\Jatu11\Desktop\carpeta compartida\pruebas.rar
C:\Users\Jatu11\Desktop\carpeta compartida\pruebas.rar = RAR5 = aplicacion.jpg - Win64/Rozena.J troyano - eliminado
C:\Users\Jatu11\Desktop\carpeta compartida\pruebas.rar = RAR5 = appvirus.jpg - Win64/Rozena.J troyano - eliminado
Cantidad de objetos explorados: 3
Cantidad de detecciones: 2
Cantidad de objetos desinfectados: 2
Tiempo restante: 14:28:23 Tiempo total de exploración: 11 seg (00:00:11)

Nota: Elaborado por Dayannara Avila y Joel Torres. Fuente: Datos de la investigación.

El archivo para comprobar que el modelo desarrollado reconoce archivos legítimos fue en archivo “conda-verify.exe”, dicho archivo es propio del software libre Anaconda 2, pero también se puede realizar esta prueba con cualquier archivo nativo de Windows.

Criterios de validación de la propuesta

El presente trabajo de titulación utilizó el juicio de expertos para su validación. Para llevar a cabo la misma, se consideró a dos docentes de la Facultad de Ciencias Matemáticas y Físicas, y a un ingeniero en sistemas computacionales, con el fin de que ellos validaran la propuesta. A los docentes considerados se les remitió un correo por parte de los investigadores del proyecto, el cual contenía la propuesta y cómo había sido el proceso para su desarrollo. Adicionalmente, se envió los respectivos formatos que debían ser llenados para la constancia del juicio de expertos realizado.

Análisis de indicadores

Entre los ingenieros que fueron seleccionados para formar parte del juicio de expertos, se encuentra el M.Sc. Jorge Charco, docente con basto conocimiento en el campo de la Inteligencia Artificial, el M.Sc. Lorenzo Cevallos, el cual forma parte de los docentes investigadores de la Facultad de Ciencias Matemáticas y Físicas y con amplio conocimiento en estadística y el Ing. Bryan Manzaba, con experiencia en desarrollo de software.

Los indicadores seleccionados para la evaluación del presente proyecto fueron: Claridad, Actualidad, Disponibilidad, Consistencia, Metodología, Aplicabilidad. Todos estos indicadores contenían un criterio.

Indicadores con su respectivo Criterio:

Claridad: Se utiliza el lenguaje de programación apropiado que facilita la comprensión

Actualidad: Está acorde a los aportes recientes en la disciplina de estudio

Disponibilidad: El producto cumple con estándares de disponibilidad.

Consistencia: Está basado en aspectos teóricos y científicos

Metodología: El instrumento se relaciona con el método planteado en el proyecto

Aplicabilidad: El instrumento es de fácil aplicación.

Validación de Expertos

Análisis de Experto: M.Sc. Jorge Luis Charco Aguirre

El M.Sc. Jorge Charco, por medio del correo enviado por los investigadores pudo conocer sobre la propuesta y el desarrollo de esta. Se receiptó los documentos enviados con su respectiva firma, dando por hecho la validación respectiva. Todos los ítems recibieron la máxima puntuación “Excelente” que arrojó un 6/6 en la sumatoria final. Por parte del M.Sc. Jorge Charco, no hubo observaciones y se procedió a adjuntar los documentos respectivos en el anexo 7.

Análisis de Experto: M.Sc. Lorenzo Jovanny Cevallos Torres

El M.Sc. Lorenzo Cevallos fue el segundo docente de la facultad que colaboró con la validación de expertos del presente proyecto. El M.Sc. Lorenzo Cevallos se interesó en conocer un poco más del tema y que se contestarán sus dudas. Para esto se realizó una reunión por la plataforma Teams con los investigadores. Entre las preguntas realizadas se mencionó el conjunto de datos utilizado y cómo este había sido dividido para el entrenamiento y prueba del modelo. Adicionalmente, el M.Sc. Lorenzo Cevallos preguntó por el lenguaje de programación usado y mencionó que para este tipo de problemas que se está solventando es recomendable usar el lenguaje de programación R. Otras de las preguntas tuvieron relación con los resultados, y posterior a ellos se indicó cuál fue el mejor modelo y el porcentaje de exactitud. Para validar aquello, se mostró el funcionamiento del software. Todas las inquietudes fueron contestadas. Como conclusión, el M.Sc. Lorenzo Cevallos pudo entender un poco más del tema y procedió a realizar la respectiva validación. Los ítems fueron calificados como “Excelente” y la sumatoria total fue 6/6. Posterior a ello se recibió los documentos con la respectiva firma y se adjuntó al anexo 7.

Análisis de Experto: Ing. Bryan Cristopher Manzaba Lindao

Con el Ing. Bryan Manzaba, se pactó una reunión por medio de la plataforma Google Meets para resolver sus inquietudes. Además, visualizó el software en funcionamiento, lo cual permitió que entendiera la propuesta y como había sido desarrollada. Las preguntas realizadas por el Ing. Bryan Manzaba se centraron en el conjunto de datos utilizados y si este había pasado por un preprocesamiento previo al entrenamiento del modelo. Adicionalmente, preguntó si el aplicativo podía ser utilizado en otro sistema operativo, y en este caso, se le mencionó que era exclusivo para el sistema operativo Windows. Por otro lado, se interesó en saber cuál era el tiempo de respuesta del modelo para realizar la predicción. Por parte del Ing. Bryan Manzaba calificó los ítems como “Excelente” y la puntuación final fue de 6/6. Se receptó el correo respectivo, el cual contenía los documentos que validaban la propuesta con la respectiva firma y se adjuntó al anexo 7.

Síntesis

De manera general, los tres expertos calificaron la propuesta como apta para el presente proceso de titulación. Todas las dudas de los expertos fueron contestadas y ayudaron a resolver sus inquietudes.

CAPÍTULO IV

En este capítulo se detalla el cumplimiento de cada uno de los objetivos específicos planteados al inicio del presente trabajo de titulación. También se redactan recomendaciones que se pueden adoptar para optimizar la técnica implementada en el presente proyecto.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Para comenzar con el desarrollo del presente trabajo de titulación, se priorizó conocer las diversas técnicas de aprendizaje automático. Esto se logró mediante la búsqueda y revisión de: 1) tesis de grado, 2) tesis doctorales y 3) artículos de revistas científicas que aborden la problemática planteada. Por lo que, se obtuvo suficiente conocimiento para identificar y seleccionar las técnicas adecuadas que ayuden a cumplir el propósito del presente proyecto.
- Se investigó también las diferentes clases de algoritmos que existen en el aprendizaje automático, identificando aquellos que han proporcionado los mejores resultados en la detección de intrusos.
- Para que el presente trabajo de titulación se catalogue como un trabajo original, se propuso una metodología de selección de subconjunto de características. Dicha metodología fue propuesta, debido que, se realizó una búsqueda de las técnicas existentes para la selección de características, y con ello, se extrajo las desventajas que se indicaban en los artículos científicos consultados, como lo indicaron Cai, Luo, Wang, y Yang (2018) en la selección de subconjunto de características existe poca diversidad de características y redundancia entre ellas. En base a estas desventajas identificadas, se propuso una metodología de selección de subconjunto de características que combine métodos de selección filter y wrapper, con el

objetivo de buscar un conjunto resultante de características relevantes y donde no exista redundancia entre ellas.

- Para comenzar el desarrollo del modelo predictivo, se realizó una lectura y análisis del conjunto de datos. En este proceso se procedió a eliminar las características irrelevantes que se encontraban dentro del conjunto de datos. Posterior a esto, se desarrolló la metodología de selección de subconjunto de características propuesta.
- Una vez definido los algoritmos clasificadores a utilizar, se procedió a ajustar sus parámetros, escogiendo los valores que permitieron optimizar el entrenamiento de los modelos con dichos algoritmos.
- Una vez evaluado cada uno de los modelos entrenados con o sin el uso de la metodología propuesta, se procedió a comparar los valores de sus métricas de rendimiento con el propósito de identificar el modelo de mejores prestaciones, dicho modelo fue el entrenado por el algoritmo “RandomForest”. Aquello, permitió evidenciar que la metodología propuesta produce similares resultados a los obtenidos por otras técnicas de selección de características, con la diferencia de que se garantiza la exclusión de características redundantes.

Recomendaciones

- Implementar otros tipos de algoritmos clasificadores ajustando sus parámetros que mejoren el rendimiento del entrenamiento.
- Revisar otras técnicas de selección de características (filter o wrapper) en la metodología propuesta, y con ello, comprobar si la técnica propuesta en el presente trabajo de investigación puede ser optimizada.
- Agregar más características al conjunto de datos, ya sean estas, propias de los archivos ejecutables de Windows u otro sistema operativo.

Trabajo futuro

- Profundizar en la fase de selección de características, buscando una mejoría de la metodología propuesta u otra ya existente.
- Extrapolar la aplicación la metodología de selección de subconjunto de características propuesta a otros dominios que requieran de un aprendizaje automático, por ejemplo, para diagnóstico clínico, detección de fraude bancario, entre otros.

REFERENCIAS BIBLIOGRÁFICAS

Referencias

- Akinsola, J. (8 de junio de 2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48, 128-138. doi:10.14445/22312803/IJCTT-V48P126
- Alan Neill, D., & Cortez Suárez, L. (2018). *Procesos y Fundamentos de la Investigación Científica*. Machala: UTMACH. Obtenido de Utmach: <http://repositorio.utmachala.edu.ec/handle/48000/12498>
- Alvarado Chang, J. E. (mayo de 2020). Análisis de ataques cibernéticos hacia el Ecuador. *Revista Científica Aristas*. Obtenido de https://revistacientificaistjba.edu.ec/images/home/documentos/Mayo_2020/2.pdf
- Alvear Reinoso, F. X. (s.f de febrero de 2019). *Análisis y diseño de una propuesta para mitigar ataques cibernéticos a correos electrónicos utilizando técnicas de hacking ético*. Obtenido de Repositorio Institucional de la Universidad Politécnica Salesiana: <http://dspace.ups.edu.ec/handle/123456789/17035>
- Amat Rodrigo, J. (s.f de octubre de 2020). *Gradient Boosting con Python*. Obtenido de Ciencia de Datos: https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html
- Avenía, C. (s.f de noviembre de 2017). *Fundamentos de seguridad informática*. Obtenido de Areandina: <https://digitk.areandina.edu.co/bitstream/handle/areandina/1367/Fundamentos%20de%20seguridad%20inform%C3%A1tica.pdf?sequence=1&isAllowed=y>
- Baca Urbina, G. (2016). *Introducción a la Seguridad Informatica*. México: Grupo Editorial Patria. Obtenido de

https://books.google.com.ec/books/about/Introducci%C3%B3n_a_la_seguridad_inform%C3%A1tica.html?id=IhUhDgAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q&f=false

Balparda, N. (24 de noviembre de 2020). *Introducción a Machine Learning*. Obtenido de gub.uy:

<https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/sites/agencia-gobierno-electronico-sociedad-informacion-conocimiento/files/2020-11/20201124%20-%20Introducci%C3%B3n%20a%20Machine%20Learning.pdf>

Bataghva Shahbaz, M., Wang, X., Behnad, A., & Samarabandu, J. (2016). On efficiency enhancement of the correlation-based feature selection for intrusion detection systems. *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (págs. 1-7). Vancouver: IEEE. doi:10.1109/IEMCON.2016.7746286

BBC. (17 de diciembre de 2016). *Cómo fue el 'hackeo' de piratas informáticos de Rusia durante las elecciones de Estados Unidos*. Obtenido de BBC: <https://www.bbc.com/mundo/noticias-internacional-38350244>

BBC. (12 de mayo de 2017). *El ciberataque de escala mundial y "dimensión nunca antes vista" que afectó a instituciones y empresas de unos 150 países*. Obtenido de BBC: <https://www.bbc.com/mundo/noticias-39903218>

BBC. (17 de julio de 2020). *Hackeo a Twitter: por qué es importante aclarar qué hay detrás del "mayor ataque de la historia" a la red social*. Obtenido de BBC: <https://www.bbc.com/mundo/noticias-53443772>

