UNIVERSIDAD CUAUHTÉMOC
VANGUARDIA Y EXCELENCIA

EDUCACIÓN A DISTANCIA
UNIVERSIDAD CUAUHTÉMOC

ACUERDO NO. 1857 CON FECHA DEL 07 DE JULIO DE 2015 DEL INSTITUTO DE EDUCACIÓN DEL ESTADO DE AGUASCALIENTES

"ANÁLISIS, CLASIFICACIÓN Y PREDICCIÓN DEL VOCABULARIO DE CIBERCRIMEN EN INTERNET USANDO MODELOS PREDICTIVOS DE MACHINE LEARNING"

TESIS PARA: **MAESTRÍA EN CIENCIAS DE LOS DATOS Y PROCESAMIENTO DE DATOS MASIVOS (BIG-DATA)**

PRESENTA(N): **JOSÉ ALEXANDER CASTAÑEDA MUÑOZ**

DIRECTOR(A) DE TESIS: **DR. IVÁN CASTILLO ZÚÑIGA**

11 de Julio de 2019. Aguascalientes, México

ASUNTO: Carta de autorización.

Aguascalientes, Ags., 11 de julio de 2019.

LIC. ROGELIO MARTÍNEZ BRIONES
UNIVERSIDAD CUAUHTÉMOC PLANTEL AGUASCALIENTES
RECTOR GENERAL

P R E S E N T E

Por medio de la presente, me permito informar a Usted que he asesorado y revisado el trabajo de tesis titulado:

" ANÁLISIS, CLASIFICACIÓN Y PREDICCIÓN DEL VOCABULARIO DE CIBERCRIMEN EN INTERNET USANDO MODELOS PREDICTIVOS DE MACHINE LEARNING "

Elaborado por el Ingeniero en Sistemas Computacionales **JOSÉ ALEXANDER CASTAÑEDA MUÑOZ**, considerando que cubre los requisitos para poder ser presentado como trabajo recepcional para obtener el grado de Maestro en Ciencias de los Datos y Procesamiento de Datos Masivos (BIG-DATA).

Agradeciendo de antemano la atención que se sirva a dar la presente, quedo a sus apreciables órdenes.

ATENTAMENTE

Dr. Iván Castillo Zúñiga
Director de tesis

EDUCACIÓN A DISTANCIA

UNIVERSIDAD CUAUHTÉMOC
VANGUARDIA Y EXCELENCIA

Plantel Aguascalientes

Acuerdo No. 1857 del 8 de abril del 2015 del
Instituto de Educación del Estado de Aguascalientes

Tesis.

Para Obtener el Grado de Maestría
en Ciencias de los Datos
y Procesamiento de Datos Masivos (BIG-DATA)

Título de la Tesis.

Análisis, Clasificación y Predicción del Vocabulario de Cibercrimen en Internet Usando Modelos Predictivos de Machine Learning.

Presenta:

José Alexander Castañeda Muñoz.

Director:

Dr. Iván Castillo Zúñiga.

CDMX, México 2019.

Índice.

Índice de figuras.

Índice de tablas.

Glosario de términos.

Término.	**Descripción.**
ADVI.	Architecture (ADVI) for the detection of cyberbullying vocabulary in internet combining techniques of big data analytics and semantic.
AES.	Automated Essay Scoring (AES), Sistema de puntuación de ensayo automatizada.
Algoritmo.	Es un conjunto ordenado de instrucciones que permiten realizar cálculos y encontrar la solución de un problema específico.
Big Data Analytics.	Proceso a menudo complejo de examinar conjuntos de datos grandes y variados.
Bigram	En el ámbito de la lingüística computacional Bigram se refiere a dos palabras.
CSV.	Valores separados por coma, del inglés Comma Separated Values.

Ciberataque. Se define como una maniobra que busca algún tipo de aprovechamiento con el objetivo de tomar posesión para luego desestabilizar o deteriorar un sistema informático.

Ciberdefensa. Se puede definir como la respuesta adecuada en el ciberespacio ante amenazas o agresiones que puedan afectar a la defensa de una nación.

Ciberdelincuente. Dícese de la persona que utiliza el ordenador y las redes de comunicación para cometer delitos.

Cibercrimen. Es toda acción que a través de vías informáticas y de redes e Internet, tiene como objetivo destruir y dañar medios electrónicos.

Crawler. Programa que visita sitios Web y lee sus páginas e información para crear entradas para un índice de motores de búsqueda.

Cresceptrón. Método de aprendizaje que reconoce y segmenta patrones de imagen que son similares a los aprendidos.

Dataset. Es un conjunto de datos normalmente tabulado.

Datos.	Es la información real que constituyen los registros de distintos tipos de datos.
DIJIN.	Dirección de Investigación Criminal e INTERPOL.
ISI.	Indexación Internacional Científica.
Lematización	Es el proceso lingüístico donde, dada una forma flexionada, se encuentra el lema correspondiente.
LDA	Latent Dirichlet Allocation
Matriz de confusión.	Herramienta utilizada en Inteligencia Artificial para observar el desempeño de un algoritmo.
Minería de texto.	Proceso donde se analizan diferentes colecciones de texto con el objetivo de determinar temas y conceptos claves, incluso relaciones ocultas entre estos.
Ontología.	En el área de Informática son clasificaciones que se usan como herramienta para categorizar o agrupar la información en clases.

PIB.	Acrónimo de Producto Interno Bruto.
PLN.	Procesamiento de Lenguajes Natural.
Script.	Programa comúnmente simple, por lo general se almacena en un archivo de texto plano.
Spammer.	Individuos o empresas que envían correo no deseado.
SVM.	Abreviatura de Máquina de Soporte Vectorial.
TF.	Frecuencia de término.
TF-IDF.	Frecuencia inversa de documento.
Unigram.	En el ámbito de la lingüística computacional Unigram se refiere a una palabra.
Web Semántica.	Conjunto de actividades en el World Wide Web Consortium, que busca la creación de procesos para divulgar datos legibles.
Weka.	Plataforma de software para el aprendizaje automático y la minería de datos escrito en lenguaje Java.

Resumen.

Siendo Big Data, en una definición amplia el manejo de grandes volúmenes de datos y por consiguiente una gran cantidad de posible información; la posibilidad de poder explotar estos grandes volúmenes de datos para un bien común, es el gran motivante para este trabajo, ya que desde un punto de vista social se pretende poder analizar los datos generados por miles de sitios Web que han sido parametrizados mediante diferentes técnicas de extracción, y tras utilizar distintas técnicas de aprendizaje supervisado poder detectar léxicos o vocablos que determinen que se habla sobre Cibercrimen o temas relacionados directamente con el tema; esto con el fin de poder descubrir esta información y es aquí donde este trabajo cobra su mayor conveniencia, ya que se puede convertir en una herramienta adecuada que ayude en la detección del vocabulario referente a Cibercrimen.

Para la presente investigación los métodos con mayor efectividad en el proceso de detección de vocabulario referente a Cibercrimen fueron el método AdaBoost con un porcentaje de acierto del 95,78%; además de los métodos C 4.5 y Random Forest con una efectividad del 93,67%; por otro lado, los métodos con menos aciertos fueron el Método K-vecinos más cercanos con el 80% y el método de Naïve Bayes con un acierto del 81%, sin dejar de lado que el análisis de los datos se estructuro para palabras en el idioma español.

Palabras Claves: Cibercrimen, Aprendizaje de Máquina, Grandes Volúmenes de Datos, Aprendizaje Supervisado, Aprendizaje Automático, Detección de Vocabulario.

Abstract

Being Big Data, in a broad definition, the handling of massive volumes of data, and therefore a large amount of possible information; the possibility of being able to exploit such amount of data towards a common good, becomes the major motivator for this work, since from a social point of view, it is intended to analyze the data generated by thousands of Web sites that have been parameterized through different techniques of extraction. Afterwards, different supervised learning techniques were used to detect lexicons or words that determine if the text concerns Cybercrime or topics directly related. And this where we find its final purpose and where present research is most convenient, discovering such information, given that the outcome can be converted into an adequate tool that helps in the detection of vocabulary related to Cybercrime.

For the next research the methods most effective in the process of detecting vocabulary related to cybercrime were the AdaBoost method with a 95.78% hit rate; in addition to methods C 4.5 and Random Forest with an effectiveness of 93.67%; On the other hand the least successful methods were the K-neighbors method nearest K-NN with 80% and the Naïve Bayes method with a success of 81%, without neglecting that the analysis of the data is structured for words in the Spanish language.

Keywords: Cybercrime, Machine Learning, Big Data, Supervised Learning, Automat Learning, Vocabulary Detection.

Agradecimiento.

Mi infinita gratitud a Dios,
quien me guía y con su bendición llena siempre mi vida.

Agradezco a mi familia por estar siempre presentes
y el tiempo que me regalaron para dedicarle al manuscrito.
A mis compañeros de estudio:
Lucho, Israel y Alejandro.

Al Dr. Castillo Zúñiga por su guía oportuna y dedicada
Y por último al personal docente de la Cuauhtémoc,
por todos los conocimientos
y calidad humana que me han aportado.

Dedicatoria.

A Dios por guiarme por este camino...

A mis viejos y en especial a mi madre
quien fue mi guía y la dueña de mi amor…

A mi tía Rosa quien es mi segunda madre.

A mi familia, por ser los que enloquecen mi mundo:
Tomy, Gaby, Mate, Diana, este mundo
sería muy raro sin ustedes.

Mis amig@s, por estar siempre ahí.

Y por supuesto, a los que no están y
que no puedo ni quiero olvidar.

Introducción.

Según estudios de la consultora IDC, entre el 2005 y el 2020 el tamaño de los datos en la Web, se elevaría 300 veces, ascendiendo de 130 exabytes (un exabyte es un millón de gigabytes) a 40 mil, duplicándose anualmente la cantidad de datos digitales. Esto nos da un promedio aproximado de 5.200 gigabytes por ser humano. Lo cual genera un gran nivel de complejidad alcanzado tanto en los datos como en su análisis. Asimismo, exige diferentes herramientas y muchas de ellas nuevas que puedan tratar estos grandes datos los cuales no se puedan tratar con el software tradicional. Y es ahí donde surge el Big Data (Gabinete de comunicación UPM, 2015).

Este aumento tan significativo de los datos genera infinidad de sitios web, los cuales pueden contener miles de datos y temas diferentes. Es aquí donde se puede determinar si un sitio o varios sitios, enfocan su vocabulario a temas tan relevantes como el Cibercrimen, lo que representa gran relevancia.

La ciberdelincuencia es tan importante que ya en el 2001 el consejo europeo planteo un “Convenio sobre la ciberdelincuencia” en la ciudad de Budapest, Hungría. Donde plantea unos criterios de unidad para atacar el Cibercrimen.

Pero en Colombia no se quedan atrás; el primer documento de, "Ciberdefensa y pautas de seguridad" de 2011, se centró en contrarrestar las amenazas cibernéticas bajo los objetivos de defensa de Colombia y la lucha contra el delito cibernético; Esa necesidad

de tener cada vez más y mejores herramientas para contrarrestar a los cibercriminales es la motivación de utilizar Big Data, para ayudar a contrarrestar los ciber criminales.

Para autores como (Arcila-Calderón, Barbosa-Caro, & Cabezuelo-Lorenzo, 2016), se pueden utilizar aplicaciones de Big Data como las técnicas de aprendizaje supervisado, las cuales usan algoritmos especializados que detecten distintos patrones en los datos; este es el énfasis de la investigación poder detectar a partir de un corpus lingüístico distintos vocablos referentes a Cibercrimen.

Para poder detectar este vocabulario se realiza un proceso de rastreo de sitios Web a partir de Crawler y posteriormente aplicando la herramienta ADVI (Castillo Zuñiga, I., Luna Rosas, F., Muñoz Arteaga, J., Lopez Veyna, 2016), en la cual se depuran los datos y se generan los distintos dataset, que serán evaluados con distintas técnicas de machine learning como, Método K-vecinos más cercanos (K-NN), Naïve Bayes, Random Forest, Arboles de Decisión, Máquinas de Soporte Vectorial (SVM), Regresión Lineal, Redes Neuronales, Adaboost y C 4.5.

Después de realizar la evaluación pertinente se determina si son o no efectivas las técnicas para la detección del vocabulario, si es así cual o cuales métodos son los más adecuados para este fin; y, por último, determinar si se presenta una gran efectividad en la detección.

Si la aplicación de una o algunas técnicas de Big Data para la detección de vocabulario referente a Cibercrimen es adecuada, en un futuro podrían utilizarse para el rastreo de otras ontologías distintas a Cibercrimen.

Capítulo I. Introducción.

1.1. Planteamiento del problema.

1.1.1 Contextualización del Cibercrimen.

El primer sitio Web salió en marcha en 1991, en el 2014 habían más de 1000 millones de páginas Web lanzadas en el mundo, a marzo de 2019 se tenían más de 1670 millones de páginas Web según el portal contador de internetlivestats.com, actualmente, esta cantidad de páginas Web supera a la población actual de China, que es cercana a los 1400 millones de personas; pero que sucede con la información y el vocabulario que se maneja en este mar de datos, es ahí en donde se enfoca el problema, ya que con la evolución de los sitios y páginas Web también han aparecido los cibercriminales, los cuales se aprovechan de sus capacidades y de la dificultad para poder contrarrestarlos par parte de las autoridades. Esta tesis busca atender la problemática asociada a la necesidad de analizar y detectar dentro del innumerable volumen de datos el vocabulario referente a Cibercrimen.

En el año 2001 el consejo europeo planteo un “Convenio sobre la Ciberdelincuencia” en la ciudad de Budapest, Hungría. Donde plantea unos criterios de unidad para atacar el Cibercrimen, el cual plantea pautas para atacar a los ciberdelincuentes, pero para esto se requieren de herramientas que permitan identificar en las distintas páginas el vocabulario referente a los cibercriminales.

Según Ortega (2017), el informe del Cybercenter de la Policía colombiana, sobre amenazas de delitos informáticos en Colombia 2016-2017, los delitos informáticos cuestan al mundo $575 mil millones anuales o el 0.5 por ciento del PIB mundial. El informe señala que la cifra es cuatro veces la cantidad de todas las donaciones de ayuda internacional para el desarrollo. En América Latina y el Caribe, el costo se estima en $92 mil millones por año o el 16 por ciento del costo total del delito cibernético en todo el mundo.

Por otro lado, Ortega, (2017), también plantea que el primer documento de CONPES, "Ciberdefensa y pautas de seguridad" de 2011, se centró en contrarrestar las amenazas cibernéticas bajo los objetivos de defensa de Colombia y la lucha contra el delito cibernético.

"El nuevo documento (Arcila, 2016), 'Política nacional sobre seguridad digital', incluye la gestión de riesgos como un elemento clave para avanzar en la seguridad digital", por ende, se plantean estrategias gubernamentales que apunten a defender a los usuarios del ciberespacio de los distintos cibercriminales.

Para poder llegar a realizar estos análisis del vocabulario referente a Cibercrimen se requieren de herramientas en este caso de análisis de Big Data Analytics que permitan realizar la búsqueda de información, para esto se utilizan distintos programas o algoritmos que pueden aprender reglas a partir de datos, adaptarse a cambios y mejorar el rendimiento con la experiencia (Blum, 2003). Además, se utilizan aplicaciones de

aprendizaje supervisado las cuales requieren algoritmos especializados que detecten patrones en los datos (Arcila-Calderón et al., 2016).

El Cibercrimen es un tema actual de gran importancia e interés, ya que representa una de las más grandes amenazas que acechan a la sociedad en el mundo. El Cibercrimen está dirigido a conseguir un beneficio principalmente económico, donde la víctima es el elemento clave en la producción del evento delictivo en Internet, ya que determina su propio ámbito de riesgo al incorporar determinados bienes al ciberespacio, al interaccionar con otros y particularmente con desconocidos, y principalmente al no utilizar todas las posibles medidas de autoprotección (Miró Llinares, 2012). Sobre estos aspectos, se derivan una serie de ventajas de las que los delincuentes se aprovechan, como, el anonimato, no existen fronteras, credibilidad sobre negocios falsos, simplicidad por desconocimiento computacional, rapidez sobre transmisión de datos e inversión mínima para cometer el delito (Moise, 2014).

(Medina & Molist, 2015), señala que el Cibercrimen es un delito informático efectuado a través de operaciones ilícitas por medio de Internet; desde otra perspectiva, (Poveda Criado & Sotos Sepúlveda, 2015), mencionan que el Cibercrimen, es cualquier conducta criminal que para su realización haga uso de la tecnología informática, ya sea como método, medio o fin. Vinculado al concepto, (Sánchez Medero, 2013), menciona que dentro de los delitos más comunes del Cibercrimen, se incluye el fraude informático, el robo de información personal, la falsificación, el hacking computacional, el espionaje informático, la piratería comercial y otros crímenes contra la propiedad intelectual, la

invasión de la intimidad, la distribución de contenidos ilegales y dañinos, la incitación a la prostitución y otros crímenes contra la moralidad y el crimen organizado.

1.1.2 Definición del problema.

Actualmente la mayoría de estudios de detección de vocabulario en la web son enfocados al análisis de sentimientos y realmente no se cuentan con suficientes estudios que identifiquen en los distintos sitios web el vocabulario referente al Cibercrimen. La literatura muestra escasos estudios que apliquen técnicas de aprendizaje de máquina (machine learning) para la detección de los distintos vocablos referentes a Cibercrimen.

El problema que aborda la investigación es, “El análisis, clasificación y predicción del vocabulario de Cibercrimen a partir de datos obtenidos de páginas de Internet”.

