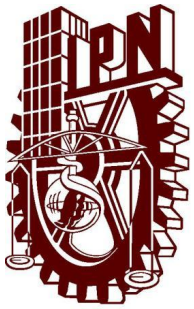

INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Laboratorio de Ciberseguridad

Selección de características para la detección de botnets

TESIS

QUE PARA OBTENER EL GRADO DE:

Maestría en Ciencias de la Computación

P R E S E N T A:

Ing. Francisco Villegas Alejandre

DIRECTORES DE TESIS:
Dr. Eleazar Aguirre Anaya
Dra. Nareli Cruz Cortés

Ciudad de México Diciembre 2016

Centro de Investigación
en Computación
Instituto Politécnico Nacional

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 13:00 horas del día 02 del mes de diciembre de 2016 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Selección de características para la detección de botnets"

Presentada por el alumno:

VILLEGAS	**ALEJANDRE**	**FRANCISCO**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	4	0	5	7	2

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Eleazar Aguirre Anaya

Dra. Nareli Cruz Cortés

Dr. Ponciano Jorge Escamilla Ambrosio

Dr. Moisés Salinas Rosales

Dr. Ricardo Menchaca Méndez

M. en C. Sandra Dinora Orantes Jiménez

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Marco Antonio Ramírez Salinas

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 6 del mes diciembre del año 2016, el (la) que suscribe Francisco Villegas Alejandre alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B140572, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Eleazar Aguirre Anaya y Dra. Nareli Cruz Cortés y cede los derechos del trabajo intitulado Selección de características para la detección de botnets, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección francisco_7766@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Francisco Villegas Alejandre

Nombre y firma

Resumen

En esta investigación se presenta un nuevo método de selección de características para detectar botnets en su etapa de Comando y Control (C&C). Un problema importante es que los investigadores del área han propuesto el uso de un conjunto de características para la detección de botnets en su etapa de C&C con base en su experiencia, sin embargo, no existe un método de selección adecuado que encuentre el mejor conjunto, debido a que el espacio de búsqueda está relacionado con la cantidad de características. A medida que se incrementa la cantidad de características bajo análisis, el espacio de búsqueda también se incrementa, por lo cual un método de búsqueda exhaustiva no resultaría viable. Para resolver este problema, se diseña una propuesta de selección de características de las conexiones de las botnets en su etapa de C&C a partir de un algoritmo genético combinado con el clasificador C4.5 para mejorar la tasa de detección. La propuesta de selección de características es la principal aportación en esta investigación. Un algoritmo genético (GA) se utiliza para seleccionar el conjunto de características que ofrece la mayor tasa de detección. Se utilizó el algoritmo C4.5 de aprendizaje automático, este algoritmo clasifica las conexiones, pertenecientes o no a una botnet. Los datos utilizados en este trabajo fueron extraídos de los repositorios ISOT e ISCX. Se realizaron pruebas para obtener los mejores parámetros en un algoritmo genético y el algoritmo C4.5. También se realizaron experimentos con el fin de obtener el mejor conjunto de características para cada botnet (específica), y para cada tipo de botnet (general). Se obtiene una reducción considerable de características y una tasa de detección más alta que trabajos representativos en el estado del arte.

Abstract

In this research, a novel method to do feature selection to detect botnets at the phase of Command and Control (C&C) is presented. A major problem is that researchers have proposed features based on their expertise, but there is no suitable selection method that finds the best set, because the search space is related to the number of features. As the amount of characteristics under analysis increases, the search space is also increased, whereby an exhaustive search method would not be feasible. With this aim, a proposal of feature selection of botnet connections in its C&C stage is designed from a genetic algorithm combined with the C4.5 classifier to improve the detection rate. is defined the feature set based on conections of botnets at their phase of C&C, that maximizes the detection rate of these botnets. The proposal of feature selection is the main contribution in this research. A Genetic Algorithm (GA) was used to select the set of features that gives the highest detection rate. We used the machine learning algorithm C4.5, this algorithm does the classification between connections belonging or not to a botnet. The datasets used in this paper were extracted from the repositories ISOT and ISCX. Tests were done to get the best parameters in a GA and the algorithm C4.5. We also performed experiments in order to obtain the best set of features for each botnet analyzed (specific), and for each type of botnet (general) too. Considerable reduction of features and a higher detection rate than representative works of the state of art were obtained.

Agradecimientos

Mi más sincero agradecimiento a:

· El Centro de Investigación en Computación (CIC), el Consejo Nacional de Ciencia y Tecnología (CONACyT) y al Instituto Politécnico Nacional (IPN) por el apoyo durante el proceso de investigación.

· A mis asesores Eleazar Aguirre Anaya y Nareli Cruz Cortés por la paciencia, la orientación y por todo el apoyo que me dieron durante el proceso de investigación.

· A mi familia, en especial, mis padres Francisco Villegas Santoyo y Veronica Alejandre Corona que siempre me han apoyado en todo cuando lo he necesitado, mi tío Cristobal Villegas Santoyo por sus consejos, ayuda y platicas, de igual manera mis hermanos Veronica y Cristobal.

· A mi amor Karen Ruby Ochoa Barajas que siempre me escucho y me impulso a terminar mi investigación.

· A mis amigos de la maestría que sin ellos no hubiese salido de varios problemas.

· Al Dr. Ricardo Menchaca Méndez por su ayuda y consejos.

· Y por último a dios y a la vida.

Sin ustedes este proceso de investigación no hubiese sido posible.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Debido al crecimiento del Internet, el número de personas que cometen cibercrimen también ha aumentado. Las botnets son una de las herramientas utilizadas por algunos usuarios para atacar el Internet.

Las botnets están compuestas por bots (dispositivos infectados) que son controlados a distancia por un maestro operador de la bot a través de un canal de Comando y Control (C&C). Cada par se comunica entre sí, es decir bots y el maestro de la bot.

Las botnets se utilizan para realizar diferentes tipos de ataques como denegación de servicio distribuido (DDoS), el robo de credenciales, spam, phishing, etc. Symantec elaboró un estudio sobre los países afectados por las botnets mundialmente [2] y en Latinoamérica [3], siendo Brasil el país más afectado en el estudio latinoamericano y China como el país más afecto mundialmente, donde en el 2014 la cantidad de dispositivos detectados por Symantec fue de 1.9 millones.