- Belavagi, M., & Muniyal, B. (31 de diciembre de 2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Procedia Computer Science*, 89, 117-123. doi:<https://doi.org/10.1016/j.procs.2016.06.016>
- Binbusayyis, A., & Vaiyapuri, T. (9 de julio de 2020). Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. *Heliyon*, 6(7). doi:<https://doi.org/10.1016/j.heliyon.2020.e04262>
- Bolón Canedo, V., & Alonso Betanzos, A. (2019). Ensembles for feature selection: A review and future trends. *Information Fusion*, 1-12. doi:<https://doi.org/10.1016/j.inffus.2018.11.008>
- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 70-79. doi:<https://doi.org/10.1016/j.neucom.2017.11.077>
- Cisco. (s.f de s.f de 2018). *Reporte Anual de Ciberseguridad de Cisco 2018*. Obtenido de Cisco: https://www.cisco.com/c/dam/global/es_mx/solutions/pdf/reporte-anual-cisco-2018-espan.pdf?dtid=oemzzz000233&oid=rptsc007228&ccid=cc000153
- Diario EL COMERCIO. (23 de septiembre de 2019). *Ecuador ocupa el séptimo lugar en ciberseguridad en América Latina*. Obtenido de EL COMERCIO : <https://www.elcomercio.com/actualidad/ecuador-ciberseguridad-region-informe-delitos.html>
- Diario EL COMERCIO. (15 de octubre de 2020). *Las ciberamenazas se multiplican a escala mundial por la pandemia*. Obtenido de EL COMERCIO: <https://www.elcomercio.com/tendencias/ciberamenazas-ataque-informatico-seguridad-instituciones.html>
- Dueñas Quesada, J. M. (2 de junio de 2020). *Aprendizaje supervisado para la detección de amenazas web mediante clasificación basada en árboles de decisión*. Obtenido de

Universitat Oberta de Catalunya:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/118166/6/joseduenasTFM0620memoria.pdf>

ESET. (s.f de s.f de 2017). *Guía de Ransomware*. Obtenido de welivesecurity: <https://www.welivesecurity.com/wp-content/uploads/2017/10/guia-ransomware-eset.pdf>

ESET. (7 de noviembre de 2018). *¿Es posible que la Inteligencia Artificial potencie el malware a futuro?* Obtenido de eset: https://www.eset.com/fileadmin/ESET/LATAM/pdf/Machine_learning_WP_ES.pdf

ESET. (s.f de s.f de 2019). *ESET Security Report Latinoamérica 2019*. Obtenido de welivesecurity: <https://www.welivesecurity.com/wp-content/uploads/2019/07/ESET-security-report-LATAM-2019.pdf>

ESET. (s.f de s.f de 2020). *ESET Security Report Latinoamérica 2020*. Obtenido de welivesecurity: https://www.welivesecurity.com/wp-content/uploads/2020/08/ESET-Security-Report-LATAM_2020.pdf

Espinosa Zuñiga, J. J. (2020). Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. *Ingeniería Investigación y Tecnología*, 21(1). doi:<https://doi.org/10.22201/fi.25940732e.2020.21n1.008>

Freire Fajardo, F. F. (20 de diciembre de 2017). *Plan de Contingencia ante Ciberataques*. Obtenido de Repositorio Dspace: <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/42124>

Freire López, K. B. (18 de septiembre de 2017). *Estudio y análisis de ciberataques en América Latina, su influencia en las empresas del Ecuador y propuesta de políticas de*

- ciberseguridad*. Obtenido de Universidad Católica de Santiago de Guayaquil:
<http://repositorio.ucsg.edu.ec/handle/3317/9203>
- Gago Utrera, R. (s.f de Junio de 2017). *Uso de algoritmos de aprendizaje automático aplicados a base de datos genéticos*. Obtenido de Universitat Oberta de Catalunya:
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65426/6/rgagoTFM0617memoria.pdf>
- García , S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing. *ELSEVIER*, 98, 1-29. doi:<https://doi.org/10.1016/j.knosys.2015.12.006>
- IESS. (12 de agosto de 2020). *IESS*. Obtenido de Twitter:
<https://twitter.com/IESSec/status/1293571624327942146/photo/1>
- Izaguirre Olmedo, J., & León Gavilánez, F. (29 de septiembre de 2018). Análisis de los ciberataques realizados en América Latina. *INNOVA Research Journal*, 3(9), 172-181.
doi:<https://doi.org/10.33890/innova.v3.n9.2018.837>
- Kok, S. H., Azween, A., & Jhanjhi, N. Z. (2020). Evaluation metric for crypto-ransomware detection using machine learning. *Journal of Information Security and Applications*, 55.
doi:<https://doi.org/10.1016/j.jisa.2020.102646>
- Llamas Covarrubias, J. Z., & Llamas Covarrubias, I. N. (2018). *Internet, ¿Arma o Herramienta?* Guadalajara: Universidad de Guadalajara. Obtenido de
https://www.researchgate.net/publication/331075407_Internet_Arma_o_Herramienta/citation/download
- Llopis Polvoreda, J. (s.f de s.f de 2017). *Sistema de monitorización Snort*. Obtenido de Repositorio Institucional Universidad Politécnica de Valencia:
<https://riunet.upv.es/bitstream/handle/10251/88474/LLOPIS%20->

%20Sistema%20de%20monitorizaci%3b3n%20del%20IDS%20Snort.pdf?sequence=1&isAllowed=y

Martínez Landrove, N. (s.f de agosto de 2019). *Ciberseguridad y riesgo operacional en las organizaciones*. Obtenido de Repositorio Universidad Pontificia Comillas: <https://repositorio.comillas.edu/jspui/bitstream/11531/42317/1/TFM001173.pdf>

Maseda Tarin, M. (02 de enero de 2019). *Reducción de la dimensionalidad mediante métodos de selección de características en microarrays ADN*. Obtenido de Univesitat Oberta de Catalunya: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/89668/11/mmasedaTFG0119memoria.pdf>

Mateo, I., & Neira Cedillo, E. (julio de 2017). Tendencias de seguridad y vulnerabilidades en sistemas basados en nube. *Revista multidisciplinaria de investigación científica*, 1(6). Obtenido de <https://www.revistaespirales.com/index.php/es/article/view/28/37>

Medina Medrano, R. F., & Ñique Chacón, C. I. (23 de septiembre de 2017). Bosques aleatorios como extensión de los árboles de clasificación con los programas R y Python. *Interfases*, 165-189. Obtenido de Dialnet: <https://dialnet.unirioja.es/servlet/articulo?codigo=6230447>

Mehmood, T., & Rais, H. B. (2016). Machine learning algorithms in context of intrusion detection. *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)* (págs. 369-373). Kuala Lumpur: IEEE. doi:10.1109/ICCOINS.2016.7783243

Mungloo Dilmohamud, Z., Marigliano, G., Jaufeerally Fakim, Y., & Peña Reyes, C. (2018). A Comparative Study of Feature Selection Methods for Biomarker Discovery. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (págs. 2789-2791). Madrid: IEEE. doi:10.1109 / BIBM.2018.8621267

pandasecurity. (s.f de s.f de s.f). *Phishing*. Obtenido de pandasecurity:

<https://www.pandasecurity.com/es/security-info/phishing/>

Parrales Bravo, F. R. (s.f de s.f de 2020). *Metodologías de procesamiento de datos en el ámbito de e-Health para la categorización de respuestas terapéuticas en pacientes con migraña*.

Obtenido de Universidad Complutense de Madrid:

<https://web.fdi.ucm.es/fdi/Documentos.aspx?cod=35875>

Pastorino, C. (14 de marzo de 2017). *8 errores de seguridad en empresas que complican la vida de sus usuarios*. Obtenido de welivesecurity: <https://www.welivesecurity.com/la-es/2017/03/14/errores-de-seguridad-empresas/>

Pushpalatha, K. R., & Gowda Karegowda, A. (2017). CFS Based Feature Subset Selection for Enhancing. *2017 2nd International Conference On Emerging Computation and Information Technologies (ICECIT)* (págs. 1-6). Tumakuru: IEEE. doi:10.1109/ICECIT.2017.8453403

Qiubing , R., Mingchao , L., & Shuai , H. (19 de febrero de 2019). Tectonic discrimination of olivine in basalt using data mining techniques based on major elements: a comparative study from multiple perspectives. *Big Earth Data*, 3, 1-18. doi:10.1080/20964471.2019.1572452

Quevedo Muñoz, J. A. (29 de junio de 2020). *Desarrollo de un modelo de clasificación de malware Linux ARM según su funcionalidad utilizando técnicas de aprendizaje automático*.

Obtenido de Archivo Digital UPM: <http://oa.upm.es/64465/>

Ramírez Hinestroza, D. (s.f de s.f de 2018). *El Machine Learning a través de los tiempos, y los aportes a la humanidad*. Obtenido de Repositorio Institucional UniLibre:

<https://repository.unilibre.edu.co/bitstream/handle/10901/17289/EL%20MACHINE%20LEARNING.pdf?sequence=1&isAllowed=y>

Real Academia Española. (s.f de s.f de s.f). *Algoritmo*. Obtenido de Real Academia Española:

<https://dle.rae.es/algoritmo?m=form>

Rivero Pérez, J. L., Ribeiro, B., & Ortiz, K. H. (diciembre de 2016). Comparación de algoritmos para detección de intrusos en entornos estacionarios y de flujo de datos. *Universidad y Sociedad*, 32-42. Obtenido de SciELO:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000400004&lng=es&tlng=es

Rodríguez Rama, J. M. (4 de junio de 2018). *Aplicación de técnicas de Machine Learning a la detección de ataques*. Obtenido de Universitat Oberta de Catalunya :

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81126/11/jmrodriguez85TFM0618memoria.pdf>

Rueda , C. (04 de marzo de 2020). *Ecuador está en el grupo de los rezagados en ciberseguridad*.

Obtenido de Expreso: <https://www.expreso.ec/actualidad/ecuador-grupo-rezagados-temas-ciberseguridad-6240.html>

Salazar Casares, P. S. (22 de mayo de 2019). *Aplicación de modelos de Feature Selection y Machine Learning para identificar inhibidores potentes de la tirosinasa*. Obtenido de

Universidad San Francisco de Quito:

<http://repositorio.usfq.edu.ec/bitstream/23000/8475/1/143730.pdf>

Sandoval, L. J. (2018). Algoritmos de aprendizaje automático para el análisis y predicción de datos. *Revista Tecnológica*(11). Obtenido de

http://www.redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

Sanna Morales, C. X., & Londoño Castaño, S. A. (s.f de s.f de 2018). *Modelo de Detección de Intrusos usando técnicas de Aprendizaje de Máquina*. Obtenido de Repositorio digital tecnológico de Antioquia:
<https://dspace.tdea.edu.co/bitstream/handle/tda/442/MODELO%20DE%20DETECCION%20DE%20INTRUSOS%20USANDO%20TECNICAS%20DE%20APRENDIZAJE.pdf?sequence=1>

scikit-learn. (s.f de s.f de s.f). *sklearn.feature_selection.chi2*. Obtenido de scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html#sklearn.feature_selection.chi2

scikit-learn. (s.f de s.f de s.f). *sklearn.feature_selection.f_classif*. Obtenido de scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html#sklearn.feature_selection.f_classif

scikit-learn. (s.f de s.f de s.f). *sklearn.feature_selection.SelectKBest* . Obtenido de scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest

scikit-learn. (s.f de s.f de s.f). *sklearn.model_selection.train_test_split*. Obtenido de scikit-learn:
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html#sklearn.model_selection.train_test_split