Problemática de manera específica:

1. Dificultad para transformar la información proveniente de las páginas Web del Cibercrimen a datos estructurados (dataset), debido a que la información se encuentra mezclada con código HTML, PHP, además contiene errores de sintaxis, polimorfismo en las palabras y errores ortográficos.
2. Dificultad para obtener valor agregado en la información (que permita sustentar la toma de decisiones), debido a la complejidad de las técnicas de aprendizaje supervisado que son implementadas.

3. Dificultad para llevar a cabo la clasificación de las páginas Web, ya que es necesaria la implementan de algoritmos de aprendizaje de máquina (machine learning).
4. Dificultad para predecir el vocabulario de Cibercrimen.

1.1.3 Preguntas de investigación.

1. ¿Qué dificultades o retos reporta la literatura en la construcción de vocabularios u ontologías semánticas para la clasificación de páginas Web?
2. ¿Al comparar las técnicas de Machine learning ¿Cuál de estas técnicas de Big Data Analytics es la más apropiada para detectar vocabulario relacionado con el Cibercrimen?
3. ¿Cuál es la efectividad de las técnicas de Big Data Analytics en la predicción de temas relacionados con Cibercrimen?
4. ¿Qué dificultades presentan las técnicas de machine learning para detectar vocabulario en grandes conjuntos de datos?

1.2. Justificación.

Siendo Big Data, en una definición amplia el manejo de grandes volúmenes de datos y por consiguiente una gran cantidad de posible información; en la actualidad se plantea la necesidad de poder darle significado a esos datos. La posibilidad de poder explotar estos grandes volúmenes de datos para un bien común, es el gran motivante

para este trabajo, ya que desde un punto de vista social se pretende poder analizar los datos generados por miles de sitios Web que han sido parametrizados mediante diferentes técnicas de extracción, y tras utilizar distintas técnicas de aprendizaje supervisado poder detectar léxicos que determinen que se habla sobre Cibercrimen o temas relacionados directamente con el tema; esto con el fin de poder descubrir esta información y utilizarla para detener o atenuar el ataque de lo cibercriminales.

1.2.1. Conveniencia.

El Cibercrimen es un problema creciente de forma exponencial a nivel mundial y el análisis de vocabulario a partir de la utilización de herramientas de machine learning puede ser una gran herramienta para poder determinar sitios y ataques de parte de cibercriminales; en la actualidad cada país invierte cada vez más en defensa de ataques y detección de vulnerabilidades en sus sistemas informáticos y es aquí donde este trabajo cobra su mayor conveniencia, ya que se puede convertir en una herramienta que ayude en la detección del vocabulario referente a Cibercrimen.

1.2.2. Relevancia social en Colombia.

En Colombia se han presentado desde el año 2011 varios documentos CONPES enfocados a temas de seguridad (Planeación, 2011), y ya hacia el año 2016 se definió uno sobre seguridad digital (Consejo nacional de política económica y social, 2016), todo esto corrobora la importancia que se da cada vez más a los temas de seguridad digital,

por ende, es de gran importancia social poder contar con distintas herramientas que permitan la defensa en cuanto a ataques de seguridad digital. El análisis de vocabulario a partir de técnicas de Big Data son algunas de las herramientas que pueden ser utilizadas para estos fines.

1.2.3. Implicaciones educativas.

El poder realizar este estudio permite dejar evidencias de la aplicabilidad de las técnicas aprendizaje supervisado, dando pautas para futuros estudios y profundización en el área de análisis de vocabulario.

1.2.4. Relevancia teórica.

La importancia teórica que presenta el siguiente trabajo es la vinculación de los temas de Big Data Analytics, vocabulario en dataset, Cibercrimen y técnicas de machine learning. Estos conceptos han dado a distintas tesis de manera individual. Sin embargo, el utilizar las técnicas de aprendizaje supervisado para detectar vocabularios no es de un estudio frecuente. Aquí, se parte de un dataset generado a partir de cientos de páginas Web mediante un Crawler, posteriormente se realiza el pre-procesamiento de datos, combinando distintas técnicas, como Procesamiento de Lenguaje Natural y Web semántica, con el propósito de construir el dataset para las pruebas de predicción. Dentro de las principales aportaciones teóricas de esta investigación, se encuentran:

- Dar a conocer la aplicabilidad de distintas técnicas de aprendizaje supervisado en la detección de léxico.

- Aplicar técnicas de machine learning para detectar vocabulario.
- Proponer nuevas herramientas para la detección de vocabulario referente a Cibercrimen generándose así conocimiento colaborativo para distintas instituciones de tipo educativo, social y gubernamental.

1.2.5. Utilidad metodológica.

El presente estudio, puede ayudar a crear un nuevo instrumento para recolectar y analizar vocabulario o léxico referente al Cibercrimen, a partir de la combinación de distintas técnicas del Big Data Analytics, Web semántica y Procesamiento de Lenguaje Natural.

1.3. Objetivos.

1.3.1. Objetivo general.

Analizar, clasificar y predecir el vocabulario de Cibercrimen a partir de datos obtenidos de páginas de Internet utilizando modelos de machine learning.

1.3.2. Objetivos específicos.

1. Demostrar la utilidad de distintas técnicas de aprendizaje automático para la detección de vocabulario.

2. Clasificar páginas Web referente a Cibercrimen utilizando distintos algoritmos de aprendizaje supervisado.
3. Realizar un proceso de análisis de los datos con el fin de obtener conocimiento y valor agregado en el proceso de aprendizaje.
4. Obtener la precisión en la detección de vocabulario de Cibercrimen a través de diferentes técnicas de aprendizaje.
5. Realizar las pruebas de predicción de vocabulario usando un dataset y determinar su porcentaje de eficiencia para detectar los términos de vocabulario de Cibercrimen.

1.4. Hipótesis.

Es posible predecir vocabulario de Cibercrimen, clasificar sitios Web, y obtener valor agregado sobre las páginas que circulan en Internet, a través de técnicas de la analítica de Big Data, Procesamiento de Lenguaje Natural, Web Semántica, y Aprendizaje de Máquina (Aprendizaje Supervisado).

1.5. Breve descripción de la organización de la tesis.

La estructura de la tesis está formada por cinco capítulos y adicionalmente las referencias bibliográficas.

El capítulo 1, Introducción, tiene como objetivo proveer al lector del contexto de la investigación, resumiendo el problema de estudio atacado, las aportaciones previas en el tema investigado, definición de conceptos clave y todo aquello que contribuya al entorno mencionado al inicio del párrafo.

El capítulo 2, Estado del arte, se describen los antecedentes de la investigación, los trabajos relacionados con la problemática que aborda el proyecto actual. Se establecen las bases teóricas en las que está sustentado el trabajo de tesis, se enfoca en conceptos relacionados con Big Data, Web Semántica, Procesamiento de Lenguaje Natural, Pre-Procesamiento de Datos y Aprendizaje de Máquina. Por último, se aborda de manera puntual el tema de Cibercrimen, el cual es el caso de estudio en la investigación.

El capítulo 3, Materiales y métodos, presenta una descripción detallada de la propuesta de solución a la problemática planteada y a los objetivos establecidos al inicio de la investigación.

El capítulo 4, Pruebas y resultados, se muestran y explican los productos obtenidos tras la evaluación/validación de la propuesta de solución desarrollada.

El capítulo 5, Conclusiones, se puntualizan las ventajas y desventajas de la investigación, considerando la problemática y los objetivos establecidos al inicio del estudio. Además, se presentan las posibilidades de trabajo futuro.

Las referencias bibliográficas se muestran en formato APA en orden alfabético.

Capítulo II. Estado del arte.

2.1. Ideas, procedimientos y teorías relacionadas al problema de investigación.

2.1.1. Big Data.

El presente trabajo se encuentra enmarcado dentro de la línea de investigación **Analítica de Big Data en la Web.** El concepto de Big data se puede definir como una ciencia encargada de analizar, manipular y extraer información relevante de grandes volúmenes de datos; así lo afirman (Arcila-Calderón et al., 2016), al decir que “Big Data se refiere a volúmenes masivos y complejos de información estructurada y no estructurada que requiere de métodos computacionales para extraer conocimiento”.

El manejo de los grandes volúmenes de datos ha presentado tres cambios de mentalidad importantes que están interconectados y se refuerzan entre sí.

- El primero es la capacidad de analizar grandes cantidades de datos sobre un tema en lugar de verse obligados a conformarse con conjuntos más pequeños.
- El segundo es una habilidad de abarcar el desorden de los datos en el mundo real en lugar de la exactitud de la mayoría de los datos.
- El tercero es un respeto creciente por las correlaciones en lugar de una búsqueda continua por la causalidad difícil de alcanzar (Mayer-Schonberger & Kenneth, 2013).

Estos cambios de mentalidad son los que han llevado a Big Data a convertirse en eso, una ciencia; que permite un nuevo descubrir en el universo de la información que

cada día se encuentra en expansión. En la actualidad somos testigos activos en los avances de la investigación, técnicas de algoritmos, desarrollos en la computación de gran nivel, minería de datos, minería de textos, entre otras, que hacen cada vez más interesante e importante el manejo de los datos.

El manejo de grandes datos sólo existe gracias a la gran cantidad de información con que se cuenta en la actualidad y es por esto que los modelos de almacenamiento y la evolución de estos influyen en gran medida en su aprovechabilidad.

Se ha evolucionado entre distintos tipos de almacenamiento, desde el almacenamiento local, al almacenamiento agrupado, almacenamiento distribuido y basado en la nube. Además, los sistemas de bases de datos han sido migrado de RDBMS tradicional a los sistemas más actuales basados en No SQL (Wu, Sakr, & Zhu, 2017).

Por otro lado, los modelos de programación de Big Data representan el estilo de programación y presentan las interfaces para que los desarrolladores escriban programas y aplicaciones de Big Data.

Un modelo de programación es el estilo primordial y plantea las interfaces para que los desarrolladores escriban programas y aplicaciones informáticas. En la programación de Big Data, los usuarios se centran en escribir programas paralelos controlados por datos que pueden ejecutarse en entornos distribuidos y de gran escala.

(Serrano-cobos, 2014), define la Analítica Web como: El proceso de analizar los datos de la interacción de los usuarios, acompañándose de gráficos para visualizar o resumir esas interacciones, normalmente en forma de series temporales, gráficos de barras, rankings de contenidos o flujos de navegación. Y dentro de la analítica web el documento se enmarca en el análisis, clasificación y predicción del vocabulario de Cibercrimen en Internet.

Teniendo en cuenta que cada día se navega en millones de sitios web, el poder aplicar distintas técnicas para analizar vocabulario es un gran reto; para esto se requiere de poder aplicar distintas **técnicas de minería de textos**. Lo cual se define como la acción de extraer patrones interesantes de varios textos. Donde aplicando distintos métodos de minería se pueden analizar los contenidos y la estructura de los textos (Moohebat, Raj, Kareem, & Thorleuchter, 2015).

Es importante tener en cuenta como se relacionan la minería de textos con la minería de grandes datos.

- En la minería de texto, los datos textuales se proporcionan como fuente en el formato uniforme, mientras que, en la minería de datos grandes, se pueden proporcionar varios formatos de elementos de datos, incluidos los formatos desconocidos.
- En la minería de texto, se supone que los elementos de datos se fijan o actualizan con poca frecuencia, mientras que en la minería de datos se supone que se actualizan con mucha frecuencia o constantemente.

- En la minería de texto, los elementos de datos se recopilan a través de Internet, mientras que, en la minería de datos, se recopilan mediante medios omnipresentes, como sensores, etiquetas RFID y teléfonos móviles. Incluso si los mensajes que se transfieren entre teléfonos móviles pertenecen a Big Data, se convierten en la fuente de minería de textos, así como textos proporcionados por internet (Jo, 2019).

En el año 2013 (Uccelli, Dobbs, & Scott, 2013), aplicaron técnicas de regresión utilizadas en minería de datos para el análisis de ensayos logrando descubrir que las calidades de la escritura académica dependen significativamente de los marcadores léxico-gramaticales y organizativos.

Al igual que estos autores han utilizado distintas técnicas de minería de texto para identificar textos o vocablos dentro de un grupo determinado, la idea central de esta investigación es aplicar distintas técnicas de machine learning o aprendizaje supervisado para lograr identificar vocablos o textos con referencia a Cibercrimen.

2.1.2. Aprendizaje Supervisado.

Para realizar estos trabajos de minería de texto se aplican distintos métodos de aprendizaje de máquina**,** donde se aplican técnicas similares al aprendizaje humano a partir de experiencias de los distintos programas que aprenden del comportamiento de los patrones.

Así lo explican (Moohebat et al., 2015), en general, un conjunto de datos se divide en dos conjuntos: un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento se usa para entrenar el sistema, y el conjunto de pruebas se usa para evaluar la precisión del modelo entrenado generado. Un procedimiento de validación cruzada intercambia los conjuntos de prueba y entrenamiento, que es útil para el aprendizaje automático porque evita el sobreajuste.

2.1.3. Técnicas de Aprendizaje de Máquina (Machine Learning).

Para realizar la clasificación y análisis de los textos coexisten varios modos de realizarlos a partir de algoritmos de clasificación supervisados conocidos y básicos, los cuales incluyen, los métodos de Máquinas de Soporte Vectorial (SVM), K-Vecinos más cercanos, Arboles de Decisión, Bosques Aleatorios y Naïve Bayes, entre otros.

En el año 2014 (Peng & Zhong, 2014), propusieron el uso de distintas técnicas de análisis de sentimientos para la detección de revisión de spam. Ellos extrajeron 5000 comentarios de Resellerrating.com, posteriormente construyeron un léxico de sentimiento combinado con un léxico de sentimiento general y un léxico de sentimiento especial para el producto. Luego propusieron un método para calcular la puntuación del sentimiento.

Utilizando el análisis de texto de lenguaje natural realizaron un análisis de dependencia superficial. Utilizando además reglas discriminativas combinándolas con el método de series de tiempo para descubrir tiendas sospechosas.

Otro panorama donde se han utilizado técnicas de machine learning, es en el proceso de compras; muchas veces los usuarios revisan los comentarios de los compradores para poder determinar si las revisiones son veraces, y esto es primordial para el consumidor, para no dejarse engañar por recomendaciones falsas.

Desafortunadamente, a menudo es difícil, o imposible, para los humanos determinar la validez de una revisión a través de la lectura del texto; sin embargo, diferentes estudios han demostrado que los métodos de aprendizaje automático son efectivos para detectar revisiones falsas (Heredia, Khoshgoftaar, Prusa, & Crawford, 2016).

(Heredia et al., 2016), propusieron una técnica de conjunto que combina múltiples métodos de aprendizaje para detectar revisiones de spam. Los investigadores compararon los resultados de Naïve Bayes, C4.5, Logistic Regression, Support Vector Machine, Random Forest y Boosting and Bagging. Descubrieron que ninguna de las técnicas ensayadas fue capaz de mejorar significativamente la detección de spam en comparación con el estándar Naïve Bayes.

También, los desarrollos realizados en el uso de las técnicas de machine learning han permitido la investigación y construcción de algoritmos como el de (Mani, Kumari, Jain, & Kumar, 2018), quienes propusieron un algoritmo para detectar las revisiones falsas o spam de los comentarios reales: Dado que el trabajo propuesto se concentra solo en el texto, se utilizaron las funciones (unigram + bigram). En la primera fase de

análisis, se utilizan tres algoritmos de clasificación SVM, Naïve Bayes y Random Forest para clasificar las revisiones como spam o no spam, indicando si una revisión es genuina o falsa con respecto a un producto. Esto, de hecho, es útil para los consumidores que tienen la intención de tomar decisiones con respecto a la compra de un producto basándose en las revisiones. Naïve Bayes dio los resultados más precisos entre los tres algoritmos de clasificación al alcanzar el 87.12% de precisión.

Por otro lado, los avances en técnicas de aprendizaje han llevado a que se desarrollen nuevas técnicas como la **Deep Learning.** La diferencia entre Machine Learning y Deep Learning es que el Deep Learning tiene más flexibilidad, ya que tiene grupos de neuronas más especializadas que resolverían problemas concretos (Merchán Macías, 2018).

(Chollet 2018), en su libro Deep Learning with Python define el aprendizaje profundo “Deep learning” como: Una nueva visión del aprendizaje de representaciones a partir de datos que pone énfasis en el aprendizaje de capas sucesivas. El aprendizaje profundo es un marco matemático para aprender representaciones a partir de datos. El aprendizaje profundo ha alcanzado un nivel de atención pública e inversión en la industria nunca antes visto en la historia de la Inteligencia Artificial, pero no es la primera forma exitosa de aprendizaje automático. Es seguro decir que la mayoría de los algoritmos de aprendizaje automático utilizados en la industria hoy en día no son algoritmos de aprendizaje profundo.

2.1.4. Cibercrimen.

Autores como (Sameera & Vishwakarna, 2017), han realizado investigaciones para la detección de cibercriminales a partir de los vocabularios planteados en chats, para ello, utilizaron un algoritmo de minería de texto para verificar continuamente si hay palabras sospechosas, incluso si están en forma de palabras de código o formas cortas.

En la misma línea de investigación (Alami & Elbeqqali, 2015), utilizaron técnicas de minería de textos, para calcular una distancia de similitud para detectar publicaciones sospechosas en micro blogs, siendo esto una forma efectiva de analizar los datos publicados en Internet.