Las botnets se clasifican como centralizadas y descentralizadas. En las botnets centralizadas, los bots contactan periódicamente con el servidor C&C para recibir instrucciones. Algunos de los protocolos de comunicación utilizados son HTTP y el IRC. En las botnets descentralizadas, también llamadas "*peer to peer*" (P2P), sólo uno de los bots recibe el mensaje directamente desde el servidor C&C, a continuación, este bot es responsable de transmitir el mensaje a otros bots y los bots a más bots sucesivamente. Algunos protocolos de comunicación utilizados son Overnet, Kademlia y HTTP2P.

El ciclo de vida de las botnets se puede dividir en cuatro fases [1]: formación, C&C, de ataque y post-ataque.

1.- Durante la fase de formación, el maestro de la bot infecta a otros dispositivos a través de Internet, estos dispositivos infectadas se convierten ahora en los bots controlados por el maestro de la bot.

2.- Durante la fase de C&C los bots reciben instrucciones del maestro de la bot.

3.- Durante la fase de ataque, los bots realizan actividades maliciosas con base en las instrucciones recibidas.

4.- Durante la fase de post-ataque, algunos bots podrían ser detectados y eliminados, por esta razón el maestro de la bot analiza la botnet (ocasionalmente) para detectar los bots que permanecen activos.

La detección de botnets durante la fase C&C es muy importante sobre todo porque permite la detección de los bots antes de la fase de ataque, lo cual impide que la botnet realice actividades maliciosas. Por otra parte, si se identifican todos los servidores C&C la botnet se deshabilitaría.

En la investigación, se presenta una propuesta para definir el conjunto de características efectivas para detectar botnets en la fase de C&C por medio de conexiones entre dispositivos, las 19 características utilizadas fueron extraídas del estado del arte relacionado con los algoritmos para la detección de botnets en la fase de C&C. Las conexiones de red se utilizaron para definir el comportamiento de las botnets. Estas conexiones se utilizan para organizar los paquetes en un 5-tupla de la siguiente manera: < dirección IP de origen, dirección IP de destino, puerto de origen, puerto de destino, protocolo >.
El algoritmo genético (AG) se utiliza para seleccionar el conjunto de características con la mejor tasa de detección en las botnets en la fase de C&C, éste genera diversas soluciones que son guiadas por sus operadores para seleccionar este conjunto, el algoritmo que se encarga de evaluar cada solución que se genera por el AG es el algoritmo de aprendizaje automático C4.5.

Cabe destacar que este problema podría ser resuelto mediante búsqueda ex-

haustiva, pero se vuelve un caso de complejidad exponencial debido al espacio de solución ($2^{19} posibilidades$) y al conjunto de datos de entrenamiento. De forma exhaustiva el tiempo en el proceso se incrementa de manera considerable y por lo tanto el tiempo necesario para encontrar el mejor conjunto/-subconjunto de características con la mejor tasa de detección se incrementa. La propuesta en la tesis es capaz de guiar las evaluaciones por una ruta, con la cual se obtiene mediante un AG el mejor conjunto/subconjunto de características para la detección de botnets. Las evaluaciones de todas las posibilidades del espacio de solución, así como las evaluaciones realizadas en la propuesta y el tiempo en cada uno de los experimentos son descritos en el apartado 5.1.3.

1.1. Metodología

La metodología en la investigación corresponde al método científico, es flexible y por lo tanto se puede volver a cualquier etapa si así se requiere. Primero que nada, mediante la observación se identifica el problema y se plantean los objetivos para resolver el problema. En la segunda etapa se hace una revisión del estado del arte para verificar que y como se han resuelto problemas similares. En la tercera etapa se hace una construcción de la hipótesis que consiste en elaborar una explicación provisional de las observaciones o experiencias y sus posibles causas. La cuarta etapa es el diseño de la propuesta para comprobar la hipótesis, resolver el problema y cumplir con el objetivo. La quinta etapa es la experimentación para prueba de la hipótesis mediante la propuesta. La sexta etapa es el análisis de resultados y elaboración de conclusiones con base en la propuesta. En la séptima etapa si la hipótesis fue verdadera o falsa o parcial se puede elaborar el documento de tesis con el proceso y en caso de no ser verdadera se puede volver a intentar con una nueva hipótesis.

En la metodología hubo una primera hipótesis en la primera iteración, la cual resulto ser parcial, es por esto que hubo un retroceso hasta la primera etapa, en la cual la hipótesis planteada, los objetivos y el planteamiento fueron alcanzados exitosamente. El escrito en la tesis se enfoca en la segunda iteración. La Figura 1.1 ilustra la metodología, en la que se puede observar el retroceso de la primera iteración.

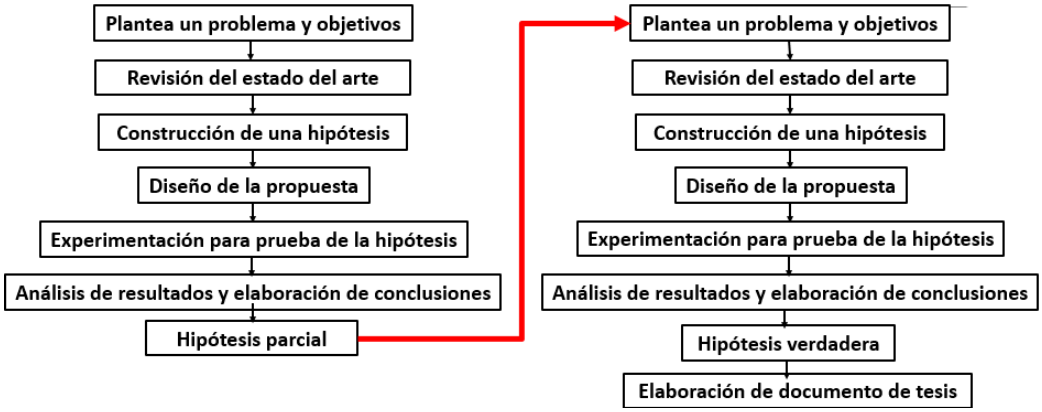

Figura 1.1: Metodología utilizada.