- Shao-bo, D. (2017). Intrusion Feature Selection Method based on Neighborhood Distance. 2017 *International Conference on Computer Systems, Electronics and Control (ICCSEC)* (págs. 748-751). Dalian: IEEE. doi:10.1109/ICCSEC.2017.8446849
- Simba Amores, J. P., & Espinoza Padilla, B. A. (s.f de s.f de 2019). *Modelo para la detección y mitigación de ataques de suplantación de identidad, utilizando aprendizaje automático*. Obtenido de Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/20581/T-ESPE-039659.pdf?sequence=1&isAllowed=y>
- Spasova Dimitrova, R. (31 de mayo de 2017). *Desarrollo y evaluación de métodos de selección de características para la predicción de eventos adversos en pacientes polimedicados*. Obtenido de Academia-e: <https://hdl.handle.net/2454/24594>
- Tharwat, A. (s.f de febrero de 2018). *AdaBoost classifier: an overview*. doi:10.13140/RG.2.2.19929.01122
- Toapanta Toapanta, S., Coello Ocha , I., Naranjo Sanchez, R., & Gallegos Mafla, L. E. (2019). Impact on Administrative Processes by Cyberattacks in a Public Organization of Ecuador. 2019 *Third World Conference on Smart Trends in Systems Security and Sustainability (WorldS4)* (págs. 270-274). London: IEEE. doi:10.1109/WorldS4.2019.8903967
- Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. *Computers & Security*, 123-147. doi:<https://doi.org/10.1016/j.cose.2018.11.001>
- Universidad de Jaén. (s.f de octubre de 2018). *Guías de seguridad UDJ Software malicioso (malware)*. Obtenido de Universidad de Jaén: https://www.ujaen.es/servicios/sinformatica/sites/servicio_sinformatica/files/uploads/guia_spracticas/Guias%20de%20seguridad%20UJA%20-%20203.%20Malware.pdf

Universidad de Jaén. (s.f de s.f de s.f). *Diseño Documental*. Obtenido de Universidad de Jaén:

http://www.ujaen.es/investiga/tics_tfg/dise_documental.html

Urcuqui López, C. (12 de julio de 2016). *Módulo de machine learning para detección de malware en Android*. doi:10.13140/RG.2.2.35287.06564

Valdez Alvarado, A. (26 de enero de 2019). *Machine Learning para todos*. doi:10.13140/RG.2.2.13786.70086

Valdez Alvarado, A. R. (12 de marzo de 2017). *Introducción al machine learning con BigML*. doi:10.13140/RG.2.2.14186.47041

Valencia Peral, A. (04 de junio de 2019). *Técnica de aprendizaje automático para la detección de ataques en el tráfico de red*. Obtenido de Universitat Oberta De Catalunya: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/97586/8/avalenciapTFM0619memoria.pdf>

Vallejo de la Torre, C. A., Marcillo Sánchez, P. M., & Uvidia Vélez, M. V. (2018). *Sistemas de Prevención de Intrusos*. Babahoyo: CIDEPRO Editorial. doi:10.29018/978-9942-792-39-6

Venkatesh, B., & Anuradha, J. (19 de marzo de 2019). A Review of Feature Selection and Its Methods. *Cybernetics and Information Technologies*, 19(1). doi:10.2478/cait-2019-0001

Ying, X. (12 de Marzo de 2019). An Overview of Overfitting and its Solutions. *IOPscience*, 1-4. doi:10.1088/1742-6596/1168/2/022022

Yinghua, Y., Yongkang, P., & Liping, Z. (2019). Fault Monitoring Method Based on Mutual Information and Relative Principal Component Analysis. *2019 Chinese Control And Decision Conference (CCDC)*, (págs. 440-444). Nanchang. doi:10.1109/CCDC.2019.8833306

- Zamorano Ruiz, J. (5 de julio de 2018). *Comparativa y análisis de algoritmos de aprendizaje automático para la predicción del tipo predominante de cubierta arbórea*. Obtenido de E-Prints Complutense : https://eprints.ucm.es/48800/1/Memoria%20TFM%20Machine%20Learning_Juan_Zamorano_para_difundir%20%282%29.pdf
- Zhang, H., Zhang, L., & Jiang, Y. (2019). Overfitting and Underfitting Analysis for Deep Learning Based End-to-end Communication Systems. *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, (págs. 1-6). Xi'an. doi:10.1109/WCSP.2019.8927876
- Zita, A. (s.f de s.f de s.f). *Exactitud y precisión*. Obtenido de diferenciador: <https://www.diferenciador.com/diferencia-entre-exactitud-y-precision/>

BIBLIOGRAFÍA

ESET. (s.f de s.f de 2018). *ESET Security Report Latinoamérica 2018* Obtenido de welivesecurity:

[https://www.welivesecurity.com/wp-](https://www.welivesecurity.com/wp-content/uploads/2018/06/ESET_security_report_LATAM2018.pdf)

[content/uploads/2018/06/ESET_security_report_LATAM2018.pdf](https://www.welivesecurity.com/wp-content/uploads/2018/06/ESET_security_report_LATAM2018.pdf)

García Monje,R.A (10 de octubre de 2017). *Seguridad Informática y El Malware. Obtenido de*

Colombia:

[http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2641/00004128.pdf?seq](http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2641/00004128.pdf?sequence=1)

[uence=1](http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2641/00004128.pdf?sequence=1)

Guevara Maldonado C B. (s.f de s.f de 2018). *Desarrollo de algoritmos eficientes para*

identificación de usuarios en accesos informáticos. Obtenido de E Prints Complutense:

<https://eprints.ucm.es/46037/1/T39510.pdf>

Hacknoid. (20 de diciembre de s.f). *Aplicaciones de Machine Learning y la IA en Ciberseguridad.*

Obtenido de Hacknoid: [https://hacknoid.com/hacknoid/aplicaciones-del-machine-](https://hacknoid.com/hacknoid/aplicaciones-del-machine-learning-y-la-ia-en-ciberseguridad/#:~:text=Detecci%C3%B3n%20de%20amenazas,de%20que%20explote%20una%20vulnerabilidad)

[learning-y-la-ia-en-](https://hacknoid.com/hacknoid/aplicaciones-del-machine-learning-y-la-ia-en-ciberseguridad/#:~:text=Detecci%C3%B3n%20de%20amenazas,de%20que%20explote%20una%20vulnerabilidad)

[ciberseguridad/#:~:text=Detecci%C3%B3n%20de%20amenazas,de%20que%20explote%](https://hacknoid.com/hacknoid/aplicaciones-del-machine-learning-y-la-ia-en-ciberseguridad/#:~:text=Detecci%C3%B3n%20de%20amenazas,de%20que%20explote%20una%20vulnerabilidad)

[20una%20vulnerabilidad](https://hacknoid.com/hacknoid/aplicaciones-del-machine-learning-y-la-ia-en-ciberseguridad/#:~:text=Detecci%C3%B3n%20de%20amenazas,de%20que%20explote%20una%20vulnerabilidad)

Lllavarason, p., & Kamachi Sundaram, B. (2019). A Study of Intrusion Detection System using

Machine Learning Classification Algorithm based on different feature selection approach.

2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and

Cloud) (I-SMAC) (págs. 295-299). Palladam: IEEE. doi: 10.1109/I-

SMAC47947.2019.9032499

Jiménez Galindo, C. (s.f de s.f de 2009). *Diseño y Optimización de un sistema de Detección de*

Intrusos Híbrido. Obtenido de La web del programador:

<https://www.lawebdelprogramador.com/pdf/1951-Diseno-y-Optimizacion-de-un-Sistema-de-Deteccion-de-Intrusos-Hibrido.html>

Kalimuthan, C., & Arokia Renjit, J. (2020). Review on intrusion detection using feature selection with machine learning techniques. *ELSEVIER*, 33(7), 3794-3802. doi:<https://doi.org/10.1016/j.matpr.2020.06.218>

López Espinosa, J. (18 de julio de 2017). *Uso de técnicas de machine learning para la detección de fraudes en los contratos de obras públicas*. Obtenido de OLACEFS: <https://www.olacefs.com/wp-content/uploads/2019/10/Primer-Premio-Jonathan-Nabor-L%C3%B3pez-Espinoza-EFS-Chile.pdf>

Mardini, J. (18 de julio de 2017). Modelo de detección de intrusos en sistemas computacionales, realizando selección de Características con Chi Square, entrenamiento y clasificación GHSOM. *Investigación e innovación en Ingenierías*, 5(1). doi: <https://doi.org/10.17081/invinno.5.1.2614>

Rivero Pérez, J. L. (01 de diciembre de 2014). Técnicas de aprendizaje automático para la detección en redes de computadoras. *Revista Cubana de Ciencias Informáticas*, 8, 52-73. Obtenido de ResearchGate: https://www.researchgate.net/publication/273476191_Tecnicas_de_aprendizaje_automatizado_para_la_deteccion_de_intrusos_en_redes_de_computadoras

Salazar Hernández, R. (02 de febrero de 2016). *Sistemas de detección de intrusos mediante modelo de URI*. Obtenido de digibug: <https://digibug.ugr.es/bitstream/handle/10481/43353/25974403.pdf?sequence=6&isAllowed=y>

Zufiarre Soto, G. (10 de Julio de 2019). *Detección de Malware mediante Aprendizaje Profundo*.

Obtenido de Universidad del País Vasco/Euskal Herriko Unibertsitatea:

<http://hdl.handle.net/10810/36853>

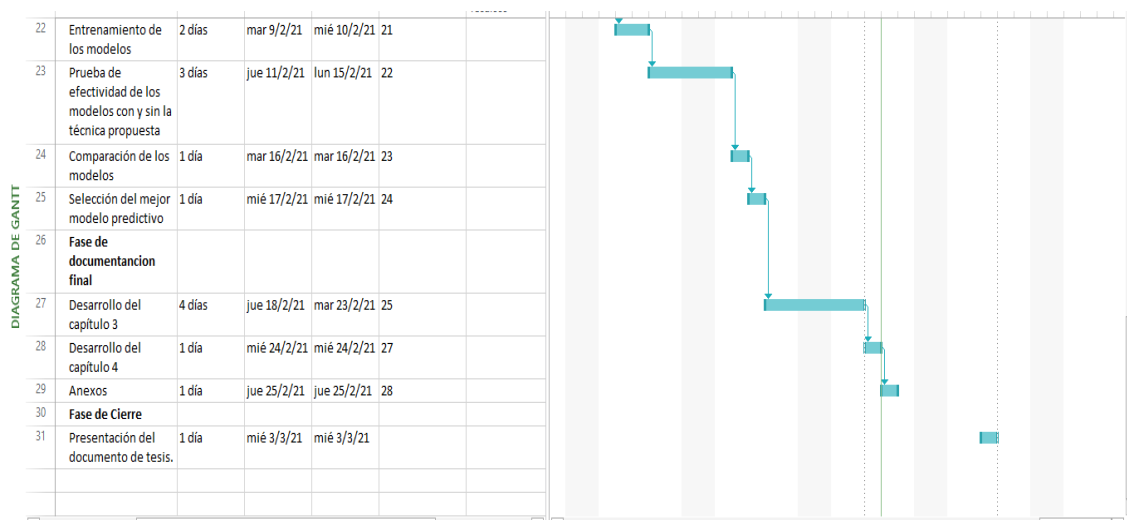
ANEXOS

En esta sección se agregan los anexos del presente trabajo de investigación. A continuación, se presenta una lista de los anexos del proyecto:

- Anexo 1. Planificación de actividades del proyecto.
- Anexo 2. Geo-localización del problema.
- Anexo 3. Fundamentación legal.
- Anexo 4. Criterios éticos a utilizarse en el desarrollo del proyecto.
- Anexo 5. Formato para la validación de expertos.
- Anexo 6. Formato de constancia de juicio de expertos.
- Anexo 7. Validación de expertos.
- Anexo 8. Acta de entrega y recepción definitiva.
- Anexo 9. Artículo científico.