Un caso muy sonado de Cibercrimen en el año 2008, fue el del Dr. Bruce Ivins, investigador de biodefensa en el Instituto de Investigación Médica del Ejército de Enfermedades Infecciosas del Ejército de EE. UU., fue sospechoso de ser enviado por correo cartas contaminadas con ántrax que causaron 5 muertes y lesiones a docenas de personas. Ivins utilizó cuentas de correo electrónico desechables con nombres falsos. Durante el tiempo de los ataques de ántrax. Aunque Ivins se suicidó antes de ser acusado, él fue el principal sospechoso en estos ataques de ántrax en 2001 (Arredondo, 2008).

Otro ejemplo de Cibercrimen es el de Higinio Ochoa, quien obtuvo acceso no autorizado a la base de datos de usuarios completa de un departamento de policía que contenía nombres de usuario, contraseñas e información personal de los empleados de

la ley en la agencia. Ochoa también publicó comentarios burlones en Twitter sobre la intrusión. En una de las publicaciones, una fotografía de una mujer tomada desde el cuello hacia abajo, en una blusa con un letrero en su falda se convirtió en una burla obvia para los investigadores. La foto en los metadatos incorporados incluía información de geolocalización, apuntando a una dirección en Australia. Una investigación posterior encontró una publicación en Internet con el nombre de usuario «w0rmer» en un sitio web de programación de software. La publicación fue firmada «Higinio Ochoa AkA w0rmer». Esto permitió su captura (Shavers, 2013).

Por otro lado, en el ámbito colombiano, cabe señalar que en el año 2015 (Cerón Guzmán & León, 2015), realizaron una investigación destinada a la detección de spammers en Twitter. El estudio estuvo centrado en detectar cuentas maliciosas que tenían como objetivo difundir el spam en un contexto político a partir del uso de distintas técnicas de aprendizaje supervisado y no supervisado.

El Cibercrimen es una amenaza cada vez más seria para la seguridad en Colombia. Durante 2017, según cifras de la DIJÍN, no sólo aumentaron este tipo de delitos un 28%, también aparecieron nuevas amenazas para la seguridad cibernética en el país que no sólo atacan el bolsillo y la privacidad de los ciudadanos, sino también atenta contra su vida. Durante este año, la Policía bloqueó 3891 páginas por alojar contenidos de pornografía infantil y capturó a 56 personas por este delito.

El otro gran objetivo de los cibercriminales sigue siendo la estafa. Durante 2007, 6372 ciudadanos colombianos reportaron haber sido estafados por Internet, por un valor que sumado supera los 15000 millones de pesos. La pericia de los cibercriminales ha escalado tanto que ni siquiera las grandes compañías, ni el estado colombiano están a salvo de sus ataques. La Policía reportó que las arcas públicas perdieron al menos 50000 millones de pesos, unos 16,5 millones de dólares en especial por causa de accesos abusivos a las cuentas de distintas alcaldías por todo el país.

Una modalidad que va en aumento, y que afecta a las empresas privadas, sobre todo, es la suplantación de los correos corporativos, por lo que se registraron pérdidas de 380 millones de pesos durante 2017.

Colombia tampoco estuvo libre de los grandes ataques cibernéticos a nivel mundial, la Policía atendió a 52 víctimas de estos ataques globales y generó 59 alertas por posibles amenazas internacionales (“Cibercrimen en Colombia: balance de 2017,” n.d.).

Todo este panorama valida la necesidad de poder contar con la mayor cantidad de herramientas que ayuden en gran medida a atacar el Cibercrimen desde los distintos frentes posibles; es aquí donde el uso de las técnicas de machine learning, el Big Data y el análisis de datos hacen parte de las herramientas emergentes que cada vez, son más utilizadas por las autoridades a nivel mundial para contrarrestar los ataques de las distintas formas de crimen cibernético.

2.2. Descripción de trabajos relacionados.

2.2.1. Identifying ISI-indexed articles by their lexical usage: A text analysis approach.

Moohebat et al., (2015), plantean en su investigación una arquitectura para investigar la existencia de probables divergencias léxicas entre los artículos, categorizados como: ISI "ISI Web of Science" y No ISI. Sobre la base de una colección de artículos indexados en las áreas de negocios e informática, se capacitan tres modelos de clasificación Naïve Bayes, K-NN y SVM. Luego de realizar el análisis con estos modelos se encontraron los siguientes resultados: La mejor recuperación se obtiene utilizando el algoritmo SVM, con un 94% de recuperación para las revistas indexadas por el ISI en ciencias de la computación, y curiosamente, la recuperación más baja la obtiene SVM en la clasificación de artículos de negocios que no son ISI. Sin embargo, el resultado de SVM para la clasificación de los artículos de ISI tanto en informática como en negocios es determinando el F-Score. El "F - Score es el medio armónico de precisión y recuperación" (Yang, Lin, & Wu, 2009). La precisión planteada por este método fue para el negocio del 71,4% y para computadora fue del 78,9%, utilizando el algoritmo K-NN fue para el negocio del 59.5% y en computadora de 74.1%, por último el artículo con Naïve Bayes con una precisión para el negocio con el 69,4% y en computadora de 67.2% (Moohebat et al., 2015).

2.2.2. An empirical analysis of machine learning models for automated essay grading.

(Madala, Gangal, Krishna, Goyal, & Sureka, 2018), presentan un estudio para realizar un análisis lingüístico de un ensayo, posteriormente se estima la habilidad de escritura o la calidad del ensayo en forma de puntaje numérico o calificación de letra. Los sistemas AES son útiles para la escuela, la universidad y la comunidad de empresas de pruebas para escalar de manera eficiente y efectiva la tarea de calificar una gran cantidad de ensayos. Los resultados indican que el uso apropiado del vocabulario, la relevancia de los términos en el ensayo con el tema dado y la coherencia entre las oraciones y los párrafos, son buenos predictores de la puntuación del ensayo. Los análisis realizados revelan que no todas las características son igualmente importantes y pocas características son más relevantes y están mejor correlacionadas con respecto a la clase objetivo. Los investigadores realizaron experimentos con K-Vecino más cercano, Regresión logística y clasificadores basados en Máquinas de Soporte Vectorial. Los resultados en 4075 ensayos en varios temas obtuvieron una precisión del 73% a 93%.

2.2.3. Aplicación de técnicas de machine learning a la detección de ataques.

El proyecto planteado por Rodríguez Rama (2018), muestra un modelo predictivo, usando primero la plataforma Weka y desarrollando posteriormente un script escrito en lenguaje Python que permite realizar el tratamiento del dataset para la detección de

conexiones maliciosas, concretamente utilizando la librería de software Scikit-Learn. El dataset utilizado es «KDD Cup 1999» que incluye una amplia variedad de intrusiones de red simuladas en un entorno de red militar. Dicho dataset fue usado para entrenar, probar y ajustar el modelo seleccionado dentro de Random Tree, J48, Random Forest, Naïve Bayes, SVM, donde todos los algoritmos obtienen resultados alrededor del 99% de precisión, excepto Naïve Bayes con el que obtenemos solamente un 93%. Los árboles de decisión muestran los mejores resultados.

2.2.4. A Text-based Deception Detection Model for Cybercrime.

(Mbaziira & Jones, 2016), plantean que los incidentes que tienen que ver con el ciberdelito donde se explota el discurso para engañar a las personas utilizando textos, están aumentando debido a la popularidad de los mensajes de texto. Ellos utilizaron el aprendizaje automático y los enfoques lingüísticos para detectar el engaño en los mensajes de texto en redes de ciberdelincuentes. Desarrollaron distintos modelos para detectar ciberdelitos por género Web. Sus aportaciones incluyen modelos capaces de detectar estafas en las redes sociales a partir de los mensajes con un 60% de exactitud en la predicción; los modelos entrenados sobre fraude en el correo electrónico pudieron predecir las estafas en un 50% de exactitud. Además, concluyen que la predicción para el modelo de correo electrónico es prometedora debido a las variaciones lingüísticas de los ciberdelincuentes. También demostraron que los modelos de detección de ciberdelito pueden construirse usando características del procesamiento del lenguaje natural y procesos psicológicos lingüísticos vinculados al delito cibernético.

2.2.5. Ancert: aplicación de técnicas de machine Learning a la seguridad.

En este trabajo el autor Merchán Macías (2018), implementa los algoritmos de predicción C5.0 y Redes neuronales aplicados a temas de seguridad informática. El proyecto mencionado, se enfoca en el uso de estas técnicas de machine learning para realizar un estudio y comparativa de las capacidades predictivas de los algoritmos elegidos al usar el dataset “KDD CUP 1999” Data, que fue utilizado para el tercer concurso internacional de herramientas para el descubrimiento de conocimientos y la minería de datos. (Merchán Macías, 2018), desarrollo un sistema de detección de intrusiones.

Este sistema utilizo cada uno de los algoritmos para mostrar un nivel de predicción al analizar el dataset “KDD CUP 1999”, y así determinar si existen o no intrusiones y ataques. Merchán consiguió un alto grado de detección con los algoritmos; con el método C5.0 fue del 96,36% y con el SVM del 96,2% en su cuarta ejecución.

2.2.6. Analyzing topics and authors in chat logs for crime investigation.

En este documento los autores (Basher & Fung, 2014), presentan el desarrollo de un modelo de búsqueda de temas para analizar los archivos de los registros de chat para separar los registros relevantes para detectar el crimen. Específicamente, proponen una extensión de la asignación basada en el modelo Dirichlet para extraer temas, calcular la

contribución de los autores en estos temas y estudiar las transiciones de estos temas a lo largo del tiempo. Además, presentan un modelo especial para caracterizar temas de autores a lo largo del tiempo. Esto es crucial para la investigación porque proporciona una visión de la actividad en la que los autores se involucran en ciertos temas. Los experimentos en dos conjuntos de datos de la vida real sugieren que el enfoque propuesto puede descubrir temas delictivos ocultos y la distribución de los autores a estos temas.

2.2.7. Detecting Social Spammers in Colombia 2014 Presidential Election.

Los autores (Cerón Guzmán & León, 2015), plantean que la gran cantidad de contenido generado por el usuario ha convertido las redes sociales en una fuente atractiva de información para comprender el comportamiento social. Alrededor del tiempo de las elecciones, los datos de Twitter se han utilizado para medir la opinión pública en temas como la predicción de resultados, la intención de votar o la alineación política. Sin embargo, el efecto de la proliferación de nuevas formas de spam en las redes sociales en este tipo de mediciones no ha sido completamente reconocido y abordado en la investigación. En el documento, se centraron en la detección de cuentas maliciosas en Twitter, cuyo objetivo es difundir el correo no deseado en un proceso electoral. Para lograr esto, ellos recopilaron un conjunto de datos de 149000 usuarios que se referían a la elección presidencial de Colombia 2014, y se rastrearon 1.7 millones de tweets y 341000 URL en la línea de tiempo. Para distinguir las cuentas maliciosas de las que no son spammer en el conjunto de datos, se implementaron varias técnicas de aprendizaje

automático como lo fueron SVM, Random Forest y Naïve Bayes en una colección etiquetada de usuarios, que se clasificaron semiautomáticamente en spammer y no spammer.

2.3. Análisis de trabajos relacionados.

La Tabla 1, muestra un análisis de las características sobre la descripción de trabajos relacionados mencionados en los objetivos anteriores, los cuales son usados como punto de comparación con la investigación actual.

Tabla 1. Características de trabajos relacionados para la detección de Cibercrimen

Trabajos Relacionados	Orientación del estudio	Técnicas para detección de cibercrimen																		
			Conjunto de datos (dataset)						Aprendizaje supervisado											Herramientas de Mineria de datos
		Fuente de datos (Corpus)	Fuente de datos (Dataset)	TF	TF - IDF	Stop Word	Sinonimos	Crawler	Naïve Bayes	SVM	Árboles	K-NN	Regresión Lineal	J48	Redes Neuronales	AdaBoost	C 4.5.	C5.0	Random Forest	
Moohebat (2015)	Vocabulario	Revistas en inglés	Revistas en inglés		↑	↑			↑	↑		↑								Propio
Madala (2018)	Calificación de ensayos	ASAF	4000 Ensayos							↑		↑	↑							Weka
Rodríguez (2018)	VocabularioDetección ataques	IDS DARPA'98,	KDD99 Cup			↑			↑	↑	↑	↑	↑	↑					↑	Weka
Mbaziira (2016)	Vocabulario ciberdelito	Facebook y correo	PrivateRecovery, Enron						↑	↑		↑								Weka
Merchán (2018)	VocabularioDetección ataques	KDD99 Cup	KDD99 Cup												↑			↑		Weka
Basher (2014)	Vocabulario Crimen	Mensajes de Yahoo, AOL	Mensajes de Yahoo, AOL			↑														Desarrollo Propio
Investigación Actual	Vocabulario Cibercrimen	Páginas Web	Páginas Web	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		↑	↑	↑		↑	Desarrollo Propio

Capítulo III. Materiales y métodos.

3.1. Fundamento del dataset.

El corpus lingüístico se administra a través de ontologías semánticas formadas con metadatos, donde la clase representa el tema principal, la subclase a los subtemas y los atributos a cada palabra que hace referencia a cada característica, como se aprecia en la Figura. 1.

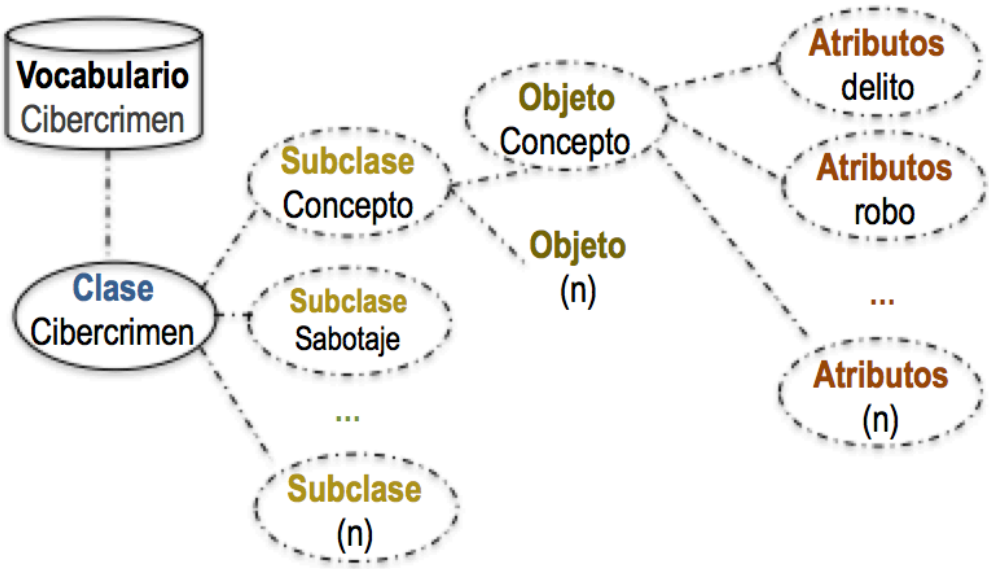


Figura 1. Modelo de ontologías orientado a objetos.

En esta tesis se consideró el corpus lingüístico de Cibercrimen, el cual se describe en la Tabla 2. Es importante mencionar que el corpus lingüístico es establecido para el idioma español.

Tabla 2. Corpus lingüístico de Cibercrimen

No.	Término	No.	Término	No.	Término	No.	Término	No.	Término
1.	Culpable.	23.	Abuso.	45.	Internet.	67.	Malware.	89.	Sabotaje.
2.	Delincuente.	24.	Daño.	46.	Red.	68.	Crakeo.	90.	Fraude.
3.	Hackers.	25.	Robo.	47.	Correo.	69.	Manipular.	91.	Falsificación.
4.	Hackeo.	26.	Hurto.	48.	Social.	70.	Modificar.	92.	Destruir.
5.	Hackear.	27.	Chantaje.	49.	Web.	71.	Pishing.	93.	Cibercrimen.
6.	Sujeto.	28.	Amenaza.	50.	Tablet.	72.	Transacción.	94.	Ciberdelito.
7.	Persona.	29	Ataque.	51.	Smartphone.	73.	Bancaria.	95.	Ciberbullying.
8.	Individuo.	30.	Perdida.	52.	Laptop.	74.	Pornografía.	96.	Ilícito.
9.	Habilidad.	31.	Estafa.	53.	Software.	75.	Prostitución.	97.	Crimen.
10.	Experto.	32.	Riesgo.	54.	Sistemas.	76.	Infantil.	98.	Criminalidad.
11.	Especialista.	33.	TIC	55.	Programar.	77.	Clonación.	99.	Delito.
12.	Cibercriminal.	34.	Tecnología.	56.	Windows.	78.	Tarjeta.	100.	Legal.
13.	Dinero.	35.	Comunicación.	57.	Linux.	79.	Cuenta.	101.	Política.
14.	Efectivo.	36.	Información.	58.	Mac.	80.	Crédito.	102.	Ley.
15.	Capital.	37.	Virtual.	59.	Digital.	81.	Magnético.	103.	Penal.
16.	Economía.	38.	Informática.	60.	Online.	82.	Cajero.	104.	Sanción.
17.	Víctima.	39.	Dispositivo.	61.	Telemática.	83.	Automático.	105.	Protección.
18.	Gobierno.	40.	Electrónico.	62.	Máquina.	84.	Piratería.	106.	Prevención.
19.	Empresa.	41.	Computador.	63.	Malicioso.	85.	Copia.	107.	Criminal.
20.	Conocimiento.	42.	Ordenador.	64.	Troyano.	86.	Película.		
21.	Injuria.	43.	Computarizado.	65.	Virus.	87.	Música.		
22.	Difamar.	44.	Nube.	66.	Spam.	88.	Contenido.		