1.2. Planteamiento del problema

Los investigadores del área han propuesto el uso de un conjunto de características para la detección de botnets en su etapa de C&C con base en su experiencia, sin embargo, no existe un método de selección adecuado que encuentre el mejor conjunto, debido a que el espacio de búsqueda está relacionado con la cantidad de características. A medida que se incrementa la cantidad de características bajo análisis, el espacio de búsqueda también se incrementa, por lo cual un método de búsqueda exhaustiva no resultaría viable.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar una propuesta de selección de características de las conexiones de las botnets en su etapa de C&C a partir de un algoritmo genético combinado con el clasificador C4.5 para mejorar la tasa de detección.

1.3.2. Objetivos específicos

- Identificar un vector base de características a partir de los trabajos representativos del estado del arte y su representación que distinga el

trafico perteneciente o no a las botnets.

- Seleccionar el algoritmo clasificador con base en su tasa de detección y el de búsqueda con base en su eficiencia.
- Seleccionar el algoritmo clasificador con base en su tasa de detección y el de búsqueda con base en su eficiencia.
- Seleccionar el conjunto de datos a evaluar de botnets tipo centralizada y descentralizada.
- Integrar los componentes para la propuesta de selección de características y aplicarla a cada botnet y tipo de botnet.

1.4. Hipótesis

Existe un conjunto de características de las conexiones de las botnets en su etapa de C&C que tiene la mayor tasa de detección sobre los demás conjuntos y puede ser encontrado con ayuda de un algoritmo genético combinado con el clasificador C4.5.

1.5. Organización de la tesis

El resto de la tesis está organizada de la siguiente manera:

Capítulo 2. Marco teórico En este capítulo se presentan los conceptos necesarios para la elaboración de la tesis. Comenzamos con los conceptos generales de las botnets, su taxonomía y principales características, después se mencionan los conceptos generales del método de clasificación utilizado y su funcionamiento, finalizando con los conceptos generales sobre algoritmos genéticos y sus características.

Capítulo 3. Estado del arte. En este capítulo se presentan trabajos relacionados identificados en el estado del arte. Se mencionan tipos de métodos utilizados para la detección de botnets, algoritmos, donde en algunos de ellos se utiliza una selección arbitraria de características.

Capítulo 4. Propuesta de solución. En este capítulo se presenta la propuesta de solución en la tesis, los algoritmos utilizados y cómo interactúan para obtener el conjunto de características con la mayor tasa de detección de las botnets.

Capítulo 5. Experimentos y resultados. En este capítulo se presenta el diseño de experimentos, las herramientas utilizadas para el desarrollo de éstos, los datos utilizados, los resultados obtenidos, la comparación y un análisis de los mismos.

Capítulo 6. Conclusiones. En este capítulo se presenta las conclusiones obtenidas en este trabajo y el posible trabajo futuro.

Capítulo 2

Marco teórico

En este capítulo se presentan los conceptos necesarios para la elaboración de la tesis. Comenzamos con los conceptos generales de las botnets, su taxonomía y principales características, después se mencionan los conceptos generales del método de clasificación utilizado y su funcionamiento, finalizando con los conceptos generales sobre algoritmos genéticos y sus características.

2.1. Taxonomía de las botnets

Botnet es el nombre genérico que denomina a cualquier grupo de dispositivos infectados y controlados por un atacante de forma remota. Generalmente, una botnet se crea usando un malware que infecta a una gran cantidad de dispositivos. Los dispositivos infectados que son parte de la botnet, son llamados "bots" o "zombies". No existe un número mínimo de equipos para crear un botnet. Una botnet pequeña puede incluir cientos de dispositivos infectados, mientras que una botnet grande puede utilizar millones. Algunos ejemplos de botnets son Storm, Waledac, Neris y RBot [4].

El ciclo de vida de las botnets se puede dividir en cuatro fases [1]: formación, C&C, de ataque y post-ataque.

1.- Durante la fase de formación, el maestro de la bot infecta a otros dispositivos a través de Internet, estos dispositivos infectados se convierten ahora en los bots controlados por el maestro de la bot.

2.- Durante la fase de C&C los bots reciben instrucciones del maestro de la bot.

3.- Durante la fase de ataque, los bots realizan actividades maliciosas con base en las instrucciones recibidas.

4.- Durante la fase de post-ataque, algunos bots podrían ser detectados y eliminados, por esta razón el maestro de la bot analiza la botnet (ocasionalmente) para detectar los bots que permanecen activos.

Las botnets se han estado adaptando frente a nuevos mecanismos de detección en los últimos años. Sus características dependen de la manera en que se propagan, la arquitectura, el mecanismo de comunicación C&C y la manera en que realizan sus ataques. Es muy importante conocer las diferentes características de las botnets con el fin de detectarlos, se pueden propagar tan fácil como un gusano y además pueden evitar la detección como si fuesen un virus [5]. Las botnets se clasifican según el tipo de comunicación con su servidor C&C de la manera siguiente [5]:

2.1.1. Arquitectura Comando y Control

La arquitectura C&C se refiere a la comunicación entre el maestro de la botnet (atacante) y sus bots (zombies) que son los dispositivos infectados [5].

Se cree que el servidor C&C es el cuello de botella de las botnets, debido a que si se logra detectar entonces la botnet será deshabilitada [11]. Si algún mecanismo de detección como los IDS (Sistema de Detección de Intrusos) es capaz de detectar el servidor o los servidores C&C utilizados por el maestro de la botnet, entonces será posible deshabilitar esa botnet si se inhabilitan los servidores detectados, debido a que se inhabilitará la comunicación entre el maestro de la bot y los bots controlados. Sin ese enlace de comunicación la botnet no será capaz de lanzar ataques tan poderosos, ya que el poder de las botnets está directamente relacionado a su tamaño (número de bots disponibles) [5].

Dos tipos diferentes de Arquitecturas C&C han sido identificadas: Centrali-

zada y Descentralizada (P2P) [11].