Anexo 1. Planificación de actividades del proyecto

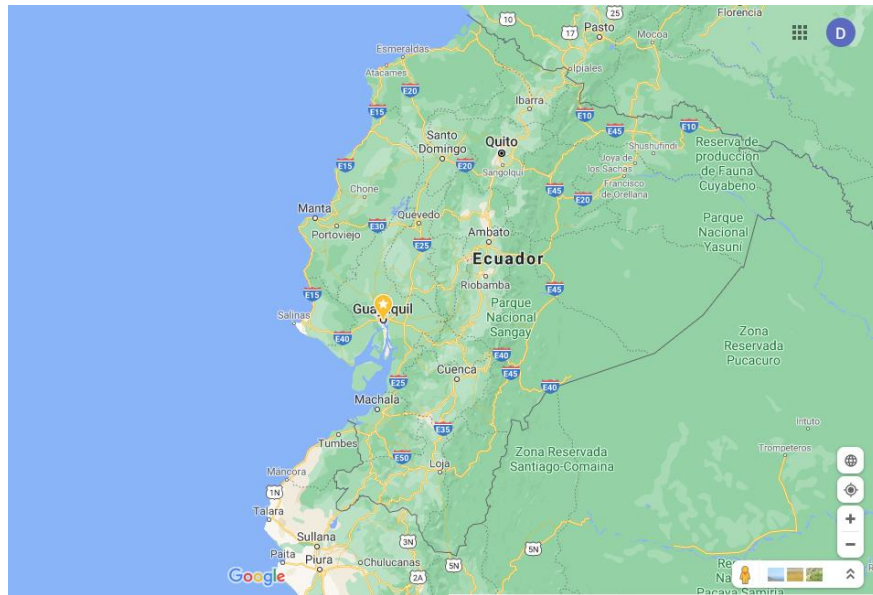




Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Anexo 2. Geo-localización del problema



Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Anexo 3. Fundamentación Legal

Las Normas Legales en un Proyecto de Titulación

Apoyo en leyes, estatutos, acuerdos, reglamentos, especialmente para proyectos especiales y factibles, debe escribir únicamente los artículos citados en la CONSTITUCIÓN DE LA REPUBLICA DEL ECUADOR; LEY ORGÁNICA DE EDUCACIÓN SUPERIOR (art. 21), REGLAMENTO DEL CONSEJO DE EDUCACIÓN SUPERIOR; LEY ORGÁNICA DE TRANSPARENCIA Y ACCESO A LA INFORMACIÓN PÚBLICA; LEY DEL SISTEMA NACIONAL DE REGISTRO DE DATOS PÚBLICOS; CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, BUEN VIVIR, etc.

El presente proyecto de titulación se fundamenta en la constitución, leyes y normas como se detalla a continuación:

ARTÍCULO DE LA LOES	CONTEXTO
¿Qué regula la LOES? ART. 1 ÁMBITO	Esta Ley regula el sistema de educación superior en el país, a los organismos e instituciones que lo integran; determina derechos, deberes y obligaciones de las personas naturales y jurídicas, y establece las respectivas sanciones por el incumplimiento de las disposiciones contenidas en la Constitución y la presente Ley ARTÍCULO 1
¿Cuál es el Objeto de esta Ley? ART. 2 OBJETO	Esta Ley tiene como objeto definir sus principios, garantizar el derecho a la educación superior de calidad que propenda a la excelencia, al acceso universal, permanencia, movilidad y egreso sin discriminación alguna.
<u>Entre las funciones</u> ART. 4 DERECHO A LA EDUCACION SUPERIOR	a) Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia; n) Garantizar la producción de pensamiento y conocimiento articulado con el pensamiento universal; y, ñ) Brindar niveles óptimos de calidad en la formación
Principio de Igualdad y Principio de Calidad	El principio de igualdad de oportunidades consiste en garantizar a todos los actores del Sistema de Educación Superior las mismas posibilidades en el acceso, permanencia, movilidad y egreso del sistema, sin discriminación de género, credo, orientación sexual, etnia, cultura, preferencia política, condición socioeconómica o discapacidad. El principio de calidad consiste en la búsqueda constante y sistemática de la excelencia, la pertinencia, producción óptima, transmisión del conocimiento y desarrollo del pensamiento mediante la autocrítica, la crítica externa y el mejoramiento permanente
ART. 87	Como requisito previo a la obtención del título, los y las estudiantes deberán acreditar servicios a la comunidad mediante prácticas o pasantías pre profesionales. debidamente monitoreadas. en los campos de su especialidad, de conformidad con los lineamientos generales definidos por el Consejo de Educación Superior.
ARTÍCULO 19.- DEL REGLAMENTO.- NÓMINA DE GRADUADOS Y NOTIFICACIÓN A LA SENESCYT	Las instituciones de educación superior notificarán obligatoriamente a la SENESCYT la nómina de los graduados y las especificaciones de los títulos que expida, en un plazo no mayor de treinta días contados a partir de la fecha de graduación. (...) este será el único medio oficial a través del cual se verificará el reconocimiento y validez del título en el Ecuador.

ARTÍCULO 144 PRINCIPIOS	Art. 144.- Tesis Digitalizadas.- Todas las instituciones de educación superior estarán obligadas a entregar las tesis que se elaboren para la obtención de títulos académicos de grado y posgrado en formato digital para ser integradas al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
------------------------------------	---

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Ley Orgánica de Educación Superior.

ARTÍCULO DE LA CONSTITUCIÓN	CONTEXTO
ARTÍCULO 22	Establece: las personas tienen derecho a desarrollar su capacidad creativa, al ejercicio digno y sostenido de las actividades culturales y artísticas, y a beneficiarse de la protección de los derechos morales y patrimoniales que les correspondan por las producciones científicas, literarias o artísticas de su autoría.
ARTÍCULO 26	La educación es un derecho de las personas a lo largo de su vida y un deber ineludible e inexcusable del Estado. Constituye un área prioritaria de la política pública y de la inversión estatal, garantía de la igualdad e inclusión social y condición indispensable para el buen vivir.
ARTÍCULO 28	La educación responderá al interés público y no estará al servicio de intereses individuales y corporativos. Se garantizará el acceso universal, permanencia, movilidad y egreso sin discriminación alguna
ARTÍCULO 350	El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo
ARTÍCULO 355 primer y segundo inciso	El Estado reconocerá a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los objetivos del régimen de desarrollo y los principios establecidos en la Constitución

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Ley Orgánica de Educación Superior.

FACTIBILIDAD LEGAL. - Comprende la viabilidad legal del proyecto, es decir, conocer los alcances y limitaciones relacionadas con el desarrollo del mismo.

- La viabilidad legal busca principalmente determinar la existencia de alguna restricción legal en la realización de un proyecto.
- Se busca determinar la existencia de normas o regulaciones legales que impidan la ejecución u operación del proyecto.
- Promover el desarrollo de proyectos sin problemas y dentro de las disposiciones legales.

- Pueden ser registrados y patentados.
- Este proyecto no transgrede ninguna norma, leyes o reglamentos establecidos en la Constitución del Ecuador ni en estamentos legales, por tanto, es factible su desarrollo y aplicación.

CODIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INVENCIÓN

Artículo 104.- Obras susceptibles de protección. - La protección reconocida por el presente Título recae sobre todas las obras literarias, artísticas y científicas, que sean originales y que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocerse. 12.- SOFTWARE

Artículo 131.- Protección de software. - El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente; es decir, en forma legible por el ser humano; o como código objeto; es decir, en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa. Se excluye de esta protección las formas estándar de desarrollo de software. En este sentido, los documentos y textos producidos en las Instituciones de Educación Superior desarrollados con el objeto de obtener sus grados académicos y/o trabajos de facultad, son autores intelectuales con el patrocinio de cada institución, por lo tanto, son acreedores a los derechos de protección intelectual dispuestos en la normativa vigente.

Adicionalmente, considere revisar las siguientes fuentes:

Artículos de la Constitución
Política vigente (Año 2018).

Se recomienda lo siguiente:

- ✓ Artículo 22
- ✓ Artículo 26
- ✓ Artículo 28
- ✓ Artículo 350
- ✓ Artículo 355 primer y segundo inciso
- ✓ Artículo 424 primer inciso

Se entiende por inciso a un
párrafo.



Artículos de la Ley Orgánica de Educación Superior:

Se recomienda lo siguiente:

- ✓ Artículo 1
- ✓ Artículo 2
- ✓ Artículo 4
- ✓ Artículo 19
- ✓ Artículo 21
- ✓ Principio de Igualdad y Principio de Calidad
- ✓ Artículo 87
- ✓ Artículo 144
- ✓ Artículo 204

En la siguiente lámina se expresan textualmente los principios de Igualdad y de Calidad.



Artículos de la Ley Orgánica de Educación Superior

El **principio de igualdad** de oportunidades consiste en garantizar a todos los actores del Sistema de Educación Superior las mismas posibilidades en el acceso, permanencia, movilidad y egreso del sistema, sin discriminación de género, credo, orientación sexual, etnia, cultura, preferencia política, condición socio económica o discapacidad.

El **principio de calidad** consiste en la búsqueda constante y sistemática de la excelencia, la pertinencia, producción óptima, transmisión del conocimiento y desarrollo del pensamiento mediante la autocritica, la crítica externa y el mejoramiento permanente.



Artículos de Código Orgánico de la economía social de los conocimientos, creatividad e invención.

Se recomienda lo siguiente:

- ✓ Artículo 104
- ✓ Artículo 131

Cualquier otra Ley o reglamento que se desee citar, debe estar alineado a lo ya establecido.



Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

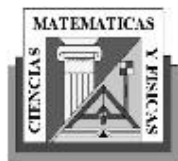
Fuente: Constitución del Ecuador (2010).

Anexo 4. Criterios éticos a utilizarse en el desarrollo del proyecto

Criterios	Características del criterio	Procedimientos
Credibilidad Valor de la verdad/autenticidad	Aproximación de los resultados de una investigación frente al fenómeno observado.	- Los resultados son reconocidos "verdaderos" por los participantes. - Observación continua y prolongada del fenómeno. - Triangulación.
Transferibilidad Aplicabilidad	Los resultados derivados de la investigación cualitativa no son generalizables sino transferibles.	- Descripción detallada del contexto y de los participantes. - Muestreo teórico. - Recogida exhaustiva de datos.
Consistencia Dependencia/replicabilidad	La complejidad de la investigación cualitativa dificulta la estabilidad de los datos. Tampoco es posible la replicabilidad del estudio.	- Triangulación - Empleo de evaluador externo. - Descripción detallada del proceso de recogida, análisis e interpretación de datos. - Reflexibilidad del investigador.
Confirmabilidad o Reflexibilidad Neutralidad / Objetividad	Los resultados de la investigación deben garantizar la veracidad de las descripciones realizadas por los participantes.	- Transcripciones textuales de las entrevistas. - Contratación de los resultados con la literatura existente. - Revisión de hallazgos por otros investigadores. - Identificación y descripción de limitaciones y alcances del investigador.
Relevancia	Permite evaluar el logro de los objetivos planteados y saber si se obtuvo un mejor conocimiento del fenómeno de estudio.	- Configuración de nuevos planteamiento teóricos o conceptuales. - Comprensión amplia del fenómeno. - Correspondencia entre la justificación y los resultados obtenidos.
Adecuación teórica-epistemológica	Correspondencia adecuada del problema por investigar y la teoría existente.	- Contratación de la pregunta con los métodos. - Ajustes de diseño.

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.



Anexo 5. Formato para la validación de expertos

Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



Datos generales

Apellidos y nombres del informante (Juez Experto):

Grado Académico:

Profesión:

Cargo:

Título: modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático						
INDICADORES DE EVALUACIÓN DEL PROYECTO DE TITULACIÓN	CRITERIOS	Deficiente	Regular	Buena	Muy Buena	Excelente
1. CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.					
2. ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.					
3. DISPONIBILIDAD	El producto cumple con estándares de disponibilidad.					
4. CONSISTENCIA	Está basado en aspectos teóricos y científicos.					
5. METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.					
6. APLICABILIDAD	El instrumento es de fácil aplicación.					
Sumatoria Parcial						
Sumatoria Total						

Anexo 6. Formato de constancia de juicio de experto

Guayaquil, marzo del 2021

Estimado(a) Ingeniero(a):

El presente instrumento certifica que se realizó la revisión del proyecto de titulación “MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO”, cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Dayannara Cindy Avila Maldonado y Joel Anthony Torres Urresto, estudiante(s) no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

Ing.
C.I. _____
Elaboración: Investigadores.
Fuente: Propia.

Anexo 7. Validación de expertos.

Juicios de expertos

Para la validación del proyecto se utilizó el instrumento de juicio de expertos con la finalidad de realizar las pruebas de funcionalidad y porcentaje de validación del software desarrollado, adicional los expertos que realicen la validación correspondiente pueda ofrecer valorización para este proyecto y que las técnicas implementadas sean las adecuadas.