El **corpus lingüístico de Cibercrimen** es sustentado en los libros: “Cibercrimen, (Medina & Molist, 2015)” y “Delitos en la Red, (Poveda, 2015)”.

3.2. Estudios de máquinas con aprendizaje supervisado.

De acuerdo con (González, 2014), el Machine Learning se divide en dos áreas principales: Aprendizaje supervisado y Aprendizaje no supervisado. Aunque pueda parecer que el primero se refiere a la predicción con intervención humana y la segunda no, estos dos conceptos tienen más que ver con qué queremos hacer con los datos.

Uno de los usos más extendidos del aprendizaje supervisado consiste en hacer predicciones a futuro basadas en comportamientos o características que se han visto en

los datos ya almacenados, el histórico de datos (González, 2014). Las máquinas de aprendizaje supervisado han presentado un desarrollo con algunos altibajos desde los albores de la computación, incluso desde antes de los primeros equipos de cómputo con transistores.

Merchán Macías (2018), González Pacheco (2019), Iars.geo (2019), Rodríguez Rama (2018), han descrito la historia de las máquinas de aprendizaje, las redes neuronales, y algunas técnicas de aprendizaje supervisado remontándose a inicios del siglo XX . En 1913 Rusell desarrollo una máquina hidráulica basada en las redes neuronales. En 1936 Turing, el precursor de la computación moderna junto con Warren McCulloch y Walter Pitts en 1943, proponen el primer modelo matemático de red neuronal sin capacidad de aprendizaje. En 1949 Donald Hebb, desarrolló un algoritmo de aprendizaje, en 1951 la IAS fue la primera computadora Digital desarrollada por Von Neumann, en 1958, Frank Rosenblatt, crea el Perceptrón.

En los años 90s, el trabajo en Machine Learning gira desde un enfoque orientado al conocimiento (knowledge-driven) hacia uno orientado al dato (data-driven). Los científicos comienzan a crear programas que analizan grandes cantidades de datos y extraen conclusiones de los resultados. En 1992 Juyang Weng, publica el Cresceptrón, un método para realizar el reconocimiento de objetos 3D; en 1997 El ordenador Deep Blue de IBM vence al campeón mundial de ajedrez Gary Kaspárov (González Pacheco, 2019).

A mediados del año 2000, el término “Aprendizaje Profundo” comienza a ganar popularidad, después de un artículo llamado Deep Boltzmann Machines de Geoffrey Hinton y Ruslan Salakhutdinov (Hinton & Salakhutdinov, 2009), en el cual muestran como una red neuronal de varias capas podía ser pre entrenada con una capa a la vez.

Hacia el año 2009 se da el NIPS (Neural Information Processing Systems) Workshop sobre Aprendizaje Profundo para reconocimiento de voz, y se descubre que, con un conjunto de datos suficientemente grande, las redes neuronales no necesitan de un pre entrenamiento y los valores relativos al error caen significativamente.

En el año 2012, el algoritmo de aprendizaje profundo de Google, es capaz de identificar gatos y hacia el 2014 Google compra la Startup de Inteligencia Artificial “DeepMind” de Reino Unido. Por otro lado, en 2015 Facebook coloca en operación la tecnología de aprendizaje profundo “DeepFace” con el fin de identificar y etiquetar automáticamente usuarios de Facebook en las fotografías.

Más adelante, en el año 2016, el algoritmo de Google DeepMind, AlphaGo, mapea el arte del complejo juego de tablero Go y vence al campeón mundial de Go, Lee Sedol, en un torneo altamente divulgado en Seúl, capital de Corea del Sur.

Finalmente, en el año 2017, adopción en masa del aprendizaje profundo en diversas aplicaciones, así como también en investigaciones científicas y académicas. Todos los eventos de tecnología ligados al Data Science, IA y Big Data, apuntan al

aprendizaje profundo (Deep Learning) como la principal tecnología para la creación de sistemas inteligentes (Iars.geo, 2019).

3.2.1. Árboles de decisión.

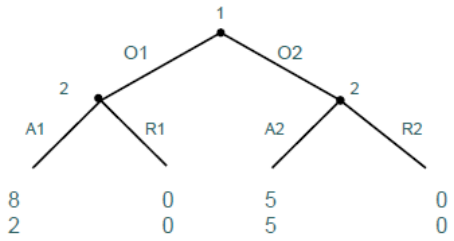
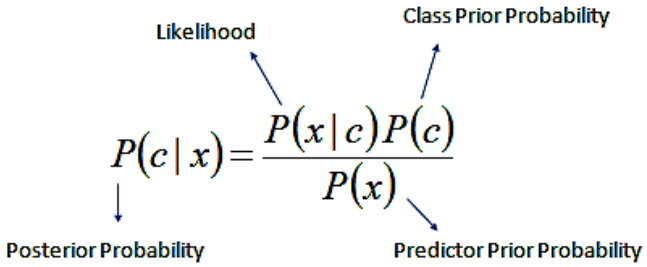


Figura 2. Esquema Árbol de Decisión (AlvaroV96, 2016).

La Fig. 2, muestra los árboles de decisión, los cuales son parte primordial entre las técnicas de machine learning y esto se debe a su facilidad tanto para ser entendidos y teniendo como primordial función la premisa de divide y vencerás.

En cada uno de los pasos, el conjunto de datos se divide en partes diferentes, mientras que cada parte debe representar mejor una de las clases posibles. El resultado final será una estructura de árbol donde cada nodo interno representa una prueba para el valor de un atributo particular y cada hoja representa la decisión para una clase particular. Un caso nuevo y desconocido se recorre el árbol hasta llegar a una de las hojas (Hofmann & Klinkenberg, 2014).

3.2.2. Naïve Bayes.



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figura 3. Teorema de Bayes. Fuente: http://uc-r.github.io/naive_bayes

La Fig. 3, presenta al clasificador Naïve-Bayes, el cual se encarga de construir modelos que predicen la probabilidad de posibles resultados, este se basa en el teorema de Bayes (1763) y en la premisa de independencia de los atributos dada una clase (Gutiérrez Esparza, Margain Fuentes, Canul Reich, & Ramírez del Real, 2017).

Una explicación más concreta de lo que es el MNB la describen, (Chandra, Gupta, & Gupta, 2007), cuando plantean que este clasificador aprende de los datos de entrenamiento y luego predice la clase de la instancia de prueba con la mayor probabilidad posterior. También es útil para datos dimensionales altos ya que la probabilidad de cada atributo se estima independientemente citado en (Mosquera, Castrillón, & Parra, 2018). Del mismo modo, lo plantean (Tan, Cheng, Wang, & Xu, 2009) y (Kaur & Singla, 2016), como uno de los métodos de aprendizaje supervisado más utilizados debido a que es posible adaptarlo para realizar análisis de emociones (Gutiérrez Esparza et al., 2017).

3.2.3. El K-vecino más cercano - KNN

La Fig. 4, exhibe otra técnica utilizada del aprendizaje supervisado y en este caso por proximidad, la cual corresponde al método K-NN. El procedimiento compara valores con los datos más próximos que son conocidos, y si son parecidos el dato se ubicará en la clase que más se acerque al valor de sus propios atributos. (Vidueira Ferreira et al., 2015).

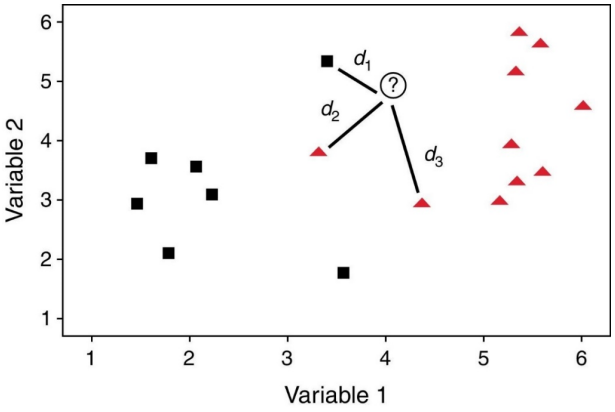


Figura 4. Proceso de clasificación de objeto desconocido. Fuente: (Vidueira Ferreira et al. 2015)

Algunas características de este algoritmo según (Quezada Lucio, 2017) son:

- Precisa de una definición de una métrica que ayude a comparar las distancias entre los objetos.
- Gozan de simplicidad conceptual: la clasificación de un nuevo espacio de representación se calcula en función de las clases conocidas de antemano, de los puntos más próximos a él. Así las muestras pertenecientes a una clase se encontrarán próximas en el de los puntos más próximos a él.

3.2.4. Redes neuronales.

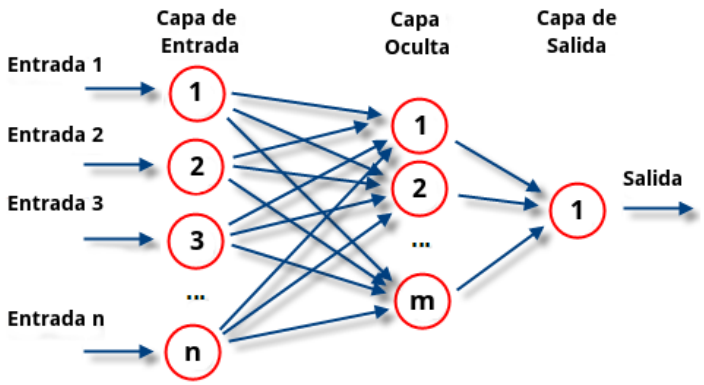


Figura 5. Red Neuronal. Fuente: (Gengiskanhg, 2004).

La Fig. 5, describe el esquema de la red neuronal, el cual se asemeja a la estructura que tienen las neuronas cerebrales; con la condición de que sus capas están interconectadas y que sus relaciones son incrementales hacia delante y no tiene posibilidad de retroceder o ir hacia los lados.

Una red neuronal puede aprender de los datos, de manera que se puede entrenar para que reconozca patrones, clasifique datos y pronostique eventos futuros. (MathWorks, 2016).

La ventaja de la red neuronal reside en el procesado paralelo, adaptativo y no lineal. Las aplicaciones de las redes neuronales, se pueden clasificar de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización.

Normalmente, las redes neuronales se utilizan para problemas como la clasificación, la predicción, el reconocimiento de patrones, los enfoques (aproximación) y las asociaciones.

Sólo necesitan aprender de algunos datos de muestra, y después de haberlos aprendido, pueden trabajar con datos de entrada desconocidos, o incluso datos de entrada ruidosos o incompletos; existen tres tipos de redes neuronales (1. Red neuronal de una sola capa (Fig.6), 2. Red neuronal multicapa (Fig. 7) y 3. Redes neuronales recurrentes (Fig. 8)), que se utilizan a menudo en función del tipo de red, a saber: (Aprilla, C, Baskoro, Ambarwati, & Wicaksana, 2013).

3.2.4.1. Red neuronal de una sola capa:

Este tipo de red se caracteriza por operar una sola capa de datos.

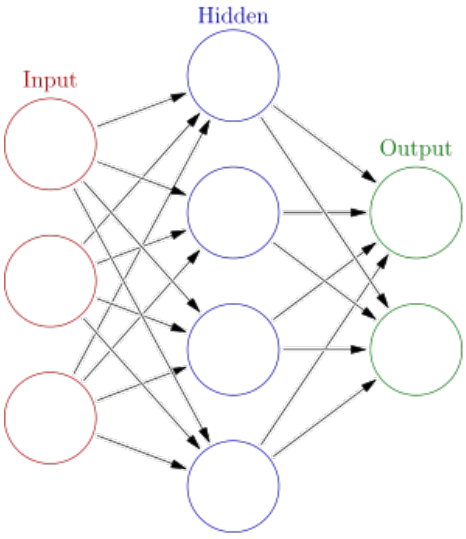


Figura 6. Red neuronal de una sola capa (Glosser.ca, 2013).

3.2.4.2. Perceptrón multicapa red neuronal.

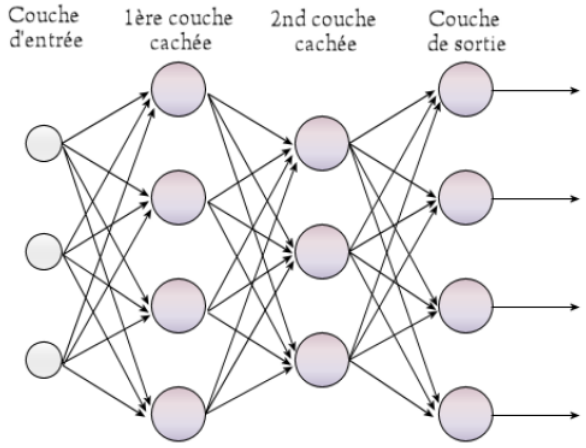


Figura. 7. Perceptrón (HRcommons, 2010).

3.2.4.3. Redes neuronales recurrentes

La desventaja de este tipo de algoritmo es el retraso de tiempo debido al proceso de retroalimentación desde la salida hasta el punto de entrada.

De abajo hacia arriba: estado de entrada, estado oculto, estado de salida. U, V, W son los pesos de la red. Diagrama comprimido a la izquierda y la versión desplegada a la derecha (Deloche, 2017).

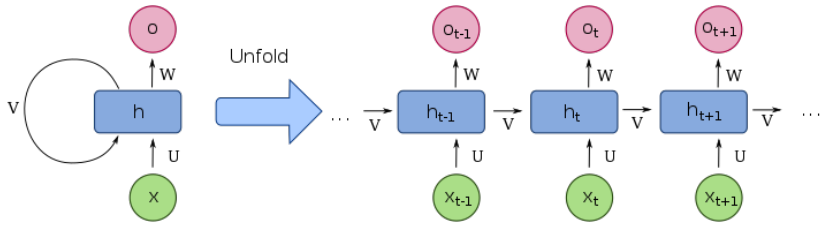


Figura 8. Red neuronal recurrente. (Deloche, 2017)

3.2.5. Máquina de Soporte Vectorial.

La Fig. 9, representa los dos tipos de división de datos que utilizan las Máquinas de Soporte Vectorial (SVM). Este tipo de algoritmo es famoso por ser lineal y que permite su utilización para la clasificación, regresión, estimación de densidad, detección de novedades, etc.

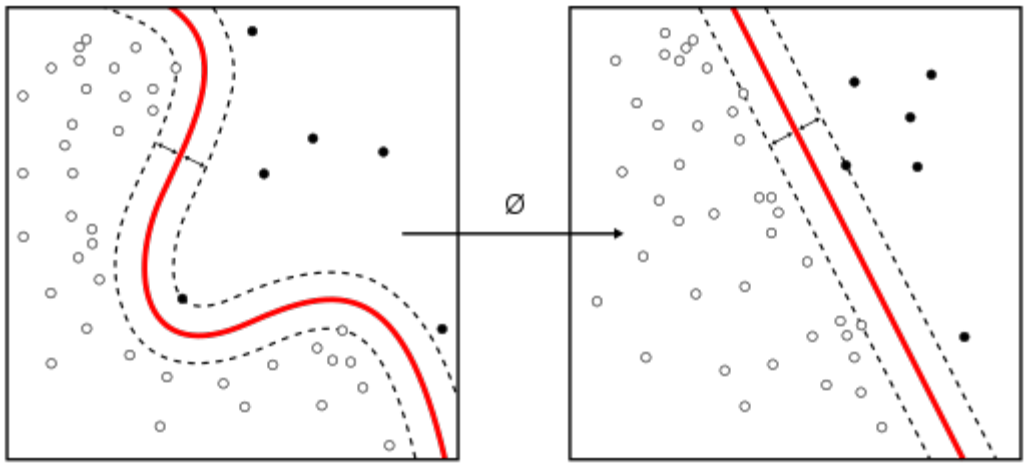


Figura 9. Ejemplo de aplicación de SVM (Vinco, 2017).

Según Zhang, en el caso más simple de clasificación de dos clases, las SVM encuentran un hiperplano que separa las dos clases de datos con un margen tan amplio como sea posible. Esto conduce a una buena precisión de generalización en datos invisibles y admite métodos de optimización especializados que permiten a SVM aprender de una gran cantidad de datos.

Como modelo lineal, no sólo trata de clasificar correctamente los datos de entrenamiento, sino que también maximiza el margen. Esta formulación conduce a un hiperplano de separación que depende sólo de los puntos de datos que se encuentran en el margen, que se denominan vectores de soporte.

Además, dado que los problemas de análisis de datos en el mundo real a menudo involucran dependencias no lineales, los SVM se pueden extender fácilmente para modelar tales no linealidades por medio de núcleos semi finitos positivos. Además, las SVM pueden ser entrenados a través de la programación cuadrática, lo que hace que el análisis teórico es más fácil y proporciona mucha conveniencia en el diseño de solucionadores eficientes que escalan para grandes conjuntos de datos (Zhang, 2017).

3.2.6. Métodos de consenso (Random Forest).

La Fig. 10, muestra el diagrama del algoritmo Random Forest, la cual representa a una técnica de aprendizaje en conjunto. El método es un híbrido entre el algoritmo de empaquetado y el método de subespacio aleatorio, y utiliza árboles de decisión como el clasificador de base.