2.1.1.1. Arquitectura centralizada

Como su nombre lo indica, el maestro de la bot elige un único nodo considerado como el punto central, en el que cada bot se conecta cuando es infectado. Este nodo debe contar con un gran ancho de banda, debido a que todas las comunicaciones pasan a través de ella. El maestro de la botnet puede ser ese nodo central, pero de ser así, tendría la desventaja de perder la dispersión en las comunicaciones. Por lo general, el nodo central es un host comprometido [11] [5].

Después de la infección y una vez que los bots se comunican con el servidor central C&C, el maestro de la botnet dará las órdenes a los bots, para lanzar ataques, infectar a otros dispositivos o simplemente devolver la información obtenida desde ese host [5]. Una representación de esta arquitectura se da en la Figura 2.1.

Esta arquitectura tiene una alta supervivencia en el mundo real debido a que aún no hay grandes contramedidas contra las botnets. A pesar de que los administradores de sistemas son más conscientes de este problema, se han implementado mecanismos como el de captchas[1] y listas negras [2] para evitar que los bots ataquen a ciertos sistemas, no todos implementan mecanismos para protegerse [5].

Otra de sus ventajas es la pequeña latencia de mensajería, la cual hace que sea mucho más fácil coordinar y poner en marcha los ataques. El principal inconveniente es que aquí sólo hay una estructura de C&C, es la única fuente de la conexión entre el maestro de la botnet y los bots. Así que una vez encontrado y desactivado, el maestro de la bot no tiene manera de comunicarse con la botnet y esto hace que la botnet quede inservible. Algunas botnets que utilizan esta arquitectura son RBot y Neris, las cuales utilizan el protocolo

[1] El captcha es una prueba desafío–respuesta utilizada en computación para determinar cuándo el usuario es o no humano [45].

[2] En Internet, una lista negra o *black list* es una lista donde se registran las direcciones IPs que tienen el acceso bloqueado. [44].

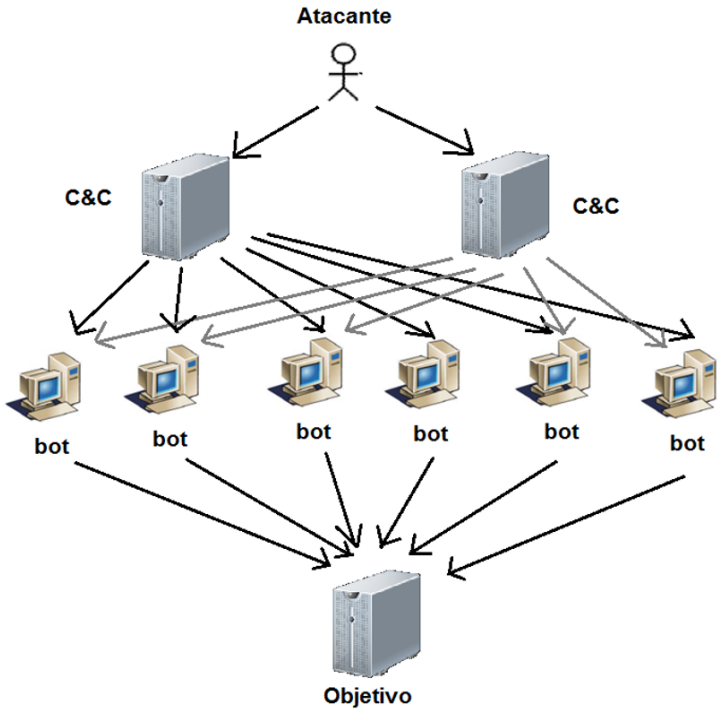

Figura 2.1: Representación de la arquitectura centralizada.

IRC [5].

La botnet RBot apareció en 2003 de acuerdo con el reporte MSRT de junio del 2006 de Microsoft ("MSRT: Progress Made,Trends Observed" Matthew Braverman), la botnet RBot tenía 1.9 millones de computadoras infectadas. Se trata de un troyano de puerta trasera que utiliza el protocolo de comunicación IRC para comunicarse con el servidor C&C. Esta botnet presentó la idea de utilizar más de una herramienta de cifrado de paquetes (por ejemplo, Morphine, UPX, ASPack, PESpin, EZIP, PEShield, PECompact, FSG, EXEStealth, PEX, MoleBox, and Petite). RBot utiliza un mecanismo propio para lograr infectar nuevos bots, analiza los sistemas en los puertos 139 y 445 (puertos abiertos de Microsoft). Además intenta adivinar contraseñas débiles, donde puede utilizar una lista predeterminada o una lista proporcionada por el maestro de la bot. También puede utilizar una lista predeterminada de identificadores de usuario y contraseñas, o usar una lista de ID de usuarios y contraseñas que se encuentran en otros sistemas [29].

2.1.1.2. Arquitectura descentralizada

En esta arquitectura el maestro de la botnet organiza los bots y los C&C como si se tratara de la red P2P. Puede ser estructurada o no estructurada [12]. Esta tecnología es mucho más resistente a la detección y desactivación que el modelo centralizado, debido a que no necesita un servidor de C&C ya que todos los bots pueden ser servidores de C&C por lo que una vez que uno se desactiva casi no tiene efecto sobre la desactivación de toda la botnet. Esta característica es la razón principal por esta arquitectura es una tendencia cada vez mayor entre la comunidad atacante. Su principal inconveniente es el problema de escalabilidad. Sólo soporta conversaciones entre grupos pequeños lo que hace que sea más difícil lanzar grandes ataques. Además, no hay garantía de entrega de mensajes y tiene una latencia más alta en la propagación de mensajes cuando se compara con el modelo centralizado. Esto trae serios problemas a la coordinación sobre cómo los bots despliegan el ataque, pero aporta un modelo mucho más robusto [5]. La Figura 2.2 muestra una posible organización para una estructura de este tipo.