ANEXO 7. VALIDACIÓN DE EXPERTOS

Experto #1: M.Sc. Jorge Luis Charco Aguirre



Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



INSTRUMENTO DE VALIDACIÓN

Datos generales

Apellidos y nombres del informante (Juez Experto):

Charco Aguirre Jorge Luis

Grado Académico:

Máster

Profesión:

Master en Inteligencia Artificial

Cargo:

Docente

Título: modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático

INDICADORES DE EVALUACIÓN DEL PROYECTO DE TITULACIÓN	CRITERIOS	Deficiente	Regular	Buena	Muy Buena	Excelente
1. CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.					X
2. ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.					X
3. DISPONIBILIDAD	El producto cumple con estándares de disponibilidad.					X
4. CONSISTENCIA	Está basado en aspectos teóricos y científicos.					X
5. METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.					X
6. APLICABILIDAD	El instrumento es de fácil aplicación.					X
Sumatoria Parcial						6
Sumatoria Total		6				

Experto #2: M.Sc. Lorenzo Cevallos Torres



Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



INSTRUMENTO DE VALIDACIÓN

Datos generales

Apellidos y nombres del informante (Juez Experto):

Lorenzo Jovanny Cevallos Torres

Grado Académico:

Master

Profesión:

Master en modelado computacional en ingeniería universidad de cádiz España

Master en gestión de productividad y calidad espol

Cargo:

Docente

Título: modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático

INDICADORES DE EVALUACIÓN DEL PROYECTO DE TITULACIÓN	CRITERIOS	Deficiente	Regular	Buena	Muy Buena	Excelente
1. CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.					X
2. ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.					X
3. DISPONIBILIDAD	El producto cumple con estándares de disponibilidad.					X
4. CONSISTENCIA	Está basado en aspectos teóricos y científicos.					X
5. METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.					X
6. APLICABILIDAD	El instrumento es de fácil aplicación.					X
Sumatoria Parcial						6
Sumatoria Total		6				

Experto #3: Ing. Bryan Cristopher Manzaba Lindao



Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



INSTRUMENTO DE VALIDACIÓN

Datos generales

Apellidos y nombres del informante (Juez Experto):

Manzaba Lindao Bryan Cristopher

Grado Académico:

Ingeniero

Profesión:

Ingeniero en sistemas computacionales

Cargo:

Desarrollador, Devops, Scrum Master

Título: modelo de detección de intrusos para detectar y evitar la inserción de malware en una red, basado en técnicas de aprendizaje automático						
INDICADORES DE EVALUACIÓN DEL PROYECTO DE TITULACIÓN	CRITERIOS	Deficiente	Regular	Buena	Muy Buena	Excelente
1. CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.					X
2. ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.					X
3. DISPONIBILIDAD	El producto cumple con estándares de disponibilidad.					X
4. CONSISTENCIA	Está basado en aspectos teóricos y científicos.					X
5. METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.					X
6. APLICABILIDAD	El instrumento es de fácil aplicación.					X
Sumatoria Parcial						6
Sumatoria Total		6				

CONSTANCIA DE JUICIO DE EXPERTO

Experto #1: MSc. Jorge Luis Charco Aguirre



Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



Guayaquil, 24 de febrero del 2021

Estimado(a) Ingeniero(a)

Jorge Charco Aguirre

El presente instrumento certifica que se realizó la revisión del proyecto de titulación "MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO", cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Dayannara Cindy Avila Maldonado y Joel Anthony Torres Urresto, estudiante(s) no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

**JORGE
LUIS
CHARCO
AGUIRRE**

Firmado
digitalmente por
JORGE LUIS
CHARCO AGUIRRE
Fecha: 2021.02.24
23:39:30 -05'00'

Ing. Jorge Charco Aguirre

C.I. 0919389692

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Experto #2: MSc. Lorenzo Cevallos Torres



Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



Guayaquil, 03 marzo del 2021

Estimado(a) Ingeniero(a) Msc

Lorenzo Jovanny Cevallos Torres

El presente instrumento certifica que se realizó la revisión del proyecto de titulación “MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO”, cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Dayannara Cindy Avila Maldonado y Joel Anthony Torres Urresto, estudiante(s) no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.



Firmado: lorc@guayaquil.edu.ec
LORENZO JOVANNY
CEVALLOS TORRES

Ing. Lorenzo Jovanny Cevallos Torres Msc

C.I. 0914517966

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Experto #3: Ing. Bryan Cristopher Manzaba Lindao

Universidad de Guayaquil
Facultad de Ciencias Matemáticas y Física
Carrera de Ingeniería en Sistemas Computacionales



Guayaquil, 3 de marzo del 2021

Estimado(a) Ingeniero(a)

Bryan Cristopher Manzaba Lindao

El presente instrumento certifica que se realizó la revisión del proyecto de titulación "MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO", cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 7, por tanto, Dayannara Cindy Avila Maldonado y Joel Anthony Torres Urresto, estudiante(s) no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, pueden continuar con el proceso de titulación en vista que no existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

Ing. Bryan Cristopher Manzaba Lindao

C.I. 0929377122

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Anexo 8. Acta de entrega y recepción definitiva

En la ciudad de Guayaquil, a ____ días del mes de _____ de ____

Por el presente documento.

Los estudiantes no titulados de la Carrera de Ingeniería en Sistemas Computacionales DAYANNARA CINDY AVILA MALDONADO con cédula de identidad N° 0929308898 y JOEL ANTHONY TORRES URRESTO con cédula de identidad N° 0958897795 hacemos la entrega del código fuente del proyecto de titulación a la Dirección de la Carrera de Ingeniería en Sistemas Computacionales en un medio magnético.

Los códigos del programa/producto que se encargaron por compromiso al estar inserto en el proceso de titulación desde fecha __ de _____.

Para efectos de dar cumplimiento a la entrega del código fuente, cedo todos los derechos de explotación sobre el programa y, en concreto, los de transformación, comunicación pública, distribución y reproducción, de forma exclusiva, con un ámbito territorial nacional.

Avila Maldonado Dayannara Cindy

Torres Urresto Joel Anthony

0929308898

Cédula de identidad N°

0958897795

Cédula de identidad N°

Elaboración: Dayannara Avila Maldonado y Joel Torres Urresto.

Fuente: Propia.

Anexo 9. Artículo científico

MODELO DE DETECCIÓN DE INTRUSOS PARA DETECTAR Y EVITAR LA INSERCIÓN DE MALWARE EN UNA RED, BASADO EN TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Joel Anthony Torres Urresto¹, Dayannara Cindy Avila Maldonado¹, Franklin Ricardo Parrales Bravo¹

¹ Carrera de Ingeniería en Sistemas Computacionales (CISC), Facultad de Ciencias Matemáticas y Físicas (FCMF), Universidad de Guayaquil (UG), Ecuador.

E-mail: joel.torresu@ug.edu.ec, dayannara.avilam@ug.edu.ec, franklin.parralesb@ug.edu.ec

Resumen – El objetivo de esta investigación es presentar un modelo de detección de intrusos que hace uso de una combinación de las técnicas filter y wrapper para la selección de características en la fase de preprocesamiento de datos. Mediante la técnica propuesta de selección de características, se obtuvieron características relevantes y no redundantes en el conjunto de datos resultante, mejorando el rendimiento del modelo clasificador. Para formular la técnica propuesta, se llevó a cabo una revisión de la literatura, encontrando las limitaciones que poseen las técnicas de selección de características existentes. Para la implementación de la técnica propuesta, se hizo uso de la librería scikit-learn, codificada en Python, la cual proporcionó los algoritmos clasificadores para el entrenamiento de los modelos. El conjunto de datos utilizado para el entrenamiento y prueba de los modelos fue obtenido de un repositorio GitHub⁸. Los algoritmos clasificadores seleccionados para el entrenamiento de los modelos fueron: 1) Random Forest, 2) Decision Tree, 3) Adaboost, 4) Gradient boosting. En base a la métrica de exactitud se seleccionó al mejor modelo de detección de intrusos, el cual fue entrenado mediante el algoritmo RandomForest. Este modelo consiguió una media del 99,42% de exactitud con la técnica de selección de características propuesta, mejorando en un 0.10% al resultado del modelo entrenado con el mismo algoritmo, pero sin el uso de la metodología propuesta. Con ello se evidencia que los modelos entrenados con la metodología propuesta proporcionan rendimientos similares a los modelos que no hacen uso de la misma, contando con la ventaja de eliminar aquellas características redundantes del conjunto de datos.

Índice de Términos - aprendizaje automático, ciberataque, filter, selección de subconjunto de características, wrapper.

INTRODUCCION

La seguridad de los sistemas informáticos o redes en ocasiones ha sido vulnerada, los métodos que usan los ciberdelincuentes para irrumpir en los sistemas cada vez afectan a miles de empresas. En el año 2017, 150 entidades (empresas, e instituciones) alrededor del mundo fueron víctimas de

ciberataques [1]. Desde visitar una página web, abrir un correo, interactuar con anuncios, puede ser el causante de este problema, provocando robo de información, y daños en los equipos informáticos. Cabe mencionar que a medida que avanza la tecnología, nuevas modalidades de vulnerar un sistema aparecen, es por eso por lo que las empresas deben implementar en su seguridad, software que cuente con técnicas que permitan la detección de actividades anormales en una red. Los IDS cumplen con esta función, ellos se encargan de implementar software que analiza el tráfico de las redes, tales como actividades inapropiadas, anómalas e incorrectas [2] permitiendo así detectar un posible ciberataque.

Una alternativa para lidiar con los ciberataques es utilizar el aprendizaje automático, debido a que aumenta la capacidad de análisis para clasificar amenazas, considerándose así una herramienta de protección [3]. El aprendizaje automático es una rama de la Inteligencia Artificial. Esta permite que las máquinas aprender de sí mismas a través del análisis de datos [4].

Existen en el mercado varios sistemas de detección de intrusos, pero esta investigación trae consigo un modelo basado en técnicas de aprendizaje automático que considera la combinación de las técnicas filter y wrapper para la selección de subconjunto de características, para solventar el problema que existe en la selección de subconjuntos de características tal como menciona Masada Tarin, las características redundantes y/o irrelevantes pueden no ser eliminados, dando lugar a los problemas de correlación entre ellas [5]. Dicha combinación permite garantizar la diversidad y nula redundancia entre las características del modelo, obteniendo un rendimiento similar a los modelos cuyas características fueron seleccionadas por las técnicas de selección de subconjuntos existentes.

CONTENIDO

2.1. Impacto de ciberataques

A medida que evoluciona la tecnología, los ciberataques se ejecutan con mayor frecuencia, ya sea en una red doméstica,

⁸ Disponible en: <https://github.com/Te-k/malware-classification/blob/master/data.csv>

PyMes o en grandes empresas. Estos ataques dirigidos por ciberdelinquentes buscan cumplir objetivos, desde dañar equipos de cómputos, colapsar las redes, hasta robo de información confidencial [6]. Los ciberataques en los últimos años se han elevado y los ciberdelinquentes han descubierto nuevos métodos para vulnerar los sistemas informáticos, así lo indica el reporte de Cisco, donde detalla que “los adversarios son cada vez más expertos en la evasión y en usar como armas los servicios de la nube y otras tecnologías utilizadas con fines legítimos” [7].

A lo largo de los años empresas alrededor del mundo han sufrido de intromisiones en sus redes informáticas, siendo estas víctimas de ciberataques. Por ejemplo, en el año 2016 se perpetró un ciberataque mientras se llevaba a cabo las elecciones para la presidencia de Estados Unidos. Este ciberataque comprometió información importante del partido demócrata y la técnica utilizada por los hackers fue mediante el método conocido como phishing [8]. Asimismo, en el año 2017 una campaña masiva de ransomware afectó a algunas empresas alrededor del mundo, ocasionando que estas empresas suspendieran sus actividades diarias [1]. En el año 2020 la red social Twitter fue víctima de un ciberataque. Los hackers intervinieron cuentas verificadas y publicaron información fraudulenta [9].

A nivel de Latinoamérica, según Izaguirre Olmedo y León Gavilanez [10] en su estudio llamado “Análisis de los Ciberataques Realizados en América Latina”, la región se ha visto envuelta en ataques de espionaje, robo de información, ataques dirigidos a las redes de computadoras e infección por malware. Estos sucesos ocurrieron durante los años 2009 hasta el 2017. Asimismo, Izaguirre Olmedo y León Gavilanez [10] mencionan que el Ecuador fue uno de los países más afectados por ciberataques mediante la aplicación de Pokémon GO en el año 2016. Ellos en su investigación detallan que los usuarios descargaron la aplicación del juego en sitios no oficiales, y el mismo no contaba con el sistema de seguridad de información adecuada [10].