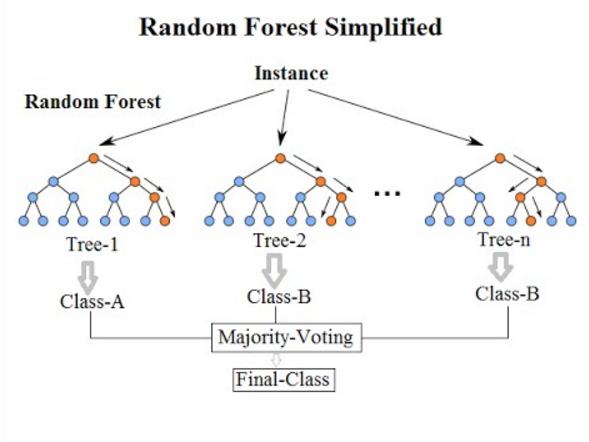


Figura. 10. Ejemplo Random Forest (Jagannath, 2017).

Cada árbol se construye a partir de una muestra de arranque del conjunto de datos original. Un punto importante es que los árboles no están sujetos a podas después de la construcción, lo que les permite estar parcialmente equipados con su propia muestra

de datos. Para diversificar aún más los clasificadores, en cada rama del árbol, la decisión de qué entidad dividir se restringe a un subconjunto aleatorio de tamaño n, del conjunto de características completo (Sammut & Webb, 2017).

3.2.7. Algoritmo C 4.5.

El algoritmo C 4.5. puede manejar datos numéricos y discretos. El algoritmo C 4.5. utiliza la relación de ganancia. Antes de calcular la relación de adquisición, es necesario calcular el valor de la información en unidades de bits de una colección de objetos, es decir, utilizando el concepto de entropía (Aprilla, C et al., 2013).

El algoritmo C 4.5., es utilizado para generar un árbol de decisión. Los árboles de decisión generados por C 4.5. se pueden usar para la clasificación, y por esta razón, a menudo se les conoce como un clasificador estadístico. Los resultados variarán significativamente si se cambian los datos de entrenamiento. Esta variación se conoce como error debido a la varianza que se puede minimizar utilizando varias combinaciones de clasificadores (Gyanchandani, Yadav, & Rana, 2010).

3.2.8. Ada Boosting.

El término boosting hace referencia a un tipo de algoritmos cuya finalidad es encontrar una hipótesis fuerte a partir de utilizar hipótesis simples y débiles. El algoritmo AdaBoost, fue creado por Freund y Schapire y es un diseño mejorado del boosting original; en términos de funcionalidad son iguales ambos algoritmos buscan crear un

clasificador fuerte cuya base sea la combinación lineal de clasificadores «débiles simples» (William L. Hosch, 2009), como se muestra en la Fig.11.

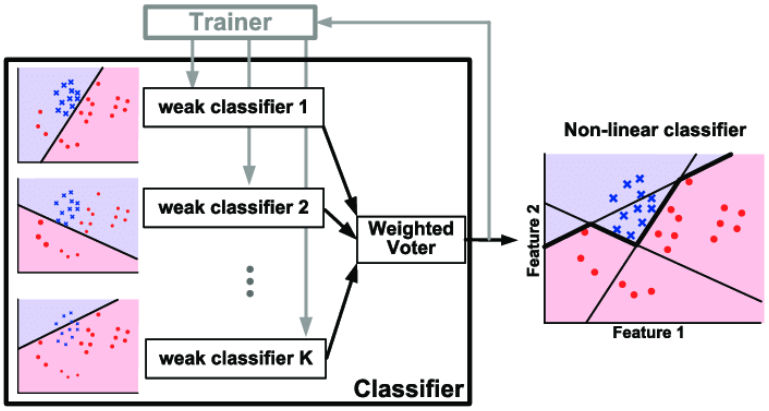


Figura 11. Algoritmo de AdaBoost para crear un clasificador fuerte basado en múltiples clasificadores lineales débiles (Wang, 2015).

3.3. Diseño del modelo de clasificación.

El diagrama mostrado en la Fig.12, describe el proceso para llevar a cabo la predicción del vocabulario referente a Cibercrimen.

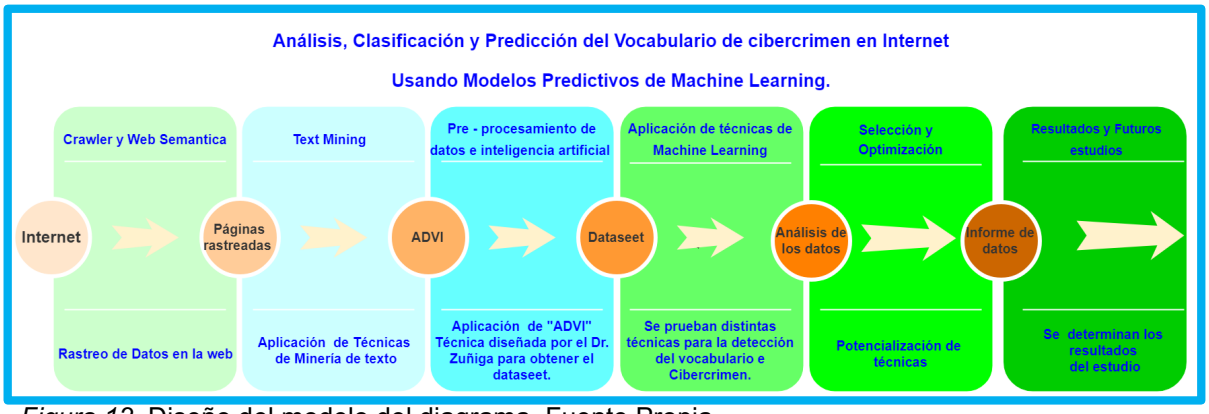


Figura 12. Diseño del modelo del diagrama. Fuente Propia

3.3.1. Crawler y Web Semántica.

Para construir la muestra de datos, se localizaron y descargaron más de 1150 páginas Web mediante un Crawler; estos rastreadores web ayudan en la recolección de información sobre un sitio web y los enlaces relacionados con ellos, además ayudan a validar el código HTML e hipervínculos (Arimetrics, 2016).

3.3.2. Text Mining.

Según (Rochina, 2017)*, La minería de datos se puede definir como el análisis matemático para deducir patrones y tendencias que existen en los datos, patrones que no pueden detectarse mediante una exploración tradicional de los datos porque las relaciones son demasiado complejas o por el volumen de datos que se maneja.*

Estos patrones y tendencias se pueden recopilar y definir como un modelo de minería de datos. Para ello, en el mismo sentido, la minería de textos comprende tres actividades fundamentales:

- Recuperación de la información: *Consiste en seleccionar los textos pertinentes.*
- Extracción de la información: *Incluida en esos textos mediante el procesamiento del lenguaje natural: hechos, acontecimientos, datos clave, relaciones entre ellos, etc.*

- Minería de datos: *Para encontrar asociaciones entre los datos clave previamente extraídos de entre los textos* (Rochina, 2017).

3.3.3. Preprocesamiento con la aplicación "ADVI".

La presente investigación, se apoya en el trabajo realizado por el Dr. Castillo-Zúñiga & otros los cuales en su documento Architecture (ADVI) for the detection of cyberbullying vocabulary in internet combining techniques of big data analytics and semantic. Explican cómo fue el proceso donde implementaron una estrategia genética en paralelo para la recuperación del vocabulario, la cual integra técnicas de Web Semántica (ontologías) y Procesamiento de Lenguaje Natural (PLN) (Tokenización, Stop Word, Frecuencia de Término (TF) y Frecuencia de Término con Frecuencia Inversa del Documento (TF-IDF)), métodos de lematización y sinónimos, con el propósito de recuperar más información (Castillo Zuñiga, I., Luna Rosas, F., Muñoz Arteaga, J., Lopez Veyna, 2016).

3.3.4. Aplicación de técnicas de Machine Learning.

Con fundamento en los trabajos relacionados fueron seleccionadas las técnicas: árboles de decisión, Naïve Bayes, K-vecino más cercano, redes neuronales, Máquinas de Soporte Vectorial, métodos de conceso (Random Forest), Ada Boosting y C 4.5. (Madala et al., 2018; Mbaziira & Jones, 2016; Moohebat et al., 2015; Rodríguez Rama, 2018).

3.3.5. Selección y optimización de técnicas.

Después de utilizar los dataset generados con la herramienta mencionada, y con base en los resultados obtenidos de las pruebas con las distintas técnicas se establece cuáles son las más apropiadas para el desarrollo de la investigación y se pretende enfocar el trabajo de análisis y mejora sobre las técnicas que presenten mayor porcentaje de aceptabilidad.

3.3.6. Resultados y futuros estudios.

En este espacio se describirán los distintos resultados obtenidos después de aplicar las herramientas de aprendizaje supervisado en el lenguaje de programación R; posteriormente, plantear futuros estudios pertinentes al tema de investigación que aporten en dar soluciones a problemas de un entorno social.

3.4. Evaluación del modelo de clasificación.

Para realizar adecuadamente la clasificación del vocabulario referente a Cibercrimen, se realizaron pruebas con los distintos algoritmos tratados en el punto 3.3.4., dentro de los cuales se aplicaron las siguientes técnicas: K-Vecinos más cercanos K-NN, Arboles de Decisión, Random Forest, Máquinas de Soporte Vectorial, Regresión Lineal, Redes Neuronales Naïve Bayes, AdaBoost y C 4. 5.

Los distintos métodos se implementaron en Rstudio, no sin antes preparar el entorno de desarrollo con las distintas librerías para el proceso como: E1071 para el algoritmo de Random Forest; kk-nn para K-vecinos más cercanos; randomForest para Bosques Aleatorios; rpart y rpart. plot para Arboles de Decisión; kernlab para Máquinas de Soporte Vectorial; VGAM para Regresión Lineal; y la librería para el algoritmo de redes neuronales fue nnet. La estructura principal para que estas librerías sean soportadas adecuadamente es la herramienta de computación científica R.

Sobre el 100% de los datos que se evaluaron se toma un porcentaje del 70% para realizar el adiestramiento del algoritmo y luego un 30% para realizar la evaluación de los datos; posteriormente a partir de las distintas pruebas se determinan cuáles son los algoritmos más adecuados para la detección de vocablos referentes a Cibercrimen.

3.5. Comparar el modelo con otros modelos similares usados en el estado del arte.

Al realizar la comparativa de resultados con respecto a la investigación adelantada por (Mbaziira & Jones, 2016). A Text-based Deception Detection Model for Cybercrime, se determinó que con el modelo desarrollado por ellos para detectar ciberdelitos por género Web capaces de detectar estafas en las redes sociales a partir de los mensajes, solo ofrece un 60% de exactitud en la predicción; los modelos

entrenados sobre fraude en el correo electrónico pudieron predecir las estafas en un 50% de exactitud.

El modelo planteado por Rodríguez Rama (2018), Aplicación de técnicas de machine learning a la detección de ataques, como ya se había planteado antes, muestra un modelo predictivo, utilizando Weka y la librería de software Scikit-Learn. El dataset utilizado es «KDD Cup 1999» que incluye una amplia variedad de intrusiones de red simuladas en un entorno de red militar, comparándolo con la investigación actual, este entorno es simulado en su dataset mientras que la investigación actual utiliza un Crawler para la extracción de datos y se realiza una comprobación de distintas técnicas de minería de datos para corroborar cuál es la más eficiente en detectar el vocabulario referente al Cibercrimen.

El trabajo de (Basher & Fung, 2014), Analyzing topics and authors in chat logs for crime investigation, presenta un desarrollo interesante en la búsqueda de temas para analizar los archivos de los registros de chat para separar los registros relevantes para detectar el crimen, ellos desarrollaron y aplicaron sus propias técnicas para LDA a lo largo del tiempo (LDA-TOT) y temas de autor a lo largo del tiempo (A-TOT), más no se puede realizar una comparación directa con los algoritmos aplicados en la investigación actual.

3.6. Herramientas usadas en la analítica de Big Data.

3.6.1. Hardware.

Para el desarrollo de las pruebas de la presente investigación, se utilizaron distintas herramientas tanto de software como de hardware, con un computador con las siguientes características: Equipo HP ProBook 6475b, con 8GB de memoria RAM DDR3, disco duro de 500 GB y procesador AMD A10.

3.6.2. Software.

El Sistema Operativo usado, es Windows 10 Pro del 2018, con una suite de ofimática con Office profesional Plus, 2016; el equipo cuenta con un entorno de programación orientada a la investigación científica mediante el lenguaje R versión 3.6.0, adicional el entorno de desarrollo R Studio en la Versión 1.1.383.

Capítulo IV. Resultados y discusión.

4.1. Procedimiento general del ensayo.

En la presente investigación el método de pruebas que se llevó a cabo para lograr el análisis, clasificación y predicción del vocabulario de Cibercrimen en Internet a través de modelos predictivos de machine learning, fue el siguiente:

1. Después de establecer los objetivos y alcance de la investigación, se determinó implementar la herramienta ADVI, para obtener el dataset que será utilizado en las pruebas de detección del Vocabulario de Cibercrimen (Castillo Zuñiga, I., Luna Rosas, F., Muñoz Arteaga, J., Lopez Veyna, 2016). En donde se lleva a cabo el siguiente procedimiento:
    - Primero, se determinan las características generales del corpus lingüístico, donde se genera una estructura a partir de metadatos y grafos para las ontologías semánticas a consultar.
    - Se localiza el grupo de páginas Web utilizando el Crawler y se almacenan en un disco duro.
    - Posteriormente, se genera el vocabulario a través de técnicas de PLN y Web semántica.
    - Por último, se enlaza la estructura de las ontologías y el vocabulario para obtener el dataset a ser evaluado.

2. El dataset está compuesto por 1162 sitios Web y 107 vocablos en español relacionados al vocabulario de Cibercrimen, el cual es sustentado en los libros:

"Cibercrimen, (Medina & Molist, 2015)" y "Delitos en la Red, (Poveda, 2015)" como se especificó anteriormente, (en el objetivo 3.1 fundamento del dataset). La estructura del dataset contiene alrededor de 120000 datos en un único archivo CSV, el cual es utilizado para realizar las pruebas con las técnicas de machine learning, y generar los modelos de detección o predicción de Cibercrimen.

3. El siguiente paso fue realizar el análisis de los datos. Para ello, se trabajó en el entorno de análisis de datos de R. Donde según (Ferrero, 2018), R es un lenguaje de programación diseñado para el análisis estadístico formado por un conjunto de herramientas muy flexibles que pueden ampliarse fácilmente mediante paquetes, librerías o definiendo nuestras propias funciones. Además, es gratuito y de código abierto, un Open Source parte del proyecto GNU, como Linux o Mozilla Firefox. La Fig. 13, muestra el emblema del lenguaje R.

Figura 13. Emblema de R (Ferrero, 2018).

Cabe mencionar, que muchas universidades (incluida la Universidad Cuauhtémoc), y empresas utilizan el software R para sus análisis, además, es uno de los

lenguajes más utilizados en investigación científica. La Fig. 14, muestra el entorno de R Studio con la versión 3.6.0. de 64 bit.

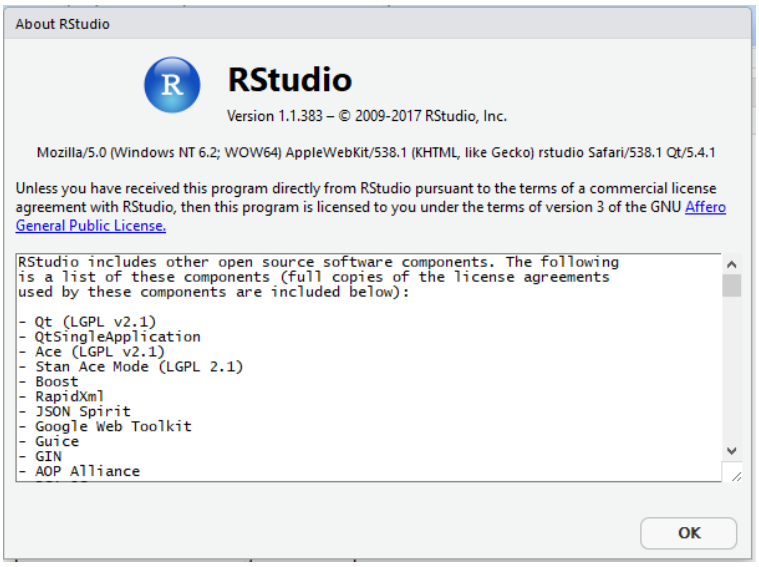


Figura 14. Imagen del entorno R Studio mediante captura de pantalla.

Para realizar las distintas pruebas con los algoritmos de aprendizaje supervisado, fue necesario la instalación de distintas librerías tanto para el entorno de R como de RStudio. La Fig. 15, muestra un ejemplo de la instalación de librerías.

Figura 15. Instalación de la librería RWeka para el algoritmo C 4.5. Fuente Propia.

Dentro de las librerías que fueron instaladas, se incluyen: ada 2.0-5, adabag 4.2, caret 6.0-84, e1071 1.7-1, ggplot2 3.1.1, kK-NN 1.3.1, nnet 7.3-12, randomForest 4.6-14, rJava 0.9-11, rpart. plot 3.0.7, RWeka 0.4-40 y VGAM 1.1-1, entre otras.

4. Se evaluó el conjunto de datos con cada algoritmo, posteriormente se determinó cual fue el algoritmo que presentó mejores resultados, estableciendo los métodos más adecuados para la detección de vocabulario referente a Cibercrimen.
5. Se asentaron los estudios futuros que se pueden realizar a partir de la presente investigación.

4.2. Pruebas con los algoritmos de aprendizaje seleccionados.

En el siguiente apartado se presentan los distintos resultados obtenidos con cada uno de los algoritmos implementados para el análisis, clasificación y predicción del vocabulario de Cibercrimen en Internet usando modelos predictivos de machine learning.