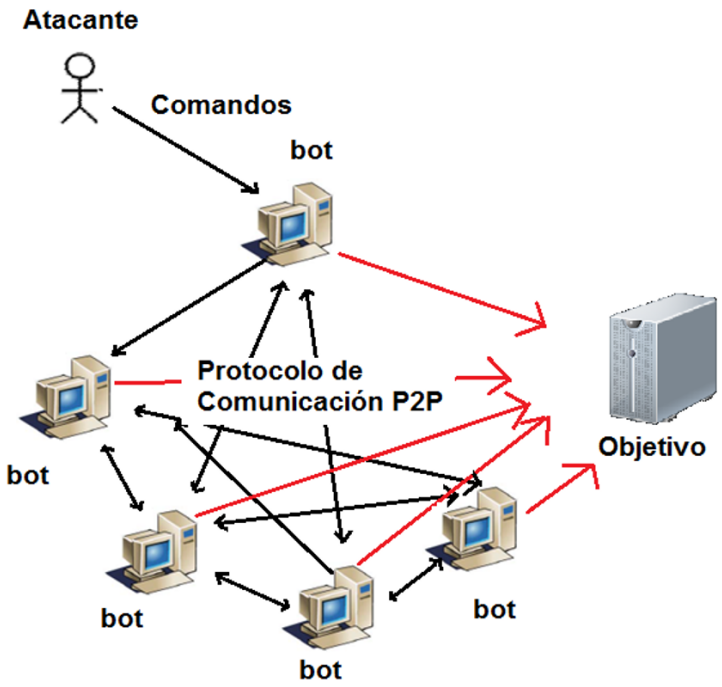

Figura 2.2: Representación de la arquitectura descentralizada.

2.1.1.2.1. Estructurada En un modelo estructurado el maestro de la botnet utiliza criterios y algoritmos para estructurar la forma en que los bots y los recursos se organizan. Esto se lleva a cabo para asegurarse de que cualquier bot puede encontrar su camino a otro, incluso cuando no están conectados directamente. Son muy similares a otra arquitectura C&C, llamada arquitectura al azar, en el que cada nodo tiene casi el mismo grado de importancia para la red [12] [5]. Cada nodo se conecta a sus k compañeros, donde k es un parámetro fijo. El valor de k depende de la topología de la red. La red se compone de un número de nodos que se comunican entre sí y almacenan información sobre los demás nodos. Las redes P2P estructuradas generalmente usan una tabla hash distribuida para organizar su lista de compañeros y su información [28]. Cada nodo es identificado de forma única con un identificador de nodo. Cada par en la red mantiene la información de contacto sobre otros compañeros en su tabla de enrutamiento [9]. Una de las redes usadas es la Overnet. Algunos botnets que utilizan esta arquitectura son Storm, Waledac.

La botnet Storm es un modelo bien conocido de una botnet descentralizada. Storm debe su nombre a su auto-propagación por spam [3] a través de correo electrónico, el cual hacía referencia a un desastre climático a principios de 2006 y tenía el asunto de: "230 dead as Storm batters Europe" [31]. Storm se convirtió en una noticia titular debido a la enorme cantidad de correos electrónicos no solicitados que generó a partir de entonces. La botnet tiene varios nombres debido a los antivirus e investigadores que la identificaron, algunos de ellos son: Peacomm (Symantec), Peed (BitDefender), Tibs (BitDefender), Dorf (Sphos), Nuwar (ESET) y Zhelatin (F-Secure y Kaspersky) [32] [33] [34] [35]. El mensaje contiene el binario de storm.exe como un archivo adjunto [32]. Aquellos que abrieron el archivo adjunto tienen sus ordenadores infectados y se convirtieron en una parte de la botnet. El 22 de enero de 2007, el malware Storm representó el 8 % de todas las infecciones de malware a nivel mundial. Storm utiliza técnicas de ingeniería social o métodos *drive-by-download* para su propagación. En el primer caso, el binario storm se instala haciendo clic en los enlaces como los de tarjetas de felicitación, los videojuegos o enlaces a "noticias importantes del día". En este último caso, el binario

[3]Spam significa correo basura o mensaje basura y hace referencia a los mensajes no solicitados, no deseados o con remitente no conocido (correo anónimo), habitualmente de tipo publicitario, generalmente son enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor [43].

se instala cuando el usuario visita un sitio web que explota vulnerabilidades de navegador web del usuario o de sus complementos. La misión principal de storm es la propagación de spam. Las redes basadas en Kademlia pueden tener más de un millón de usuarios simultáneos y se ha convertido en uno de los protocolos con base en DHT (*distributed hash table*) más usados. Storm botnet utiliza el protocolo Overnet que a su vez se basa en la Kademlia [36] [9].

La botnet Waledac surgió a finales de 2008 como un posible sucesor de la botnet Storm. La botnet Waledac se puede describir como un generador de spam. La arquitectura Waledac es descentralizada similar a la botnet Storm. La botnet Waledac utiliza el protocolo HTTP y P2P para la comunicación, paquetes cifrados e ingeniería social para propagarse. La botnet Waledac comunica a los bot entre sí usando un protocolo P2P a través de HTTP [10].

2.1.1.2.2. Sin estructura Aquí no hay requisitos que se impongan para la organización. Los bots se comunican de manera ad-hoc y en el mundo ideal este tipo de modelo no tendrían ningún mando centralizado, pero en realidad se pueden encontrar en 3 maneras diferentes: puro, híbrido y centralizado. En el tipo puro, cada nodo tiene el mismo grado o rol y sólo hay una capa de enrutamiento sin nodos preferidos. En el tipo centralizado se utiliza un único servidor para etiquetar las funcionalidades y configurar todo el sistema, pero las conexiones entre los pares no son controladas por algún algoritmo. El tipo híbrido es una mezcla entre lo puro y centralizado donde hay algunos nodos llamados súper nodos que tienen un mayor grado de mando. El principal inconveniente de esta implementación es que a veces cuando se trata de encontrar información (que se realiza por la saturación) de algún nodo o la comunicación entre nodos, resulta más complejo, debido a que encontrar un nodo es más difícil que en el tipo con estructura, es difícil si no está bien conectado por la red o si la red está muy congestionada, lo cual puede suceder debido a la saturación [5].

2.2. Protocolos de comunicación de las botnet

Los protocolos de comunicación utilizados por las botnets para el intercambio de información son de una enorme utilidad para los investigadores y administradores de sistemas. La comprensión y el conocimiento de cómo se

comunican pueden responder a muchas preguntas, como dónde buscar y qué tipo de tráfico observar. En primer lugar, porque si se conoce el protocolo utilizado se puede monitorear por dónde se envía normalmente el tráfico y filtrarlo, y, en segundo lugar; se puede definir un patrón normal de un paquete o flujo del protocolo y filtrar lo que parece sospechoso. Además de eso, esta información proporciona la comprensión de las botnets, de dónde viene y que posibles herramientas de software se están utilizando. En esta sección se mencionan algunos protocolos utilizados por las botnets [5].