Cabe mencionar que en las empresas ecuatorianas se percibe la ciber vulnerabilidad, es por ello por lo que en los últimos meses se han reportado ciberataques. Estos ciberataques son independientes, esto quiere indicar que los ciberataques afectan tanto a las empresas públicas como privadas [11].

La mayoría de las organizaciones en Ecuador mostró, a nivel individual y durante la pandemia, limitaciones propias en cuanto a ciberseguridad. Cada empresa debe diseñar procedimientos de acuerdo con estándares internacionales: buenas prácticas, metodología de ‘hacking’ ético o pruebas de penetración a los sistemas para detectar vulnerabilidad [12].

En otras palabras, las empresas ecuatorianas deben tomar en cuenta los estándares internacionales para fortalecer la ciberseguridad.

2.2. Causas y consecuencias del problema

Los ciberataques implican pérdidas económicas, de reputación y de clientes para una empresa u organizaciones. Los objetivos de un ataque informático es obtener la información que manejan las empresas; como datos personales, datos financieros de los clientes o de personas que laboran en el interior de esta y perjudicar el funcionamiento de los equipos informáticos.

Además, mantener un nivel de seguridad informático bajo, falta de capacitación al personal de las empresas, ingeniería social son las principales causas de un ciberataque. Asimismo, utilizar aplicativos que posean una debilidad, puede generar que un software malicioso se infiltre en la red.

TABLA I
CAUSAS Y CONSECUENCIAS DEL PROBLEMA

Causas	Consecuencias
Sistemas operativos y programas instalados no actualizados	Posible inserción de malware en los equipos informáticos
Falta de capacitación al personal sobre los sistemas informáticos.	Probabilidad de ser víctima de ciberataque.
Softwares deficientes para detectar amenazas.	Sistemas informáticos vulnerables.
Visitar páginas web no seguras	Posibilidad de permitir el ingreso de algún software malicioso.

2.3 Alternativa para lidiar con la problemática

Una alternativa para lidiar con los ciberataques es utilizar el aprendizaje automático. Con este se podría elaborar un modelo predictivo el cual reduciría el riesgo de sufrir un ciberataque. Aun así, esta alternativa tiene limitaciones. ESET, menciona que en el aprendizaje automático existen limitantes y una de ellas viene dada por parte del conjunto de datos. En el caso de desarrollar un modelo para detectar intrusos, previamente se debe tener los datos suficientes. Posterior a ello, los datos deben ser divididos en: maliciosos, no infectados y potencialmente no seguros o no deseados, pero este proceso no asegura que el modelo pueda identificar los nuevos datos que ingresen al mismo. Por lo que se necesita de supervisión humana y experiencia [13].

2.4. Revisión de literatura

Esta investigación aborda el tema de aprendizaje automático como alternativa para mejorar la seguridad informática. Entiéndase por seguridad informática a la disciplina que, con base en políticas y normas internas y externas de la empresa, se encarga de proteger la integridad y privacidad de la información que se encuentra almacenada en un sistema informático, contra cualquier tipo de amenazas, minimizando los riesgos tanto físicos como lógicos, a los que está expuesta. [14].

De acuerdo con Ucci et al. [15] los algoritmos más utilizados para la detección de intrusos (malware) debido a sus buenos resultados son: Decision Tree, Random Forest, Gradient Boosting, Naive Bayes, SVM, etc. Además, el estudio realizado por Valencia Peral [16] hace uso de los siguientes algoritmos: árboles de clasificación, Naive Bayes, Random Forest. Todos

estos algoritmos obtuvieron una precisión mayor del 92 %. Como conclusión, demostró que la precisión máxima la generó el modelo entrenado por árboles de clasificación con un 92.987 %. Asimismo, Rodríguez Rama [17] utilizó los algoritmos Random Tree, Random Forest, Naive Bayes, SVM, y Regresión Logística, haciendo uso del conjunto de datos KDD Cup 1999 y llegando a la conclusión de que los modelos entrenados con árboles de clasificación proporcionaron los mejores resultados. Así pues, Random Tree obtuvo una precisión del 99.9271 % mientras que Random Forest obtuvo una precisión del 99.609 %. Por otra parte, Mehmood y Rais [18] realizaron una comparación de algoritmos de aprendizaje supervisado para la detección de anomalías, utilizando el conjunto de datos kddcup.data_10_percent (KDD99_10 %) pertenecientes al programa de evaluación DARPA' 98 ID. Para sus experimentos eligieron los siguientes algoritmos: máquina de soporte (SVM), Naive Bayes y C4.5. Luego, Mehmood y Rais [18] compararon los modelos entrenados, concluyendo que la precisión del modelo entrenado con C4.5 fue muy alta con respecto a los demás. Por otro lado, el modelo entrenado con Naive Bayes fue el peor resultado de precisión.

En síntesis, los trabajos mencionados anteriormente, utilizan conjuntos de datos distintos, pero llegan a la misma conclusión, a saber: los modelos entrenados mediante el algoritmo de árboles de clasificación obtuvieron los mejores resultados.

En la literatura también se menciona la importancia de tener un buen conjunto de datos previo al entrenamiento del modelo. Como menciona ESET [13] los datos deben ser divididos en: maliciosos, no infectados y potencialmente no seguros o no deseados, pero este proceso no asegura que el modelo pueda identificar los nuevos datos que ingresen al mismo. Por ello, se necesita de supervisión humana y experiencia. Asimismo, Shao-bo [19] indica que las características redundantes e irrelevantes en un conjunto de datos ocasionan lentitud de los algoritmos y baja tasa de detección, añadiendo complejidad a su posterior tratamiento [20].

Por otro lado, en la selección de subconjunto de características, las características redundantes y/o irrelevantes pueden no ser eliminados, dando lugar a los problemas de correlación entre características [5]. Con respecto a este mismo problema, Venkatesh y Anuradha [21] menciona que las técnicas de filtrado en ocasiones no descartan las características redundantes e incompletas. Resumiendo lo planteado anteriormente, en la selección de subconjunto de características existe un problema con respecto a la redundancia en los datos, y la no eliminación de estos. Es por ello que se deben estudiar nuevas técnicas de selección de características que permitan disminuir este problema.

Entre las técnicas de selección de características se encuentran: las técnicas de filter, wrapper, embedded. La técnica filter como lo menciona Maseda Tarin [5] asigna valores para cada una de las características por medio de una función, esta las clasifica y las ordena de mayor a menor. Las características con mayor puntuación son seleccionadas para ser usadas en el método de aprendizaje a utilizar. Este método es de bajo coste computacional a diferencia de los demás [5]. Mientras que la técnica wrapper utiliza algoritmos de aprendizaje para conocer la eficacia de las características basándose en el nivel de

predicción del algoritmo. Este método elige subconjuntos de características y les asigna un valor. Como resultado se obtiene al mejor conjunto de características con base a la evaluación del algoritmo aplicado [5]. Y en última instancia, se encuentra la técnica embedded, que incluye la búsqueda del subconjunto óptimo de características. Además, el algoritmo de aprendizaje sabe que se está realizando la selección. Tiene un menor coste computacional, a diferencia de las técnicas de wrappers [22].

Conociendo las técnicas de selección de características existentes se busca solventar el problema que existe en la selección de subconjunto de características. Es por ello, por lo que en este artículo se propone una nueva técnica de selección de subconjunto de características que, combinando las técnicas filter (chi cuadrado, *f_classif*, *mutual_info_classif*) y wrapper (OLS o selección hacia adelante), busca separar características irrelevantes y redundantes para luego puntuar dichas características y elegir sólo aquellas que poseen mayor puntuación. Este proceso permitirá entrenar el modelo con un subconjunto de características relevantes y no redundantes.

MATERIALES Y METODOS

En esta sección se detalla la metodología de investigación utilizada, la cual permitió realizar el análisis de los datos y selección de características. Además, se detallan los algoritmos clasificadores utilizados y el proceso del entrenamiento del modelo predictivo. Se utilizó la revisión documental, para el desarrollo de la parte teórica del presente artículo.

3.1 Tipo de investigación

Para el desarrollo del presente artículo se seleccionó el tipo de investigación experimental, este tipo de investigación nos permite identificar un problema, desarrollar soluciones y probar cada una de ellas. Con la finalidad de someter a pruebas algunas de las soluciones y comprobar si adoptar dicha solución es factible.

3.2 Diseño metodológico de la investigación

3.2.1 Árbol de problemas

El árbol de problema ilustrado en la **Fig. 1**, se muestra las causas de los problemas encontrados y los efectos que se originan, este árbol ayuda a entender de forma precisa los problemas a resolver en el presente artículo científico.

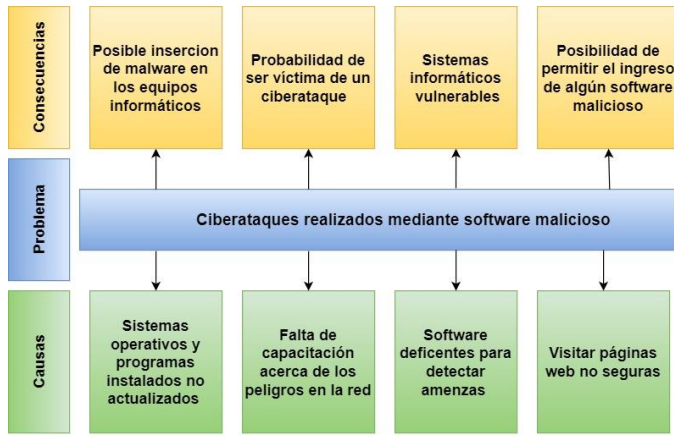


Fig. 1. Diagrama Árbol de problemas

3.2.2 Metodología de selección de subconjunto de datos

En la Fig. 2 Se detallan los pasos que se cumplieron para desarrollar la metodología propuesta.



Fig. 2. Fases de la metodología propuesta para detección de intrusos

Lectura y limpieza de datos: En esta sección se comprobó que el conjunto de datos no cuente con valores faltantes y se eliminaron columnas con información irrelevante, estas columnas contenían información como: 1) El nombre de la maquina “Name”, 2) La encriptación “Md5” y 3) La columna que identifica si el archivo es legítimo o malicioso “Legitimate”. El conjunto de datos cuenta con 54 columnas, en la **Tabla III** Se detalla el conjunto de datos usado.

TABLA III
DETALLE DEL CONJUNTO DE DATOS

Detalle	Cantidad
No. filas con registros	138.047
No. columnas con registros	57
Filas con características de datos legítimos	41.322
Filas con características de datos maliciosos	96.724

Selección de características: Se propuso una nueva metodología de selección de subconjunto de características, la cual consiste en combinar dos tipos de métodos de selección, métodos filter y wrapper. Estos métodos filter y wrapper fueron seleccionados por que son los métodos que mejor rendimiento pueden dar en el entrenamiento de modelos utilizando algoritmos de clasificación [23] [24].

Los métodos filter utilizados fueron chi cuadrado (chi2), f_classif y mutual_info_classif. Estos métodos filter necesitan recibir el valor del parámetro k que indica la cantidad de características a seleccionar. Con la finalidad de encontrar el valor óptimo de k , se evaluó cada uno de los métodos de selección filter, entrenando modelos clasificadores con los algoritmos e identificando el valor que brinde el mejor rendimiento. “RandomForestClassifier” fue el algoritmo clasificador utilizado para evaluar modelos para encontrar los valores k para los métodos filter (chi2, f_classif y mutual_info_classif). Este algoritmo mencionado fue seleccionado en base a una lista de algoritmos clasificadores utilizados en la detección de intrusos descritos en el trabajo de Ucci, Aniello, y Baldoni [15].

En la Fig. 3, Fig. 4 y Fig. 5, se presentan los diagramas que muestran los valores de k para los métodos chi cuadrado (chi2), f_classif y mutual_info_classif respectivamente.