4.2.1 Método K-Vecinos más cercanos KNN.

La Fig. 16, muestra la implementación de un script en RStudio con los parámetros necesarios para poder probar el dataset con el algoritmo K– vecinos más cercanos K-NN. En el cual se genera la función modelo a partir de la tabla de aprendizaje, posteriormente se predicen los datos para el modelo de prueba y la matriz de confusión.

La matriz de confusión contiene información acerca de las predicciones realizadas por el sistema de clasificación, comparando individuos de la tabla de aprendizaje con individuos de la tabla de pruebas (testing), obteniendo el porcentaje de precisión sobre la detección del vocabulario del Cibercrimen (Rodríguez, 2013).

```
##---------------Método K-vecinos mas cercanos KNN---------------------

suppressWarnings(suppressMessages(library(kknn)))

#Generamos la función modelo con la tabla de aprendizaje
#especificando la columna de la variable discriminante

modelo<-train.kknn(Correcta~.,data=taprendizaje,kmax=9)
modelo

#Generamos los valores predichos por el modelo en
#los datos de prueba

prediccion<-predict(modelo,ttesting[,-107])
prediccion

## Matriz de Confusión
MC<-table(ttesting[,107],prediccion)
MC

##Obtenemos el nivel de aciertos
acierto<-(sum(diag(MC)))/sum(MC)
acierto

#Obtenemos el error
error<-1-acierto
error
##Fin K vecinos mas cercanos
```

Figura 16. Código K-vecinos más cercanos K-NN para evaluar los dataset. Fuente propia

La Fig. 17, muestra los resultados del modelo K– vecinos más cercanos K-NN, la cual obtiene un 80% en la precisión del vocabulario de Cibercrimen.

```
> #Generamos los valores predichos por el modelo en
> #los datos de prueba
>
> prediccion<-predict(modelo,ttesting[,-107])
> prediccion
  [1] Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si
 [37] Si Si No Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si No Si Si Si Si Si Si
 [73] Si Si Si Si Si Si No Si No Si Si Si Si Si No Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si Si Si Si Si Si
[109] Si Si Si Si Si Si Si No No Si Si Si Si Si No Si Si Si Si No No Si No No No Si No No Si No Si No No No No Si Si
[145] Si Si Si Si Si Si Si Si Si Si Si No No Si No Si No No Si Si No No Si Si No Si Si Si Si Si Si No Si No Si Si Si
[181] No Si No Si Si Si No No No Si No No Si No Si No No Si No Si No No No No Si Si Si Si No Si No No Si No Si No
[217] No Si No Si No Si Si No No No No No No Si Si No No No No No No No No No Si No No No No Si No No No No No No
[253] No No No No No No No No No No No No No No No No No No No No No No No No No No No No No No No Si Si No Si
[289] No No Si No No No No No No Si No No No No No No No No No No No No No No No Si No No No No No No No No No No
[325] No No No No No No No No
Levels: No Si
>
> ## Matriz de Confusión
> MC<-table(ttesting[,107],prediccion)
> MC
    prediccion
      No  Si
  No  99  10
  Si  56 167
>
> ##Obtenemos el nivel de aciertos
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.8012048
>
> #Obtenemos el error
> error<-1-acierto
> error
[1] 0.1987952
> ##Fin K vecinos mas cercanos
```

Figura 17. Resultados modelo K-NN. Fuente propia.

4.2.2 Método Naïve Bayes.

La Fig. 18, Presenta el desarrollo de la implementación de un script en RStudio con los parámetros necesarios para poder probar el dataset generado con la herramienta (ADVI) en el algoritmo Naïve Bayes.

```
##---------------Método de Bayes--------------------------------

suppressWarnings(suppressMessages(library(e1071)))

#Generamos el modelo Naive-Bayes, especificando la variable discriminante
modelo<-naiveBayes(Correcta~.,data=taprendizaje)
modelo

#Generamos los valores predichos por el modelo en los datos de prueba
prediccion<-predict(modelo,ttesting[,-107])
prediccion

## Matriz de Confusion
MC<-table(ttesting[,107],prediccion)
MC

##Obtenemos el nivel de aciertos
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

#Obtenemos el error
error<-100-acierto
error
##Fin Método Bayes
```

Figura 18. Código Naïve Bayes para evaluar los dataset. Fuente propia

A partir de la tabla de aprendizaje se genera la función modelo. La Fig. 19, muestra el proceso de análisis de los valores de las características a evaluar sobre cada sitio Web localizado de Cibercrimen, estableciendo la matriz de confusión.

```
> #Generamos los valores predichos por el modelo en los datos de prueba
> prediccion<-predict(modelo,ttesting[,-107])
> prediccion
   7    8   20   25   31   37   43   46   49   50   53   57   59   66   68   70   71   78   80   87   90
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
  94   95   97  109  113  117  120  124  125  128  129  143  147  148  152  158  159  161  162  163  165
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 168  173  187  189  190  201  208  214  216  226  230  232  236  247  253  255  256  257  262  272  274
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 283  287  291  292  293  298  301  306  308  309  311  315  316  320  324  328  332  336  338  343  345
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 349  357  360  361  365  366  367  369  372  373  379  385  387  390  392  393  398  402  421  424  425
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 430  433  434  440  448  450  457  459  462  464  465  470  475  476  484  485  487  492  501  502  507
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 516  517  523  524  526  535  541  547  548  560  563  567  569  575  577  580  582  583  588  590  601
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 602  606  609  616  621  624  630  632  633  634  638  640  651  653  656  660  663  664  665  672  674
  Si   Si   Si   Si   Si   Si   No   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 677  678  681  690  692  694  698  700  701  703  706  707  720  721  723  724  734  737  758  763  767
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 778  782  785  787  788  793  794  798  801  802  805  809  832  835  836  840  842  844  847  852  857
  Si   Si   Si   Si   Si   Si   Si   No   Si   Si   Si   Si   Si   Si   Si   Si   Si   No   Si   Si   Si
```

Figura 19. Predicción de los datos. Fuente propia.

La Fig. 20, presenta los resultados de la matriz de confusión, la cual muestra información acerca de las predicciones realizadas por el sistema de clasificación, comparando individuos de la tabla de aprendizaje con individuos de la tabla de pruebas (testing), obteniendo el porcentaje de precisión sobre la detección del vocabulario del Cibercrimen (Rodríguez, 2013).

```
> ## Matriz de Confusion
> MC<-table(ttesting[,107],prediccion)
> MC
    prediccion
      No  Si
  No 102   7
  Si  53 170
>
```

Figura 20. Datos de la Matriz de confusión para Naïve Bayes. Fuente propia.

Al implementar este método se específica la variable discriminante y luego se generan los valores predichos por el modelo en los datos de prueba, y se genera la matriz de confusión, donde se establecen los niveles de acierto y el porcentaje de error respectivo presentados en la Fig. 21., y como resultado se aprecia un 82% en la precisión del vocabulario de Cibercrimen.

```
>
> ##Obtenemos el nivel de aciertos
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.8192771
>
> #Obtenemos el error
> error<-1-acierto
> error
[1] 0.1807229
> ##Fin Método Bayes
>
```

Figura 21. Porcentaje de aciertos del método Bayes. Fuente Propia.

4.2.3 Método Random Forest.

La imagen de la Fig. 22, describe el código del programa en Rstudio, resaltando los parámetros necesarios para poder probar el dataset generado con la herramienta (ADVI) en el algoritmo Random Forest.

```
##---------------Método de Random Forest------------------------------

suppressWarnings(suppressMessages(library(randomForest)))

#Generamos el modelo, especificando la variable discriminante
modelo<-randomForest(Correcta~.,data=taprendizaje,importance=TRUE)
modelo
#plot(modelo)

#Generamos los valores predichos por el modelo en los datos de prueba
prediccion<-predict(modelo,ttesting[,-107])
prediccion

## Matriz de Confusion
MC<-table(ttesting[,107],prediccion)
MC

# Porcentaje de buena clasificación
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del método Random Forest
```

Figura 22. Código Random Forest para evaluar los dataset. Fuente propia.

Posteriormente, se genera la función modelo a partir de la tabla de aprendizaje, tal y como se muestra en la Fig. 23, donde se predicen los datos para el modelo de prueba y la matriz de confusión.

```
> #Generamos los valores predichos por el modelo en los datos de prueba
> prediccion<-predict(modelo,ttesting[,-107])
> prediccion
  [1] Si Si No Si Si Si Si Si No Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si
 [51] Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si
[101] Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si Si
[151] Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si
[201] Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si Si
[251] Si Si Si No Si Si Si Si Si Si Si Si Si Si Si Si Si Si No Si Si Si Si Si No Si Si Si Si Si Si Si No Si Si Si Si Si Si Si Si Si
[301] Si Si Si Si Si Si Si No Si Si Si Si Si Si Si Si Si No Si Si Si Si Si No Si No Si Si Si Si No Si Si Si Si Si Si Si Si Si Si Si
[351] Si No Si Si No Si No Si Si Si No Si Si Si Si Si No Si Si Si No No Si Si No Si Si Si Si Si Si No Si Si No Si Si Si Si No Si Si No Si Si
[401] Si No Si Si Si No Si Si No Si Si Si Si Si No Si No No No No No Si Si No Si Si Si Si Si Si Si No Si Si No Si Si No No No No Si Si Si
[451] Si Si Si Si Si No No Si Si No Si No No Si Si Si Si Si Si Si No No Si Si Si Si No No No No Si Si No No No No Si Si Si No Si Si Si Si Si Si
[501] Si Si No Si No No Si Si No Si Si No Si No No Si No Si Si No Si No Si No Si No No No No Si Si Si Si No No Si No Si Si Si No No No Si No
```

Figura 23. Predicción de los datos. Fuente propia.

La Fig. 24, presenta los resultados de la matriz de confusión.

```
>
> ## Matriz de Confusion
> MC<-table(ttesting[,107],prediccion)
> MC
    prediccion
      No  Si
  No  92  17
  Si   4 219
>
```

Figura 24. Datos de la Matriz de confusión para Random Forest. Fuente propia.

Al implementar este método se específica la variable predictora y luego se generan los valores a predecir por el modelo, se obtiene la matriz de confusión, y se establecen los niveles de acierto y el porcentaje de error respectivo para el método Random Forest, representados en la Fig. 25., generando un 94,72% de efectividad en la precisión del vocabulario de Cibercrimen.

```
>
> # Porcentaje de buena clasificación
> acierto<-(sum(diag(MC)))/sum(MC)*100
> acierto
[1] 94.72362
>
> error<-100-acierto
> error
[1] 5.276382
> ## Fin del método Random Forest
```

Figura 25. Porcentaje de aciertos del método Random Forest. Fuente Propia

4.2.4 Método Árboles de decisión.

La siguiente implementación (Fig. 26), presenta el desarrollo de un script en RStudio con los parámetros necesarios para poder probar el dataset generado con la herramienta (ADVI), para realizar pruebas con la técnica de Arboles de decisión.

```
##---------------Método de Arboles de Desición---------------

suppressWarnings(suppressMessages(library(rpart)))
suppressWarnings(suppressMessages(library(rpart.plot)))

#Generamos el modelo, especificando la variable discriminante
modelo <- rpart(Correcta~.,data=taprendizaje)
modelo
plot(modelo)
text(modelo)
prp(modelo,extra=104,branch.type=2, box.col=c("pink", "palegreen3")[modelo$frame$yval])

#Generamos los valores predichos por el modelo en los datos de prueba
prediccion <- predict(modelo, ttesting[,-107], type='class')
prediccion

## Matriz de Confusion
MC<-table(ttesting$Correcta,prediccion)
MC

# Porcentaje de buena clasificación
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del método Arboles de decisión
```

Figura 26. Código Arboles de decisión para evaluar los dataset. Fuente propia

Este método genera imágenes gráficas de los datos a ser valorados, en donde se representa a cada dato como una rama del árbol. Las Figuras 27 y 28 muestran la representación de los datos analizados en los árboles de decisión.

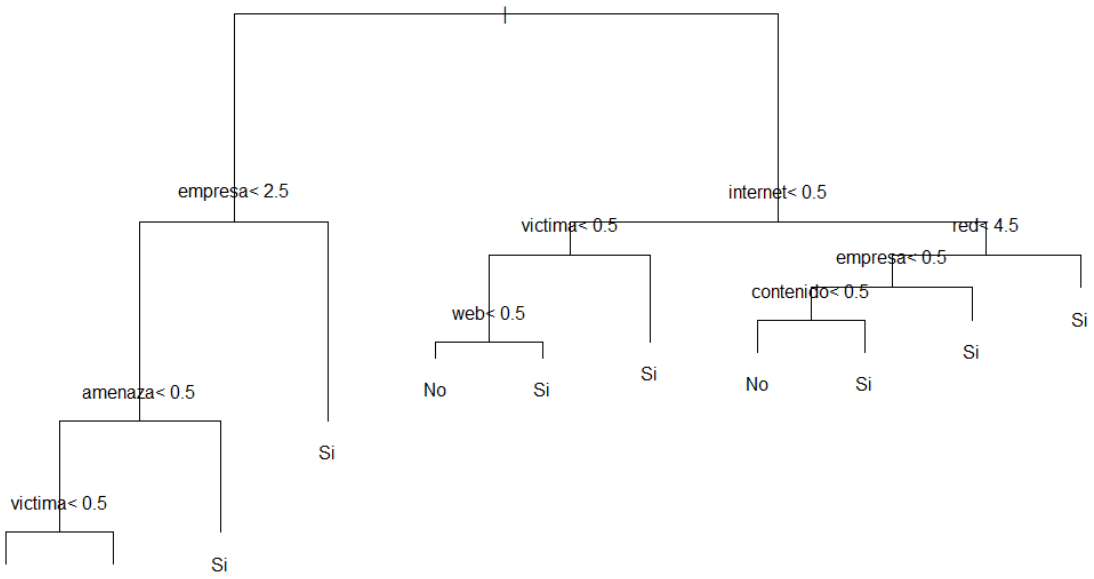


Figura 27. Representación de los datos en Árbol de decisión 1. Fuente propia.

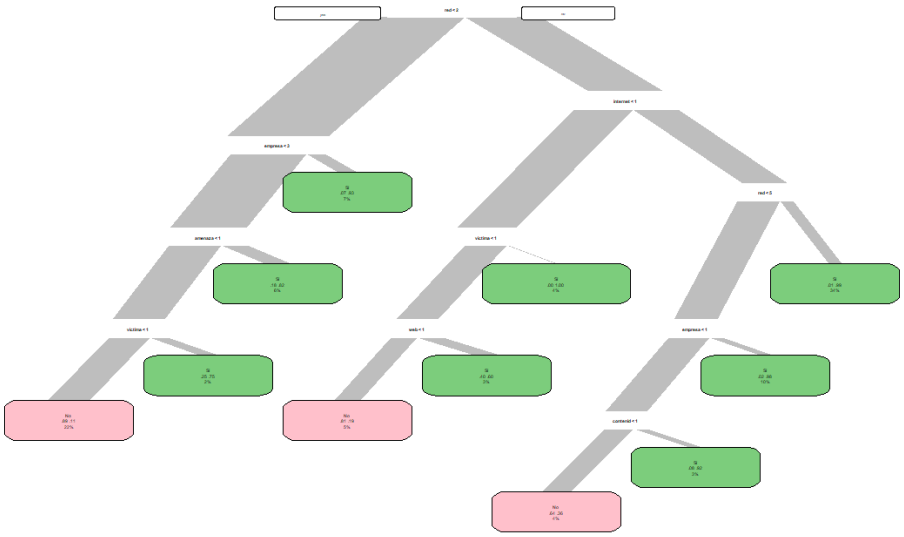


Figura 28. Representación de los datos en Árbol de decisión 2. Fuente propia.

En el proceso del algoritmo, se obtiene la función modelo y la matriz de confusión a partir de la tabla de aprendizaje, como se muestra en la Fig. 29. Además de generar los resultados de la precisión con que se detecta el Cibercrimen, como se muestra en la Fig. 30.

```
> #Generamos los valores predichos por el modelo en los datos de prueba
> prediccion <- predict(modelo, ttesting[,-107], type='class')
> prediccion
   1    2    3    4    7    8    9   13   14   15   16   17   19   21   22   23   24   25   26   27   28   29   30   31   32   34   36   38
 39   41   42
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 Si   Si   Si
  44   45   46   47   48   49   50   51   54   55   56   59   60   62   63   65   67   68   71   72   73   74   81   83   85   86   88   89
 90   91   92
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 Si   Si   Si
  94   95   96   98   99  101  102  103  104  105  106  107  108  109  110  111  112  114  115  116  118  119  120  121  122  123  124  127
128  130  131
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
 Si   Si   Si
 132  135  136  138  139  140  141  142  145  146  150  151  152  153  155  156  157  158  160  162  163  164  165  166  168  169  170  172
173  175  176
  Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si   Si
```

Figura 29. Predicción de los datos. Fuente propia.

```
> ## Matriz de Confusion
> MC<-table(ttesting$Correcta,prediccion)
> MC
    prediccion
      No  Si
  No  82  27
  Si  17 206
>
```

Figura 30. Datos de la Matriz de confusión para arboles de decisión. Fuente propia.