2.2.1. Protocolo IRC

Es el protocolo más utilizado por los maestros de las botnets para ponerse en contacto con sus bots, debido a que fue el primer protocolo usado por las botnets. El protocolo IRC está diseñado principalmente para las comunicaciones entre grupos grandes pero permite las comunicaciones privadas entre las entidades individuales, lo que da al maestro de las botnets la herramienta adecuada para comunicarse de una manera fácil y flexible [11]. Normalmente, todos los bots se conectarán con el mismo canal de IRC en el servidor C&C designado o en un servidor IRC libre. Aquí el maestro de la bot emitirá una orden y los bots utilizarán su programa para interpretar los comandos y seguir la orden, ya sea para atacar u otras actividades maliciosas. Se podría pensar que al examinar el contenido del tráfico IRC, se podría detectar comandos botnet, pero existen inconvenientes al tratar de examinar el contenido. Primero los canales de IRC permiten a los usuarios tener una contraseña en el canal por lo que los mensajes están cifrados. Segundo, cada bot tiene sintaxis diferente y puesto que hay una gran cantidad de familias y cientos de variaciones, prácticamente imposible probar todas [5].

En la mayoría de las redes corporativas el uso de clientes IRC o servidores se bloquea normalmente, lo que hace que sea más fácil saber si un bot está actuando ya que no hay tráfico IRC y si lo hay significa que alguien ha sido infectado. Hoy en día y para pasar este obstáculo, los bots han sido capaces de crear un túnel bajo HTTP para utilizar el protocolo IRC, lo que hace que sea más difícil para los métodos de detección, el detectarlo. Ya hay algunos IDS (sistema de detección de intrusiones) que pueden detectar este tipo de tráfico [11] [5].

A pesar de que en los lugares corporativos han tomado algunas medidas

para detectar este tipo de actividad, la mayoría de los hogares y las pequeñas empresas no están bien protegidos contra este tipo de actividad lo que hace que este protocolo sea una herramienta muy atractiva, para ser utilizada por los atacantes ya que hay un gran número de mecanismos de software que hacen que sea muy fácil de configurar y usar [5].

2.2.2. Protocolo HTTP

La popularidad del uso de este protocolo se ha relacionado con la atención que se ha prestado a la utilización de IRC. La principal ventaja es fácil de notar, la mayor parte del tráfico en Internet es HTTP por lo que es mucho más difícil encontrar actividades maliciosas ya que el objetivo a examinar es mucho más grande [5].

Su funcionamiento es sencillo, sólo necesita usar la URL para publicar los comandos necesarios. Después de ser infectado, el bot visita una URL con su ID, la cual está en el servidor controlado por el maestro de la botnet, el servidor responde con una nueva dirección URL que será analizada por el bot para interpretar las órdenes de la actividad maliciosa [5].

Normalmente los cortafuegos bloquean el tráfico entrante de IRC, pero no pueden hacer lo mismo para HTTP (o bloquearían todas las conexiones normales HTTP), entonces se tendrían que aplicar los filtros adecuados para las cabeceras anómalas o el payload (si no está cifrado). La práctica de actividades maliciosas que utilizan este protocolo tiende a crecer y tal vez un día podría ser el más utilizado, la ventaja de pasar por los cortafuegos de una manera fácil y el porcentaje de tráfico a analizar, hace que sea una herramienta poderosa. Tal vez el principal inconveniente es la latencia inherente a ponerse en contacto con un gran número de bots y unir fuerzas para lanzar un gran ataque [5].

2.2.3. Protocolo Overnet

Overnet tiene una estructura descentralizada P2P de intercambio de archivos. Posee una tabla de hash distribuida con base en el protocolo Kademlia [36]. Overnet fue implementado por Edonkey [37]. A finales de 2006, fue oficialmente dado de baja [37] como resultado de la acción legal de la

"*Recording Industry Association of America*" (RIAA) y otros, pero los pares Overnet benignos todavía existen en Internet [9].

2.3. Clasificación

Las técnicas de aprendizaje automático se pueden dividir en 2 enfoques: supervisados y no supervisados [14].

2.3.1. Aprendizaje supervisado

El aprendizaje supervisado es también llamado proceso de clasificación [30]. El aprendizaje supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten en pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. La salida de la función es una etiqueta de clase que define a que clase pertenece el nuevo objeto analizado. El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos o datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente [39]. En este tipo de problemas, se estudia un fenómeno representado por un caso vector conocido que puede ser clasificado de maneras diferentes de acuerdo a un vector conocido de etiquetas o clases. El aprendizaje supervisado y los algoritmos de clasificación hacen uso de un conjunto de datos de entrenamiento previamente etiquetados que, en este caso; se corresponde con las etiquetas de tráfico legítimo o procedente de una botnet en función del origen de las representaciones utilizadas para los experimentos realizados en este trabajo de investigación [8]. Con tal fin, se dispone de un conjunto D de entrenamiento D = $\{(X_i, Y_i)\}$ i=1 hasta n, para n casos, donde X_i representa los valores correspondientes al caso; mientras que Y_i es la etiqueta que lo sitúa en la categoría que el clasificador asume como correcta [8]. Un segundo conjunto de datos de prueba sirve para validar la efectividad del aprendizaje. Consiste en una colección de datos para los que el método de aprendizaje ya entrenado debe predecir sus etiquetas basado en sus atributos. Estas predicciones luego son comparadas con las etiquetas de cada objeto de la muestra de prueba para medir el desempeño del método de aprendizaje [40]. Un tercer conjunto de datos de validación se utiliza

para ajustar los parámetros de un método de aprendizaje. Los métodos de aprendizaje usualmente tienen uno o más parámetros y la muestra de validación es utilizada para seleccionar sus valores apropiados [40]. El conjunto de datos de prueba, así como el de validación, mantienen la misma estructura del conjunto de datos de entrenamiento. Dentro del aprendizaje supervisado se encuentran los árboles de decisión. En la figura 2.3 se puede observar un ejemplo de cómo se realiza este proceso de aprendizaje supervisado en general.