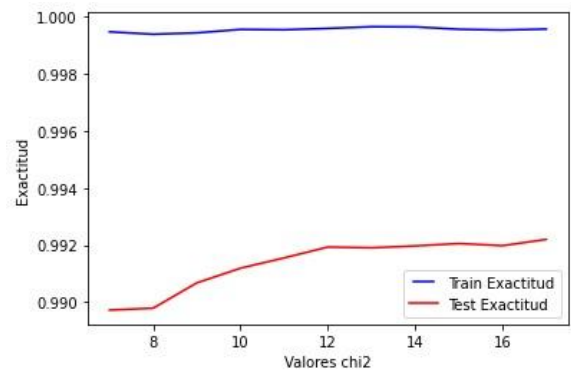
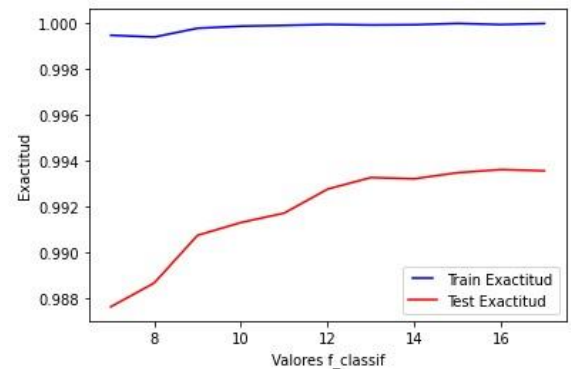
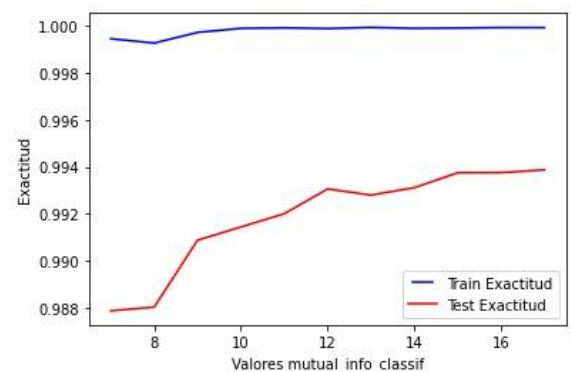
Fig. 3. Cálculo de variable k para método chi cuadrado (chi2)Fig. 4. Cálculo de variable k para método f_classif

Fig. 5. Cálculo de variable k para método mutual_info_classif

Se puede observar en la **Fig 3**, que el valor k que brinda un mejor rendimiento usando el método de selección chi2 es 14 ($k=14$), en la **Fig. 4**, se observar que el valor k más óptimo para el método de selección f_classif es 15 ($k=15$) y, por otro lado, para el método mutual_info_classif en la **Fig. 5**, se evidencia que el valor de k que permite un mejor rendimiento en el modelo es 15 ($k=15$).

Luego, cada uno de los métodos filter reciben: 1) las matrices de características “X”, y 2) la matriz objetivo “Y”. De los tres métodos filter considerados en el presente trabajo de titulación, se extrajeron 11 características, obteniéndose un subconjunto de características por cada método. Este procedimiento se puede observar en el diagrama de la **Fig. 6**.

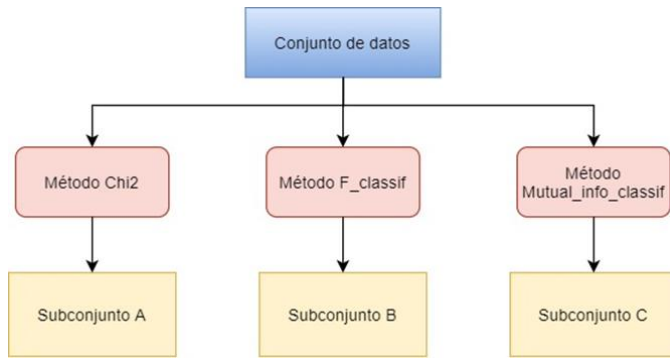


Fig. 6. Selección de características con métodos filter

Con el fin de evitar que existan muchas características relevantes en un mismo subconjunto, ya sea este el subconjunto A, B o C (ilustrados en la **Fig. 6**), se procede a distribuir uniformemente las características almacenadas en estos subconjuntos (A, B y C) en 3 grupos diferentes.

En la **Tabla IV**, se observa el pseudocódigo del algoritmo que permite almacenar uniformemente las características en 3 grupos diferentes.

TABLA IV
PSEUDOCÓDIGO PARA ALMACENAR CARACTERÍSTICAS

PSEUDOCÓDIGO DEL ALGORITMO PARA ALMACENAR LAS CARACTERÍSTICAS UNIFORMEMENTE ENTRE LOS 3 GRUPOS CREADOS
Algoritmo DistribuciónCaracteristicas index \leftarrow 0 bandera \leftarrow false contador \leftarrow false n_subconjunto \leftarrow 3 Para i \leftarrow 0 Hasta tamaño(listaClasificadores) Con Paso 1 Hacer n \leftarrow tamaño(listaClasificadores[i]) Si contador = false contador \leftarrow true Para k \leftarrow 0 Hasta n Con Paso 1 Hacer Si index < n_subconjunto Y bandera = false lista[index] \leftarrow listaClasificadores[i][k] index \leftarrow index+1 SiNo index = n_subconjunto O bandera = True Si bandera = False index \leftarrow index-1 FinSi lista[index] \leftarrow listaClasificadores[i][k] index \leftarrow index-1

```

Si index < 0
  bandera  $\leftarrow$  False
  index  $\leftarrow$  index+1
SiNo
  bandera  $\leftarrow$  True
FinSi
SiNo index < 0 Y bandera = False
  lista[index]  $\leftarrow$  listaClasificadores[i][k]
  index  $\leftarrow$  index+1
FinSi
FinPara
SiNo contador = True
  contador  $\leftarrow$  False
Para k  $\leftarrow$  n Hasta 0 Con Paso -1 Hacer
  Si index < n_subconjunto Y bandera = False
    lista[index]  $\leftarrow$  listaClasificadores[i][k]
    index  $\leftarrow$  index+1
  SiNo index = n_subconjunto O bandera = True
    Si bandera = False
      index  $\leftarrow$  index-1
    FinSi
    lista[index]  $\leftarrow$  listaClasificadores[i][k]
    index  $\leftarrow$  index-1
    Si index < 0
      bandera  $\leftarrow$  False
      index  $\leftarrow$  index+1
    SiNo
      bandera  $\leftarrow$  True
    FinSi
  SiNo index < 0 Y bandera = False
    lista[index]  $\leftarrow$  listaClasificadores[i][k]
    index  $\leftarrow$  index+1
  FinSi
FinPara
FinAlgoritmo

```

Luego de haber almacenado las características uniformemente entre los 3 subconjuntos, se procede a extraer las mejores características de cada uno de estos subconjuntos utilizando un método de selección wrapper (OLS), este proceso se puede ver ilustrado en la **Fig. 7**. Con esto se obtendrá un conjunto de características final, en dicho conjunto, se observará las características redundantes y serán eliminadas del mismo.

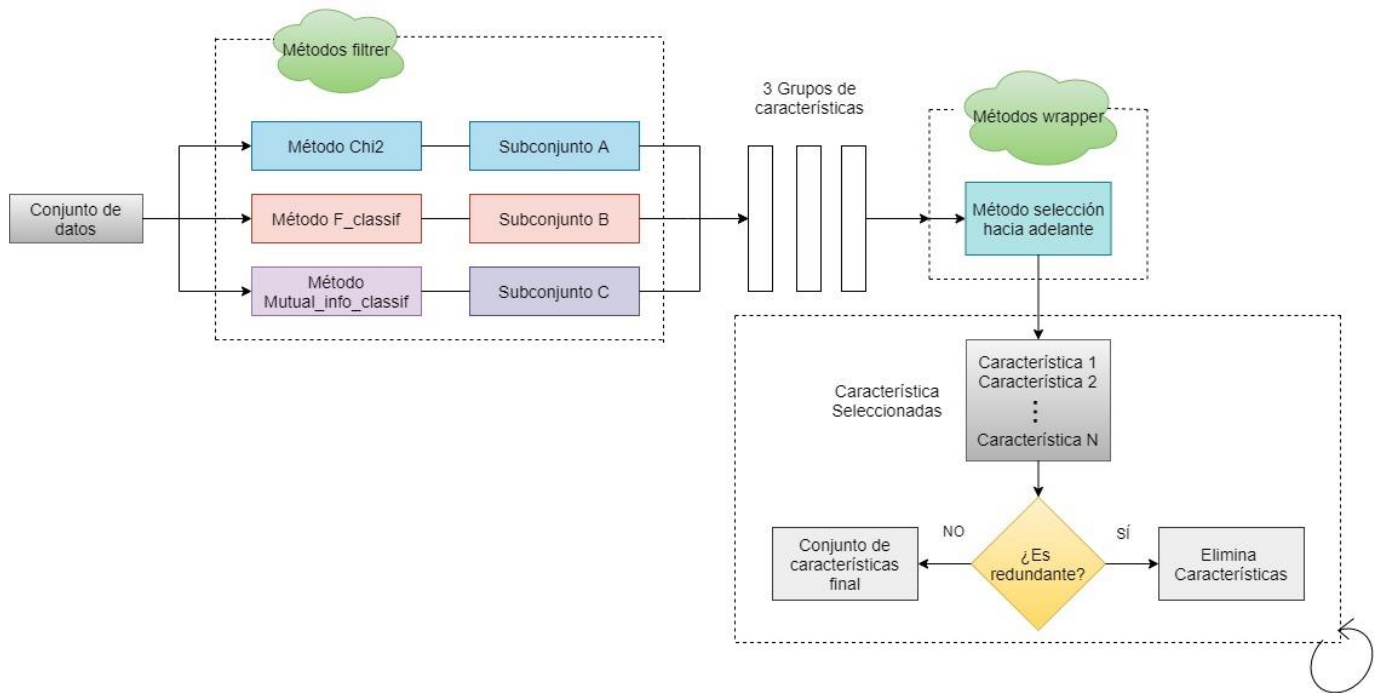


Fig. 7. Metodología propuesta para la selección de subconjunto de características

División del conjunto de datos: El conjunto de datos que contiene las características seleccionadas anteriormente es dividido en dos grupos: grupo de entrenamiento (que corresponde al 75% del conjunto de datos) y grupo de prueba (que corresponde al 25% restante del conjunto de datos).

Entrenamiento de los modelos por los algoritmos de clasificación: [15] mencionan una lista de algoritmos clasificadores y sus utilidades, una de ellas es la detección de intrusos. En base a esto, se escogió estos algoritmos porque, como indican en la literatura consultada, son los algoritmos que más se utilizan y que mejor rendimiento brindan en la detección de intrusos. En la **Tabla V**, **Tabla VI**, **Tabla VII** y **Tabla VIII** se detallan los parámetros de los algoritmos “*DecisionTreeClassifier*”, “*RandomForestClassifier*”, “*GradientBoostingClassifier*”, “*AdaBoostClassifier*” respectivamente.

TABLA V

ALGORITMO DECISIONTREECLASSIFIER Y SUS PARÁMETROS

PARÁMETRO	Descripción
criterion=” gini”	La función para medir la calidad de una división (impureza y ganancia de información).
Splitter=”best”	La estrategia utilizada para elegir la división en cada nodo.
Max_depth=25	La profundidad máxima del árbol.
Min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.
Min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
Min_weight_fraction_leaf =0.0	La fracción ponderada mínima de la suma total de pesos (de todas las muestras de

entrada) que se requiere para estar en un nodo hoja.

Max_features=12 La cantidad de características a considerar al buscar la mejor división.

TABLA VI

ALGORITMO RANDOMFORESTCLASSIFIER Y SUS PARÁMETROS

PARÁMETRO	Descripción
n_estimators=40	Cantidad de árboles en el bosque.
Criterion=” gini	La función para medir la calidad de una división (impureza y ganancia de información).
Max_depth=25	La profundidad máxima del árbol.
Min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.
Min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
Max_features=6	La cantidad de características a considerar al buscar la mejor división.

TABLA VII

ALGORITMO GRADIENTBOOSTINGCLASSIFIER Y SUS PARÁMETROS

PARÁMETRO	Descripción
loss=”deviance”	La función de pérdida a optimizar.
Learning_rate=1	La tasa de aprendizaje reduce la contribución de cada árbol.
n_estimators=40	El número de etapas de impulso a realizar.
max_depth=4	La profundidad máxima de los estimadores de regresión individuales. La profundidad máxima limita el número de nodos en el árbol.
min_samples_split=2	El número mínimo de muestras necesarias para dividir un nodo interno.

min_samples_leaf=1	El número mínimo de muestras necesarias para estar en un nodo hoja.
max_features=6	La cantidad de características a considerar al buscar la mejor división.