Se implementa el método especificando la variable a predecir (correcta) y luego se generan los valores predichos por el modelo en los datos de prueba, y se construye la matriz de confusión, se establecen los niveles de acierto y el porcentaje de error respectivos, como se presenta en la Fig. 31., en donde se obtiene un 86,74% en la precisión del vocabulario de Cibercrimen.

```
> # Porcentaje de buena clasificación
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.8674699
>
> error<-1-acierto
> error
[1] 0.1325301
> ## Fin del método Arboles de decisión
```

Figura 31. Porcentaje de aciertos del método arboles de decisión. Fuente Propia

4.2.5 Método de Máquinas de Soporte Vectorial (SVM).

La Fig. 32, muestra el programa con los parámetros necesarios para poder probar el dataset con el algoritmo de Máquinas de Soporte Vectorial en RStudio.

```
##---------------Método Maquinas de Soporte Vectorial (SVM)---------------

# El paquete 'kernlab' ofrece la función 'ksvm'.
suppressWarnings(suppressMessages(library(kernlab)))

#Generamos el modelo, especificando la variable discriminante
modelo <- ksvm(Correcta~.,
     data=taprendizaje,
     kernel="rbfdot", prob.model=TRUE, kpar=list(sigma=0.05),C=5,cross=3)
modelo

#Generamos los valores predichos por el modelo en los datos de prueba
prediccion <- predict(modelo, ttesting)
prediccion

## Matriz de Confusion
MC<-table(ttesting$Correcta,prediccion)
MC

# Porcentaje de buena clasificación
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del método Maquinas de Soporte Vectorial (SVM)
```

Figura 32. Código SVM para evaluar los dataset. Fuente propia.

El procedimiento genera la función modelo a partir de la tabla de aprendizaje, y realiza las pruebas de predicción con el dataset, obteniendo el modelo de prueba y la matriz de confusión, como se muestra en la Fig. 33., y Fig. 34.

```
> #Generamos los valores predichos por el modelo en los datos de prueba
> prediccion <- predict(modelo, ttesting)
> prediccion
  [1] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
 [51] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[101] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[151] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[201] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[251] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[301] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[351] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[401] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[451] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[501] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[551] si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si si
si si si si si
[601] si si si si si si si si si si si si si si si si si si si si si si si No si si si si si si si si si si si si si si si si si si si si si
```

Figura 33. Predicción de los datos. Fuente propia.

```
> ## Matriz de Confusion
> MC<-table(ttesting$Correcta,prediccion)
> MC
    prediccion
      No  Si
  No  76  33
  Si   0 223
>
```

Figura 34. Datos de la Matriz de confusión para SVM. Fuente propia.

En este método se aplica la función de base radial gaussiana, usando el algoritmo “ksvm” incluido en las bibliotecas de R, con el propósito de separar un hiperplano en el espacio transformado, para lograr una mejor clasificación de los datos y realizar la detección del vocabulario del Cibercrimen, como se describe en la Fig.35.

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 5

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.05

Number of Support Vectors : 841

Objective Function Value : -182.596
Training error : 0
Cross validation error : 0.096595
Probability model included.
```

Figura 35. Modelo Gaussiano generado por Ksvm. Fuente propia.

Finalmente, se presentan los porcentajes de predicción sobre la clasificación de los datos en la Fig. 36., con un 90% en la precisión del vocabulario de Cibercrimen.

```
> # Porcentaje de buena clasificación
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.9006024
>
> error<-1-acierto
> error
[1] 0.09939759
> ## Fin del método Maquinas de Soporte Vectorial (SVM)
>
```

Figura 36. Porcentaje de aciertos método de Máquinas de Soporte Vectorial.

4.2.6 Método de Regresión Lineal.

En la Fig. 35, se describe el código del método de Regresión Lineal, estableciendo los parámetros necesarios para poder realizar las pruebas al conjunto de datos establecido mediante la herramienta (ADVI).

```
##---------------Método Regresión Lineal----------------------------------


suppressWarnings(suppressMessages(library(VGAM)))

#Generamos el modelo, especificando la variable discriminante
modelo <- glm(Correcta ~ .,
    data=taprendizaje,
    family=binomial(link="logit"))
# summarize the fit
summary(modelo)
plot(modelo)

#Generamos los valores predichos por el modelo en los datos de pureba
prediccion <- as.vector(ifelse(predict(modelo,
                        type    = "response",
                        newdata = ttesting[,-107]) > 0.5, "Si", "No"))

## Matriz de Confusion
MC<-table(ttesting[,107],prediccion)
MC

# Porcentaje de buena clasificación
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del método Regresión Lineal
```

Figura 37. Código de Regresión lineal para evaluar los dataset. Fuente propia.

Posteriormente en la Fig. 38, se presentan los resultados de la matriz de confusión con la información acerca de las predicciones realizadas por el sistema de clasificación, identificando valores correctos representados por la variable predictora como un “Si”, y los valores no correctos, representados con el valor “No”.

```
> ## Matriz de Confusion
> MC<-table(ttesting[,107],prediccion)
> MC
    prediccion
      No  Si
  No  76  33
  Si   0 223
```

Figura 38. Resultados de la matriz de confusión método de Regresión Lineal. Fuente Propia.

Por último, se establecen los niveles de acierto y el porcentaje de error respectivo presentados en la Fig. 39; con un 90% en la precisión del vocabulario de Cibercrimen y un 10% de error.

```
>
> # Porcentaje de buena clasificación
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.9006024
>
> error<-1-acierto
> error
[1] 0.09939759
> ## Fin del método Regresión Lineal
>
```

Figura 39. Porcentaje de aciertos del método de Regresión lineal. Fuente Propia.

4.2.7 Método de Redes Neuronales.

La Fig. 40, contiene el código del programa de Redes Neuronales, escrito en lenguaje R, en el cual se especifican los parámetros necesarios para el conjunto de datos, puntualizando los porcentajes tanto para el aprendizaje, como para las pruebas.

```
##---------------Método Redes Neuronales-----------------------------------
#1.7 MÃ©todo Redes Neuronales
suppresswarnings(suppressMessages(library(nnet)))

set.seed(121)
#Generamos el modelo, especificando la variable discriminante
modelo <- nnet(as.factor(Correcta) ~ .,
               data=taprendizaje,
               size=10, skip=TRUE, MaxNWts=10000, trace=FALSE, maxit=100)
summary(modelo)


#Generamos los valores predichos por el modelo en los datos de pureba
prediccion <- predict(modelo, newdata=ttesting[,-107], type="class")


## Matriz de Confusion
MC<-table(ttesting[,107],prediccion)
MC

# Porcentaje de buena clasificaciÃ³n
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del método Redes Neuronales
```

Figura 40. Código de Redes Neuronales para evaluar los dataset. Fuente propia.

La Fig. 41, presenta la matriz de confusión con la información acerca de las predicciones realizadas por el sistema de clasificación.

```
> ## Matriz de Confusion
> MC<-table(ttesting[,107],prediccion)
> MC
    prediccion
      No  Si
  No  93  16
  Si  15 208
```

Figura 41. Datos de la Matriz de confusión para el método de Redes Neuronales. Fuente propia.

Por último, se establecen los niveles de acierto y el porcentaje de error respectivo, presentados en la Fig. 42., con un 90,66% en la precisión del vocabulario de Cibercrimen y un 9% de error.

```
> # Porcentaje de buena clasificaciÃ³n
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.9066265
>
> error<-1-acierto
> error
[1] 0.09337349
> ## Fin del mÃ©todo Redes Neuronales
>
```

Figura 42. Porcentaje de aciertos método de Redes Neuronales.

4.2.8 Método de Adaboost.

La Fig. 43, presenta el desarrollo de un script en RStudio para el método AdaBoost, con el propósito de realizar pruebas de detección de Cibercrimen con el dataset establecido a través de la herramienta ADVI. Cabe resaltar, que este método, presentó una gran validación de los resultados y entrego un muy buen resultado con la interpretación del dataset referente al Cibercrimen.

```
##---------------Método Adaboost----------------------------
#1.8 MÃ©todo Adaboost
# El paquete 'adabag' ofrece la funciÃ³n 'boosting'.
suppressWarnings(suppressMessages(library(adabag)))

#Generamos el modelo, especificando la variable discriminante
modelo <- boosting(Correcta~., data=taprendizaje, boos=TRUE, mfinal=15)
modelo


#Generamos los valores predichos por el modelo en los datos de pureba
prediccion <- predict(modelo, ttesting)
prediccion

## Matriz de Confusion
MC<-table(ttesting$Correcta,prediccion$class)
MC

# Porcentaje de buena clasificaciÃ³n
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del mÃ©todo Ada Boost
```

Figura 43. Código del método AdaBoost para evaluar los dataset. Fuente propia.

La Fig. 44, muestra los resultados de la matriz de confusión con la información acerca de las predicciones realizadas por el sistema de clasificación en el método Adaboost.

```
>
> ## Matriz de Confusion
> MC<-table(ttesting$Correcta,prediccion$class)
> MC

     No  Si
  No 99   9
  Si  5 219
```

Figura 44. Datos de la Matriz de confusión para el método de Adaboost. Fuente propia.

Finalmente, se establecen los niveles de acierto y el porcentaje de error respectivo presentados en la Fig. 45. Estos resultados plantean hasta ahora el mejor

acierto entre los distintos métodos evaluados, con un 95,78% en la precisión del vocabulario de Cibercrimen y con tan sólo un error del 4%.

```
>
> ## Matriz de Confusion
> MC<-table(ttesting$Correcta,prediccion$class)
> MC

     No  Si
  No 99   9
  Si  5 219
>
> # Porcentaje de buena clasificaciÃ³n
> acierto<-(sum(diag(MC)))/sum(MC)*100
> acierto
[1] 95.78313
>
> error<-100-acierto
> error
[1] 4.216867
> ## Fin del mÃ©todo Ada Boost
```

Figura 45. Porcentaje de aciertos método de AdaBoost. Fuente propia.

4.2.9 Método C 4.5.

La Fig. 46, contiene el desarrollo del script del método C.45 en Rstudio, con el propósito de realizar las pruebas para determinar el porcentaje de predicción de Cibercrimen que arroja esta técnica. Se hace resaltar que, el método mencionado, presentó una gran complejidad por el tipo de bibliotecas necesarias para su ejecución.

```
##---------------Metodo C 4.5----------------------------

# Metodo C4.5

# Necesitamos Rweka para obtener el algoritmo J48(C4.5) en R
library(rJava)
library(RWeka)
library(caret)
modelo <- train(Correcta ~., method="J48", data=taprendizaje,
                tuneLength = 5,
                trControl = trainControl(
                  method="cv"))

#Generamos los valores predichos por el modelo en los datos de prueba
prediccion <- predict(modelo, ttesting)
prediccion

## Matriz de Confusion
MC<-table(ttesting$Correcta,prediccion)
MC

# Porcentaje de buena clasificacion
acierto<-(sum(diag(MC)))/sum(MC)*100
acierto

error<-100-acierto
error
## Fin del metodo C4.5
```

Figura 46. Código del método C4.5 para evaluar los dataset. Fuente propia

La Fig. 47, proyecta los resultados de la matriz de confusión y se establecen los niveles de acierto y el porcentaje de error respectivo con la información acerca de las predicciones realizadas por el sistema de clasificación en el método C4.5., el cual arroja un 90% en la precisión del vocabulario de Cibercrimen.