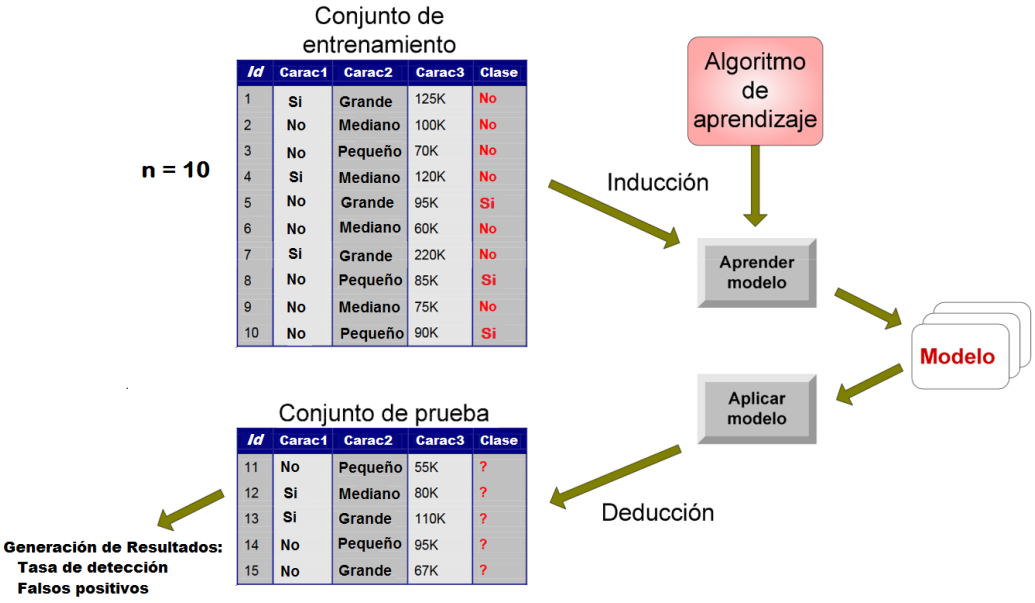

Figura 2.3: Proceso de Aprendizaje Supervisado General.

2.3.1.1. Árboles de decisión

Los árboles de decisión son un tipo de clasificadores de aprendizaje supervisado que pueden ser representados gráficamente como árboles. Los nodos interiores representan las condiciones o estados posibles de las variables del problema y los nodos finales u hojas constituyen la decisión final del clasificador [15]. Un árbol de decisión es, formalmente, un grafo G = (V, E) que consiste en un conjunto no vacío de nodos finitos V y un conjunto de aristas E. Si el conjunto de las aristas E está compuesto a su vez por bituplas ordenadas de nodos de la forma (v; w), entonces se puede decir que el grafo G es

dirigido. Un camino en el grafo G se define como una secuencia de aristas de la forma $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$. Los caminos se pueden expresar por el origen, su final y la distancia recorrida, entendiendo como tal el mínimo número de aristas desde el origen hasta el final. Además, si (v, w) es una arista en el árbol, entonces se considera a v nodo padre de w, mientras que w sería el nodo hijo de v. El único nodo en el árbol sin padres se denomina nodo raíz. Cualquier otro nodo en el árbol, es un nodo interno. Para construir la representación gráfica de un árbol, se contesta a un conjunto de preguntas binarias. Hay varios algoritmos de aprendizaje supervisado, los cuales utilizan para aprender la estructura de un árbol mediante un conjunto de datos etiquetados. Uno de los algoritmos de árboles de decisión es J48, el cual es la implementación de WEKA [62] del algoritmo C4.5 [17] [8].

2.3.1.1.1. Algoritmo C4.5 El algoritmo C4.5 crea árboles de decisión dada una cantidad de información de entrenamiento usando el concepto de entropía de la información [16]. En cada nodo del árbol de decisión, el algoritmo elige un atributo de los datos que más eficazmente dividen el conjunto de muestras en subconjuntos enriquecidos en una clase u otra utilizando como criterio de selección la diferencia de entropía o la función de ganancia normalizada. J48 es una implementación de código abierto para Weka del algoritmo C4.5 [17] que a su vez es una extensión del algoritmo ID3 [15]. El algoritmo construye árboles de decisión utilizando un conjunto de datos etiquetados y mediante el concepto de entropía de la información, el cual mide la incertidumbre asociada con una variable aleatoria. Este concepto, también conocido como entropía de Shannon [18] [19], mide el valor esperado de información o la ganancia de la información (GI) dentro de un conjunto de datos [8].

Para generar un árbol de decisión mediante el algoritmo C4.5, se requiere un conjunto de datos de entrenamiento S = $\{s_1, s_2, ..., s_{n-1}, s_n\}$ previamente etiquetado. Así, los datos de entrenamiento conforman un corpus de ejemplos $s_i = x_1, x_2, ..., x_m$ ya clasificados en los que $x_1, x_2, ..., x_{m-1}, x_m$ representan los m atributos o características del ejemplo. El atributo m de cada representación corresponderá con alguna de las o etiquetas C = $\{c_1, c_2, .., c_{o-1}, c_o\}$ y representarán la clase a la que pertenece dicho espécimen s_i [8].

Para cada nodo del árbol, el algoritmo C4.5 elige el atributo dentro de los

datos que, de forma más eficiente, divide el conjunto restante de especímenes en subconjuntos de una clase o de la otra. El criterio para dividir el conjunto es la ganancia de la información (GI), la cual calcula la dependencia estadística entre dos variables aleatorias. De este modo, el atributo con el mayor GI se selecciona para ser el nodo que divida el nuevo conjunto de datos. Este proceso se repite recursivamente hasta que no se puedan realizar más divisiones del conjunto de datos [8].

El algoritmo C4.5 es capaz de manejar tanto atributos continuos como discretos, datos sin completar, atributos con diferentes costes y, además, puede podar los árboles después de generarlos [8]. El algoritmo general del C4.5 se encuentra en Algoritmo 1.

Algoritmo 1 Algoritmo General C4.5 [41].