TABLA VIII
ALGORITMO ADABOOSTCLASSIFIER Y SUS PARÁMETROS

PARÁMETRO	Descripción
learning_rate=1	La tasa de aprendizaje reduce la contribución de cada árbol.
n_estimators=85	El número de etapas de impulso a realizar.

Evaluación del modelo: Una vez entrenados los modelos de detección de intrusos con cada uno de los algoritmos presentados en el paso anterior, se procede a comparar el rendimiento de los modelos con la finalidad de seleccionar a aquel que presente la mayor exactitud. Este proceso se lo puede observar en la **Fig. 8**.

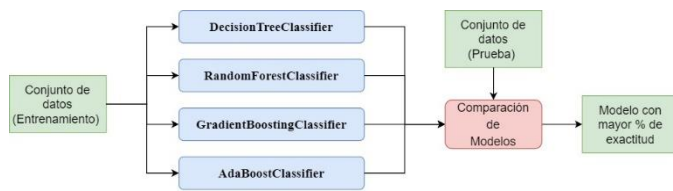


Fig. 8. Comparación de modelos clasificadores

Validación del modelo: El paso final ilustrado en la **Fig. 9**, es realizar la validación del modelo entrenado. Para ello, se desarrollaron dos archivos maliciosos. El primer archivo consiste en una puerta trasera (BackDoor) “PrimerVirus” y el segundo se trata de un virus espía (KeyLogger) “SegundoVirus”. Estos archivos maliciosos fueron creados con la herramienta msfvenom en el sistema operativo Kali Linux.



Fig. 9. Pasos de la validación del modelo

RESULTADOS

El producto resultante del presente trabajo de titulación es un modelo de detección de intrusos que ha sido construido a partir de las diversas técnicas de aprendizaje automático. Adicionalmente, se ha propuesto una metodología de selección de subconjunto de características que busca obtener diversidad y nula redundancia entre características. Dicha metodología combina los métodos filter y wrapper al distribuir uniformemente las características obtenidas por los métodos

filter en 5 grupos diferentes, y posterior a esto, seleccionar las mejores características de cada uno de estos 5 grupos utilizando un método de selección wrapper.

Con el nuevo conjunto de datos obtenido, estos se dividen en: 1) datos de entrenamiento y 2) datos de prueba. Una vez obtenida esta división, se entrena el modelo con cada uno de los algoritmos de clasificación mencionados en apartados anteriores.

Posteriormente, se ha evaluado cada uno de los modelos obtenidos mediante diversas métricas con la finalidad de seleccionar aquel que obtenga la mejor exactitud.

El modelo desarrollado fue entrenado en diez ocasiones, tanto con la metodología de selección de subconjunto de características propuesta y sin la metodología propuesta

Por cada una de estas iteraciones se obtuvieron las métricas de: Exactitud (E), sensibilidad (+), especificidad (-) y F1-score o puntaje (P). En la **Tabla IX** y **Tabla X**, se evidencia el cálculo de la media y desviación estándar de las métricas mencionadas.

TABLA IX
CON LA METODOLOGÍA PROPUESTA

	E	+	-	P
DecisionTree	99.15 ± 0.04	98.71 ± 0.11	99.35 ± 0.02	98.67 ± 0.06
RandomForest	99.43 ± 0.01	99.17 ± 0.04	99.54 ± 0.01	99.07 ± 0.01
GradientBoosting	99.07 ± 0.04	98.58 ± 0.10	99.28 ± 0.03	98.52 ± 0.05
AdaBoost	98.74 ± 0.00	97.87 ± 0.00	99.12 ± 0.00	97.92 ± 0.00

TABLA X
SIN LA METODOLOGÍA PROPUESTA

	E	+	-	P
DecisionTree	99.10 ± 0.02	98.65 ± 0.05	99.29 ± 0.02	98.54 ± 0.19
RandomForest	99.33 ± 0.01	99.02 ± 0.04	90.46 ± 28.46	99.00 ± 0.02
GradientBoosting	98.91 ± 0.07	98.24 ± 0.12	99.19 ± 0.05	98.40 ± 0.06
AdaBoost	98.56 ± 0.00	97.41 ± 0.00	99.06 ± 0.00	97.58 ± 0.00

Como se evidencia en la **Tabla IX** y **Tabla X**, mediante la aplicación de la metodología propuesta se ha obtenido una ligera mejoría en el rendimiento de los modelos entrenados por los algoritmos de clasificación. Esto se puede observar al comparar los resultados de las métricas de: exactitud, sensibilidad, especificidad y F1-score (puntaje). Así, por ejemplo, el resultado del modelo entrenado con mejor puntuación de exactitud, es decir, aquel modelo entrenado con RandomForest, tuvo una mejoría aproximadamente del 0.10% en la media de exactitud con respecto al modelo entrenado sin la metodología propuesta. Con ello, se está aportando con una nueva metodología de selección de características cuyo rendimiento es similar a los métodos existentes en la literatura.

Por otra parte, el no considerar la metodología propuesta tiene como punto débil el no garantizar la exclusión de características redundantes. En cambio, la metodología propuesta permite

separar características irrelevantes y redundantes para luego puntuar dichas características y elegir sólo aquellas que poseen mayor puntuación. Para ello, se hace uso de una combinación de técnicas wrapper y filter para mitigar la poca diversidad de características en los subconjuntos y la redundancia puede existir entre ellas, tal como lo menciona [25].

CONCLUSIONES

La presente investigación aborda la creación de una herramienta para la detección de intrusos que se encuentre a disposición de las empresas ecuatorianas y blindar sus sistemas informáticos de los ciberataques.

La metodología propuesta en este trabajo hace uso del aprendizaje automático para detectar de intrusos, entrenando el modelo clasificador con los algoritmos de mayor uso en el área y haciendo uso de un preprocesamiento previo del conjunto de datos, a saber: la selección de un subconjunto de características en la que se buscó eliminar la poca diversidad y la redundancia que existe entre ellas [23]. Para ello, se ha propuesto una combinación de las técnicas filter y wrapper que permiten separar y filtrar las características de mayor relevancia.

Según los resultados obtenidos en este trabajo, los objetivos planteados se cumplieron demostrando que con la metodología propuesta se logra una mejoría aproximadamente del 0.10% en la media de exactitud con respecto al modelo entrenado sin la selección de subconjunto de características propuesta. Con esta metodología, se está aportando una nueva técnica de selección de características cuyo rendimiento es similar a los métodos existentes en la literatura.

Además, la aplicación de la metodología propuesta de selección de subconjunto de características propuesta puede ser extrapolable a otros dominios que requieran de un aprendizaje automático, por ejemplo, para diagnóstico clínico, detección de fraude bancario, entre otros.

REFERENCIAS

- [1] BBC, "El ciberataque de escala mundial y "dimensión nunca antes vista" que afectó a instituciones y empresas de unos 150 países," 12 mayo 2017 [En línea]. Available: <https://www.bbc.com/mundo/noticias-39903218>.
- [2] C. A. Vallejo de la Torre, P. M. Marcillo Sánchez y M. V. Uvidia Vélez, *Sistemas de Prevención de Intrusos*, Babahoyo: CIDEPRO Editorial, 2018.
- [3] R. Chavez, "Aprendizaje automático: la nueva defensa contra ciberataques," 22 diciembre 2020. [En línea]. Available: <https://ideasatcloud.azurewebsites.net/aprendizaje-automatico-la-nueva-defensa-contra-ciberataques/#:~:text=Actualmente%2C%20los%20algoritmos%20de%20aprendizaje,una%20poderosa%20herramienta%20de%20protecci%C3%B3n>.
- [4] A. R. Valdez Alvarado, *Introducción al machine learning con BigML*, 2017.
- [5] M. Maseda Tarin, "Reducción de la dimensionalidad mediante métodos de selección de características en microarrays ADN," 02 enero 2019. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/89668/1/mmasedaTFG0119memoria.pdf>.
- [6] F. X. Alvear Reinoso, "Análisis y diseño de una propuesta para mitigar ataques cibernéticos a correos electrónicos utilizando técnicas de hacking ético," s.f febrero 2019. [En línea]. Available:

- <http://dspace.ups.edu.ec/handle/123456789/17035>.
- [7] Cisco, "Reporte Anual de Ciberseguridad," s.f.s.f 2018 "[En línea]. Available: https://www.cisco.com/c/dam/global/es_mx/solutions/pdf/reportes-anual-cisco-2018-espan.pdf?dtd=oemzzz000233&oid=rptsc007228&ccid=cc000153.
- [8] BBC, "Cómo fue el 'hacking' de piratas informáticos de Rusia durante las elecciones de Estados Unidos," 17 de diciembre 2016 [En línea]. Available: <https://www.bbc.com/mundo/noticias-internacional-38350244>.
- [9] . BBC, "Hacking a Twitter: por qué es importante aclarar qué hay detrás del "mayor ataque de la historia" a la red social," 17 julio 2020. [En línea]. Available: <https://www.bbc.com/mundo/noticias-53443772>.
- [10] J. Izaguirre Olmedo y F. León Gavilán, "Análisis de los ciberataques realizados en América Latina," *INNOVA Research Journal*, vol. 9, n° 3, pp. 172-181, 29 septiembre 2018.
- [11] S. M. Toapanta Toapanta, I. N. Coello Ocha, R. A. Naranjo Sanchez y L. E. Gallegos Mafla, "Impact on Administrative Processes by Cyberattacks in a Public Organization of Ecuador," de *2019 Third World Conference on Smart Trends in Systems Security and Sustainability (WorldSS4)*, London, 2019.
- [12] Diario EL COMERCIO, "Las ciberamenazas se multiplican a escala mundial por la pandemia," 15 octubre 2020. [En línea]. Available: <https://www.elcomercio.com/tendencias/ciberamenazas-ataque-informatico-seguridad-instituciones.html>.
- [13] ESET, "¿Es posible que la Inteligencia Artificial potencie el malware a futuro?," 7 noviembre 2018. [En línea]. Available: https://www.eset.com/fileadmin/ESET/LATAM/pdf/Machine_learning_WP_ES.pdf.
- [14] G. Baca Urbina, *Introducción a la Seguridad Informática*, México: Grupo Editorial Patria, 2016.
- [15] D. Ucci, L. Aniello y R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, pp. 123-147, 2019.
- [16] A. Valencia Peral, "Técnica de aprendizaje automático para la detección de ataques en el tráfico de red," 04 junio 2019. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/97586/8/avalenciaiapTFM0619memoria.pdf>.
- [17] J. M. Rodríguez Rama, "Aplicación de técnicas de Machine Learning a la detección de ataques," 4 junio 2018. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81126/1/jmrodriguez85TFM0618memoria.pdf>.
- [18] T. Mehmood y H. B. M. Rais, "Machine learning algorithms in context of intrusion detection," de *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, 2016.
- [19] D. Shao-bo, "Intrusion Feature Selection Method based on Neighborhood Distance," de *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Dalian, 2017.
- [20] Z. Mungloo Dilmohamud, G. Marigliano, Y. Jaufeerally Fakim y C. Peña Reyes, "A Comparative Study of Feature Selection Methods for Biomarker Discovery," de *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Madrid, 2018.
- [21] B. Venkatesh y J. Anuradha, "A Review of Feature Selection and Its Methods," *Cybernetics and Information Technologies*, vol. 19, n° 1, 19 marzo 2019.
- [22] R. Spasova Dimitrova, "Desarrollo y evaluación de métodos de selección de características para la predicción de eventos adversos en pacientes polimedicados," 31 mayo 2017. [En línea]. Available: <https://hdl.handle.net/2454/24594>.
- [23] scikit-learn, "sklearn.feature_selection.SeleccioneKBest," s.f.s.f.s.f. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html#sklearn.feature_selection.SelectKBest.
- [24] P. S. Salazar Casares, "Aplicación de modelos de Feature Selection y Machine Learning para identificar inhibidores potentes de la tirosinasa," 22 mayo 2019. [En línea]. Available: <http://repositorio.usfq.edu.ec/bitstream/23000/8475/1/143730.pdf>.
- [25] J. Cai, J. Luo, S. Wang y S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, pp. 70-79, 2018.