```
> ## Matriz de Confusion
> MC<-table(ttesting$Correcta,prediccion)
> MC
    prediccion
      No  Si
  No  95  14
  Si   7 216
>
> # Porcentaje de buena clasificaciÃ³n
> acierto<-(sum(diag(MC)))/sum(MC)
> acierto
[1] 0.936747
>
> error<-1-acierto
> error
[1] 0.06325301
> ## Fin del mÃ©todo C4.5
```

Figura 47. Matriz de Confusión y Porcentaje de aciertos método de C 4.5.

4.3. Resultados.

El análisis de resultados se describe a continuación, el cual está organizado de acuerdo al procedimiento general del ensayo.

1. Construcción del vocabulario de Cibercrimen.

   Al buscar la respuesta a las dificultades o retos que reporta la literatura en la construcción de vocabularios u ontologías semánticas para la clasificación de páginas Web nos encontramos con el reto de establecer cuál o cuáles vocablos son los más apropiados para este fin, es aquí donde los libros: “Cibercrimen, (Medina & Molist, 2015)” y “Delitos en la Red, (Poveda, 2015)”, se convirtieron en los textos guías para este fin.

2. Determinar cuál de las técnicas seleccionadas de aprendizaje supervisado, es la más apropiada para la detección de Cibercrimen.

   Como resultado se obtiene, que el algoritmo de AdaBoost, obtuvo el porcentaje de predicción más alto, con un 95% de precisión, comparadas con K-vecinos más cercanos, Naïve Bayes, Random Forest, Arboles de Decisión, Máquinas de Soporte Vectorial, Regresión Lineal, Redes Neuronales, y C.4.5.

3. Cabe señalar que las técnicas mencionadas, presentaron una efectividad en la detección del vocabulario de Cibercrimen superior al 80% de precisión.

4. Retos identificados que se presentan con las técnicas de machine learning para detectar vocabulario en grandes conjuntos de datos.

   - Dificultad para extraer los posibles datos para ser depurados y llegar a ser evaluados.
   - Otro reto de gran importancia se da en el poder identificar la técnica de aprendizaje de máquina más apropiada, para el caso de la identificación del vocabulario de Cibercrimen fue el modelo AdaBoost; mas no quiere inferir esto que en otros corpus lingüísticos o diferentes ontologías sea el método igual de efectivo.

4.4. Discusión.

La Tabla 3, presenta un concentrado de los algoritmos de Aprendizaje Supervisado seleccionados para realizar pruebas para la detección del vocabulario de Cibercrimen, resaltando la técnica usada, el porcentaje de precisión y el porcentaje de error. Además de resaltar con negritas el resultado mayor y el resultado menor, en esta prueba de ensayo. En donde se obtuvo el porcentaje mayor, con un 95.78% de acierto con el algoritmo AdaBoost conocimiento pleno acerca del tipo al que pertenecen cada uno delos datos que se van a utilizar y el porcentaje menor, con un 80.12% de acierto con el algoritmo K-vecinos más cercanos.

Tabla 3. Porcentajes de acierto y error de las distintas técnicas de aprendizaje.

Técnica utilizada	% de Acierto	% de Error
Método K-vecinos más cercanos K-NN	**80,12**	**19,88**
Naïve Bayes	81,92	18,08
Random Forest	93,67	6,33
Arboles de Decisión	86,75	13,25
Máquinas de Soporte Vectorial (SVM)	90,06	9,94
Regresión Lineal	90,06	9,94
Redes Neuronales	90,66	9,34
Adaboost	**95,78**	**4,21**
C 4.5	93,67	6,33

De acuerdo a los resultados mostrados en la Tabla 3, podemos observar que los algoritmos con mejor porcentaje de acierto son en su respectivo orden, son: Adaboost, Random Forest y C 4,5.

Por otro lado, cabe señalar que el algoritmo que se aleja mucho de ser un porcentaje realmente relevante es el de método K-vecinos más cercanos, con un porcentaje de acierto del 80%.

Al comparar los datos obtenidos, en general se determinó que los distintos algoritmos presentaron resultados muy superiores a los de la investigación de (Mbaziira & Jones, 2016), ya que en la investigación actual los valores de acierto se encuentran por encima del 80% y llegando a un valor excelente, como es el caso del 95% con el algoritmo AdaBoost, mientras que en la investigación de por (Mbaziira & Jones, 2016) no superan el 65%.

4.4.1. Discusión de los objetivos.

Con los resultados obtenidos en la investigación, se logran alcanzar los objetivos planteados para la investigación, los cuales se discuten a continuación.

1. Demostrar la utilidad de distintas técnicas de aprendizaje supervisado para la detección de vocabulario y la clasificación de las páginas Web.

Para la detección de vocabulario de Cibercrimen, se aplicaron los algoritmos de

aprendizaje supervisado: K-vecinos más cercanos, Naïve Bayes, Random Forest, Arboles de Decisión, Máquinas de Soporte Vectorial, Regresión Lineal, Redes Neuronales, AdaBoost y C.45., con los cuales se obtuvieron altos porcentajes, oscilando del 80 al 96% de precisión en la clasificación de las páginas Web, con lo cual se demuestra la utilidad al aplicar distintas técnicas de aprendizaje supervisado en la clasificación de los sitios Web.

2. Realizar un proceso de análisis de los datos con el fin de obtener conocimiento y valor agregado en el proceso de aprendizaje.

   La metodología utilizada en el presente estudio, describe los pasos de inicio a fin para la obtención de conocimiento y valor agregado de la información de Internet. Iniciando con la obtención de las páginas Web, seguido con el preprocesamiento de datos a través de la herramienta ADVI, integrando los procesos de Crawler, Procesamiento de Lenguaje Natural, y generando un dataset para las pruebas de inteligencia artificial, mediante algoritmos de aprendizaje supervisado, obteniendo como resultado un modelo y el porcentaje de precisión en la clasificación de las páginas Web, descubriendo conocimiento útil para la toma de decisiones.

3. Realizar las pruebas de predicción de vocabulario usando un dataset y determinar su porcentaje de eficiencia para detectar los términos de vocabulario de Cibercrimen.

Cabe resaltar que se utilizaron nueve algoritmos de aprendizaje supervisado para analizar el dataset construido para las pruebas de predicción de Cibercrimen, en donde se obtuvieron los siguientes resultados: K-vecinos más cercanos (80.12%), Naive Bayes (81.92%), Random Forest (93.67%), Arboles de Decisión (86.75%), Máquinas de Soporte Vectorial (90.06%), Regresión Lineal (90.06%), Redes Neuronales (90.66), AdaBoost (95.78%) y C.45 (93.67%).

4. Obtener la precisión en la detección de vocabulario de cibercrimen a través de diferentes técnicas de aprendizaje.

Destacar que el algoritmo que descubrió mejores resultados en la clasificación del Cibercrimen en sitios Web provenientes de Internet, es el algoritmo de aprendizaje supervisado AdaBoost, con una precisión del 95.78% en su predicción. Haciendo notar, que los resultados fueron validados con técnicas de validación cruzada.

4.4.2. Discusión de la hipótesis.

De acuerdo con la hipótesis planteada: “Es posible predecir vocabulario de Cibercrimen, clasificar sitios Web, y obtener valor agregado sobre las páginas que circulan en Internet, a través de técnicas de la analítica de Big Data, Procesamiento de Lenguaje Natural, Web Semántica, y Aprendizaje de Máquina (Aprendizaje Supervisado)”. Y a los resultados obtenidos en esta investigación, se puede aseverar, que, en efecto, es posible llegar a predecir vocabulario referente a Cibercrimen. Es importante mencionar, que los vocablos utilizados fueron delimitados a términos del idioma español y con las técnicas

mencionadas en la hipótesis, se lograron obtener porcentajes de acierto bastantes significativos, con un 95% de precisión, lo cual da sustento a la presente investigación.

Capítulo V. Conclusiones.

5.1. Conclusiones generales.

La presente investigación se enfoca en el Análisis, Clasificación y Predicción del Vocabulario de Cibercrimen en Internet Usando Modelos Predictivos de Machine Learning. Para esto se tomó el dataset generado por la herramienta ADVI y luego se evaluó con las técnicas de Método K-vecinos más cercanos, Naïve Bayes, Random Forest, Arboles de Decisión, Máquinas de Soporte Vectorial (SVM), Regresión Lineal, Redes Neuronales, Adaboost y C 4.5.

Los métodos con menor efectividad fueron Método K-vecinos más cercanos y Naïve Bayes con una efectividad del 80%, y aunque cumplen a cabalidad con el objetivo de detectar vocablos referentes a Cibercrimen se presentaron otros métodos con una mayor efectividad.

Se puede determinar que el algoritmo con mayor tasa de efectividad en el proceso de detectar los términos referentes a Cibercrimen a partir del corpus lingüístico determinado fue el algoritmo de AdaBoost; el cual presento un porcentaje de efectividad por encima del 95% incluso superando al algoritmo de Random Forest que es uno de los más recomendados entre las técnicas de machine learning.

Es de alta relevancia tener en cuenta la efectividad de las distintas técnicas de máquinas de aprendizaje utilizadas, como: AdaBoost, C 4.5. y Random Forest, los cuales

fueron superior al 93% para detectar vocablos referentes a Cibercrimen sin dejar de lado que el análisis se estructuro para palabras en el idioma español.

5.2. Ventajas de la investigación.

Una de las grandes ventajas, se enfoca en la gran variedad de algoritmos utilizados, ya que esto permitió poder evaluarlos y poder obtener resultados bastante alentadores en la detección de vocabularios referentes a Cibercrimen; al no enfocar la investigación en un único método, se pudo establecer que, por ejemplo, el método de Random Forest que es uno de los favoritos en investigación en este caso fue desplazado por el algoritmo de AdaBoost con mejores aciertos.

Una limitante en este tipo de investigación, está en poder determinar cuál o cuáles vocablos incluir, y que además estén relacionados al problema de investigación, y al tema como caso de estudio, ya que, si no se cuenta con un **corpus lingüístico adecuado sobre el tema de Cibercrimen,** no podría avanzarse de manera correcta.

Por otro lado, una restricción referente al corpus lingüístico está en que sólo se da en idioma español y por lo pronto se limita a este lenguaje.

5.3. Trabajos futuros.

El uso de las distintas técnicas de machine learning esgrimidas en la presente investigación para la detección de vocabulario referente a Cibercrimen, se convierten en herramientas importantes en la lucha actual para contrarrestar los cibercriminales; la aplicabilidad de estas técnicas está delimitada en la obtención de distintos corpus lingüísticos y los dataset para poder buscar términos referentes a una actividad determinada. Aunque el enfoque de la investigación está dado para el corpus de cibercriminales, si a futuro se desea detectar vocablos referentes a un producto, lugar o tema distinto se podrían utilizar estas técnicas para determinar su efectividad.

Referencias.

Alami, S., & Elbeqqali, O. (2015). Cybercrime profiling: Text mining techniques to detect and predict criminal activities in microblog posts. *2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015*. https://doi.org/10.1109/SITA.2015.7358435

AlvaroV96. (2016). Arbol de decision.png - Wikipedia. Retrieved April 1, 2019, from https://es.wikipedia.org/wiki/Archivo:Arbol_de_decision.png

Aprilla, C, D., Baskoro, D. A., Ambarwati, L., & Wicaksana, I. W. S. (2013). Data Mining dengan Rapid Miner, 139.

Arcila-Calderón, C., Barbosa-Caro, E., & Cabezuelo-Lorenzo, F. (2016). Técnicas big data: análisis de textos a gran escala para la investigación científica y periodística. *El Profesional de La Información*, *25*(4), 623. https://doi.org/10.3145/epi.2016.jul.12

Arimetrics. (2016). Qué es Crawler - Definición de Crawler. Retrieved May 10, 2019, from https://www.arimetrics.com/glosario-digital/crawler

Arredondo, M. (2008). No Title. Retrieved from http://www.fbi.gov/about-us/history/famous-cases/anthrax-amerithrax/08-489-m-01.pdf/at_download/file

Basher, A. R. M. A., & Fung, B. C. M. (2014). Analyzing topics and authors in chat logs for crime investigation. *Knowledge and Information Systems*, *39*(2), 351–381. https://doi.org/10.1007/s10115-013-0617-y

Blum, A. (2003). Tutorial FOCS'03 sobre Teoría del Aprendizaje Automático. Retrieved March 24, 2019, from https://www.cs.cmu.edu/~avrim/Talks/FOCS03/

Castillo Zuñiga, I., Luna Rosas, F., Muñoz Arteaga, J., Lopez Veyna, J. . (2016). Architecture (ADVI) for the detection of cyberbullying vocabulary in internet combining techniques of big data analytics and semantic., *3*, 12. https://doi.org/http://dx.doi.org/10.6036/NT8032

Cerón Guzmán, J. A., & León, E. (2015). Detecting Social Spammers in Colombia 2014 Presidential Election. Cuernavaca, Morelos, Mexico: 14th Mexican International Conference on Artificial Intelligence. https://doi.org/10.1007/978-3-319-27101-9_9

Chandra, B., Gupta, M., & Gupta, M. P. (2007). Robust Approach for Estimating Probabilities in Naïve-Bayes Classifier. In *Pattern Recognition and Machine*

Intelligence (pp. 11–16). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-77046-6_2

Cibercrimen en Colombia: balance de 2017. (n.d.). Retrieved April 19, 2019, from https://www.semana.com/nacion/articulo/cibercrimen-en-colombia-balance-de-2017/551979

Consejo nacional de política económica y social. (2016). CONPES 3854 - Politica Nacional de Seguridad Digital. Retrieved from https://colaboracion.dnp.gov.co/CDT/Conpes/Económicos/3854.pdf

Deloche, F. (2017). Archivo: Red neuronal recurrente unfold.svg - Wikimedia Commons. Retrieved April 21, 2019, from https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg

Ferrero, R. (2018). Qué es R Software | Máxima Formación. Retrieved May 24, 2019, from https://www.maximaformacion.es/blog-dat/que-es-r-software/

Gabinete de comunicación UPM. (2015). Big Data: El valor de los datos. *31*. Retrieved from https://www.etsisi.upm.es/sites/default/files/noticias_tic/BigData_RevistaUPM.pdf

Glosser.ca. (2013). File:Colored neural network.svg - Wikimedia Commons. Retrieved April 21, 2019, from https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg

González, A. (2014). Conceptos básicos de Machine Learning -. Retrieved April 1, 2019, from https://cleverdata.io/conceptos-basicos-machine-learning/

González Pacheco, V. (2019). Big Data & Data Science Blog: Una Breve Historia del Machine Learning. Retrieved April 1, 2019, from https://data-speaks.luca-d3.com/2018/11/una-breve-historia-del-machine-learning.html

Gutiérrez Esparza, G., Margain Fuentes, L., Canul Reich, J., & Ramírez del Real, T. A. (2017). Un modelo basado en el Clasificador Naïve Bayes para la evaluación del desempeño docente. *RIED. Revista Iberoamericana de Educación a Distancia*, *20*(2), 293. https://doi.org/10.5944/ried.20.2.17717

Gyanchandani, M., Yadav, R. N., & Rana, J. L. (2010). Intrusion Detection using C4 . 5 : Performance Enhancement by Classifier Combination. *ACEEE Int. J. on Signal & Image Processing, 01*(03), 46–49.

Heredia, B., Khoshgoftaar, T. M., Prusa, J., & Crawford, M. (2016). An Investigation of Ensemble Techniques for Detection of Spam Reviews. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 127–133). IEEE. https://doi.org/10.1109/ICMLA.2016.0029

Hinton, G., & Salakhutdinov, R. (2009, April 15). Deep Boltzmann Machines. Retrieved from http://proceedings.mlr.press/v5/salakhutdinov09a.html

Hofmann, M., & Klinkenberg, R. (2014). *Data Mining and Knowledge Discovery Series : RAPID MINER Data Mining Use Cases and Business Analytics Applications*. (M. Hofmann & R. Klinkenberg, Eds.). Florida. https://doi.org/78-1-4822-0550-3

HRcommons. (2010). Perceptron 4layers.png - Wikimedia Commons. Retrieved April 21, 2019, from https://commons.wikimedia.org/wiki/File:Perceptron_4layers.png

Iars.geo. (2019). Breve Historia de las Redes Neuronales Artificiales (Articulo 1) — Steemit. Retrieved March 31, 2019, from https://steemit.com/spanish/@iars.geo/breve-historias-de-las-redes-neuronales-artificiales-articulo-1

Jagannath, V. (2017). Random forest diagram complete.png - Wikimedia Commons. Retrieved April 21, 2019, from https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png

Jo, T. (2019). *Text Mining Concepts, Implementation, and Big Data Challenge*. (J. Kacprzyk & W. Poland, Eds.), *Springer* (Vol. 45). Springer. https://doi.org/10.1007/978-3-319-91815-0

Kaur, G., & Singla, A. (2016). Sentimental Analysis of Flipkart reviews using Naïve Bayes and Decision Tree algorithm, *5*(1), 148–153. Retrieved from http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-1-148-153.pdf

Madala, D. S. V., Gangal, A., Krishna, S., Goyal, A., & Sureka, A. (2018). An empirical analysis of machine learning models for automated essay grading. *PeerJ Preprints*, *6*, e3518v1. https://doi.org/10.7287/peerj.preprints.3518v1

Mani, S., Kumari, S., Jain, A., & Kumar, P. (2018). Spam Review Detection Using Ensemble Machine Learning (pp. 198–209). https://doi.org/10.1007/978-3-319-96133-0_15

MathWorks. (2016). ¿Qué es una red neuronal? - MATLAB & Simulink. Retrieved

April 2, 2019, from https://la.mathworks.com/discovery/neural-network.html

Mayer-Schonberger, V., & Kenneth, C. (2013). *BIG DATA A Revolution Will transform How We Live, Work, and Think*. *Houghton*. New York.

Mbaziira, A., & Jones, J. (2016). A Text-based Deception Detection Model for Cybercrime, (December), 1–8.

Medina, M., & Molist, M. (2015). *CIBERCRIMEN: ¡Protégete del Bit-Bang!* (1st ed.). Barcelona. Retrieved from Tibidabo Ediciones, SA.

Merchán Macías, F. J. (2018). *Ancert: aplicación de técnicas de maching learning a la seguridad*. UNIVERSITAT OBERTA DE CATALUNYA UNIVERSITAT AUTÓNOMA DE BARCELONA UNIVERSITAT ROVIRA I VIRGILI. Retrieved from http://hdl.handle.net/10609/88925

Miró Llinares, F. (2012). El cibercrimen. Fenomenología y criminología de la delincuencia en el ciberespacio. *Revista Para El Análisis Del Derecho*.

Moise, A. C. (2014). Some Considerations on the Phenomenon of Cybercrime. *Journal of Applied Economic Sciences Quarterly*, *V*(1(9)). Retrieved from https://www.econbiz.de/Record/some-considerations-on-the-phenomenon-of-cybercrime-moise-adrian-cristian/10011191590

Moohebat, M., Raj, R. G., Kareem, S. B. A., & Thorleuchter, D. (2015). Identifying ISI-indexed articles by their lexical usage: A text analysis approach. *Journal of the Association for Information Science and Technology*, *66*(3), 501–511. https://doi.org/10.1002/asi.23194

Mosquera, R., Castrillón, O. D., & Parra, L. (2018). Máquinas de Soporte Vectorial , Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos ., *29*(6), 153–162. https://doi.org/10.4067/S0718-07642018000600153

Ortega, M. (2017). Fuerzas Armadas de Colombia contrarrestan ciberdelincuencia :: Dialogo Americas. Retrieved March 24, 2019, from https://dialogo-americas.com/es/articles/colombian-armed-forces-counter-cybercrime

Peng, Q., & Zhong, M. (2014). Detecting Spam Review through Sentiment Analysis. *Journal of Software*, *9*(8). https://doi.org/10.4304/jsw.9.8.2065-2072

Planeación, D. N. de. (2011). Documento Conpes 3701 de julio 14 de 2011, 43. Retrieved

from http://www.mintic.gov.co/index.php/docs-normatividad?task=download.file&fid=46.741&sid=54

Poveda Criado, M. A., & Sotos Sepúlveda, J. (2015). *Delitos en la red : cibercrimen, ciberdelitos, ciberseguridad, ciberespionaje y ciberterrorismo*. Fragua. Retrieved from https://www.dykinson.com/libros/delitos-en-la-red/9788470746826/

Quezada Lucio, N. (2017). *K-Vecino más próximo en una aplicación de clasificación y predicción en el Poder Judicial del Perú*. *Pesquimat*. https://doi.org/10.15381/pes.v21i1.15077

Rochina, P. (2017). ¿Qué es el Text Mining? Aplicaciones de la Minería de Texto. Retrieved May 10, 2019, from https://revistadigital.inesem.es/informatica-y-tics/text-mining/

Rodriguez, O. (2013). Aprendizaje Supervisado K - Vecinos más cercanos.

Rodríguez Rama, J. M. (2018). *APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING A LA DETECCIÓN DE ATAQUES*. Retrieved from http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81126/11/jmrodriguez85TFM0618memoria.pdf

Sameera, K., & Vishwakarna, P. (2017). *Cybercrime: To Detect Suspected User ' s Chat Using Text Mining*. *(Ictis 2017)* (Vol. 2). Mumbai: Springer Singapore. https://doi.org/10.1007/978-981-13-1742-2

Sammut, C., & Webb, G. I. (Eds.). (2017). *Encyclopedia of Machine Learning and Data Mining*. Boston, MA: Springer US. https://doi.org/10.1007/978-1-4899-7687-1

Sánchez Medero, G. (2013). *Revista Cenipec.* Retrieved from http://www.saber.ula.ve/handle/123456789/36770

Serrano-cobos, J. (2014). BigData y analitica Web, 561–566. Retrieved from http://recyt.fecyt.es/index.php/EPI/article/download/epi.2014.nov.01/16929

Shavers, B. (2013). *Cybercrime Investigation Case Studies*. *Syngress*. Kidlington.

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis (pp. 337–349). https://doi.org/10.1007/978-3-642-00958-7_31

Uccelli, P., Dobbs, C. L., & Scott, J. (2013). Mastering Academic Language. *Written Communication*, *30*(1), 36–62. https://doi.org/10.1177/0741088312469013

Vidueira Ferreira, J. E., da Costa, C. H. S., de Miranda, R. M., de Figueiredo, A. F., Ferreira, J. E. V., da Costa, C. H. S., … de Figueiredo, A. F. (2015). The use of the k nearest neighbor method to classify the representative elements. *Educación Química*, *26*(3), 195–201. https://doi.org/10.1016/j.eq.2015.05.004

Vinco, S. (2017). Kernel Machine.svg - Wikimedia Commons. Retrieved April 21, 2019, from https://commons.wikimedia.org/wiki/File:Kernel_Machine.svg

Wang, Z. (2015). Ilustración del algoritmo de AdaBoost para crear un clasificador fuerte ... | Descargar Scientific Diagram. Retrieved April 21, 2019, from https://www.researchgate.net/figure/Illustration-of-AdaBoost-algorithm-for-creating-a-strong-classifier-based-on-multiple_fig9_288699540

William L. Hosch. (2009). Clasificadores Débiles - AdaBoost. *Britannica Articles*, 21–31. Retrieved from https://www.britannica.com/technology/machine-learning#Article-History

Wu, D., Sakr, S., & Zhu, L. (2017). Big Data Storage and Data Models. In *Handbook of Big Data Technologies* (pp. 3–29). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-49340-4_1

Yang, Z., Lin, H., & Wu, B. (2009). BioPPIExtractor: A protein–protein interaction extraction system for biomedical literature. *Expert Systems with Applications*, *36*(2), 2228–2233. https://doi.org/10.1016/j.eswa.2007.12.014

Zhang, X. (2017). Support Vector Machines. In *Encyclopedia of Machine Learning and Data Mining* (pp. 1214–1220). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4899-7687-1_810