Para cada característica C_i (i=1...l) donde l es el total de características.
Hacer
Encontrar la ganancia de información normalizada de la división de C_i.
Fin Para
Dejar que C_i_best sea la característica con la ganancia de información normalizada más alta.
Crear un nodo de decisión que divida C_i_best.
Repetir en las sublistas obtenidas por división de C_i_best y agregar estos nodos como hijos de nodo.

2.4. Algoritmos Genéticos

El contenido de esta sección fue obtenido de los apuntes del Dr. Coello [42].

Los algoritmos genéticos (AG) (denominados originalmente "planes reproductivos genéticos") fueron desarrollados por John H. Holland a principios de los 1960s [20] [21], motivado por resolver problemas de aprendizaje de máquina.

El algoritmo genético enfatiza la importancia de la cruza sexual (operador principal) sobre el de la mutación (operador secundario) y usa selección probabilística.

El AG es conocido por su capacidad para resolver problemas de optimización. El AG Simple se puede ver en algoritmo 2.

Algoritmo 2 AG Simple

Generar (aleatoriamente) una población inicial.
Mientras condición de parada no sea alcanzada. **Hacer**
Calcular aptitud de cada individuo.
Seleccionar (probabilísticamente) con base en su aptitud.
Aplicar operadores genéticos (cruza y mutación) para generar la siguiente población.
Fin Mientras

La representación tradicional es la binaria, tal y como se ejemplifica en la figura 2.4. A la cadena binaria se le llama "cromosoma" ó "individuo". A cada

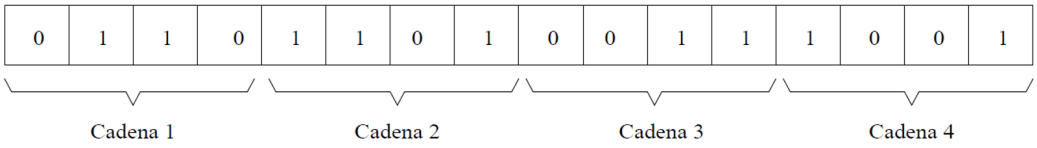

Figura 2.4: Ejemplo de la codificación (mediante cadenas binarias) usada tradicionalmente con los algoritmos genéticos.

posición de la cadena se le denomina "gen" y al valor dentro de esta posición se le llama "alelo".

La figura 2.4 contiene 4 individuos formados por 4 genes cada uno.

Para poder aplicar el algoritmo genético se requiere de los 5 componentes básicos siguientes:

- Una representación de las soluciones potenciales del problema.

- Una forma de crear una población inicial de posibles soluciones (normalmente un proceso aleatorio).
- Una función de evaluación que juegue el papel del ambiente, clasificando las soluciones en términos de su "aptitud".
- Operadores genéticos que alteren la composición de los hijos que se producirán para las siguientes generaciones.
- Valores para los diferentes parámetros que utiliza el algoritmo genético (tamaño de la población, probabilidad de cruza, probabilidad de mutación, número máximo de generaciones, etc.)

2.4.1. Selección

Una parte fundamental del funcionamiento de un algoritmo genético es, sin lugar a dudas, el proceso de selección de candidatos a reproducirse. En el algoritmo genético este proceso de selección suele realizarse de forma probabilística (es decir, aún los individuos menos aptos tienen una cierta oportunidad de sobrevivir). Las técnicas de selección usadas en algoritmos genéticos pueden clasificarse en tres grandes grupos: Selección proporcional, Selección mediante torneo, Selección de estado uniforme. La técnica utilizada en esta investigación es la de torneo por algunas de sus características mencionadas a continuación.

2.4.1.1. Selección por torneo

Esta técnica fue propuesta por Wetzel [23] y estudiada en la tesis doctoral de Brindle [22].

La idea básica del método es seleccionar con base en comparaciones directas de los individuos.

Hay 2 versiones de la selección mediante torneo: Determinística y Probabilística. El tipo de torneo utilizado fue el torneo Determinístico.

El algoritmo del Torneo Determinístico se puede observar en Algoritmo 3.
A continuación, se puede ver un ejemplo de su funcionamiento, donde se tienen 6 aptitudes que corresponden a 6 individuos, con 2 barajeos, cada

Algoritmo 3 Algoritmo Torneo Determinístico.

Barajar los individuos de la población.
Escoger un número p de individuos (típicamente 2).
Compararlos con base en su aptitud.
El ganador del "torneo" es el individuo más apto.
Debe barajarse la población un total de p veces para seleccionar n padres (donde n es el tamaño de la población).

barajeo tiene 2 selecciones, el ganador será el individuo con la aptitud más alta y los ganadores serán los padres de la siguiente generación.

Orden	Aptitud
(1)	254
(2)	47
(3)	457
(4)	194
(5)	85
(6)	310

Barajar 1		
Selección1	Selección2	Ganador
(2)	(6)	(6)
(1)	(3)	(3)
(5)	(4)	(4)

Barajar 2		
Selección1	Selección2	Ganador
(4)	(1)	(1)
(6)	(5)	(6)
(2)	(3)	(3)

Padres elegidos
(6) y (1), (3) y (6), (4) y (3)

Las ventajas de este tipo de selección son:

- La versión determinística garantiza que el mejor individuo será seleccionado p veces.
- Complejidad:
- La técnica eficiente y fácil de implementar.

- Cada competencia requiere la selección aleatoria de un número constante de individuos de la población. Esta comparación puede realizarse en $O(1)$.
- Se requieren "n" competencias de este tipo para completar una generación.
- Por lo tanto, el algoritmo es $O(n)$.

2.4.2. Cruza

En los sistemas biológicos, la cruza es un proceso complejo que ocurre entre parejas de cromosomas. Estos cromosomas se alinean, luego se fraccionan en ciertas partes y posteriormente intercambian fragmentos entre sí. En computación evolutiva se simula la cruza intercambiando segmentos de cadenas lineales de longitud fija (los cromosomas). Aunque las técnicas de cruza básicas suelen aplicarse a la representación binaria, éstas son generalizables a alfabetos de cardinalidad mayor, si bien en algunos casos requieren de ciertas modificaciones. Comenzaremos por revisar las tres técnicas básicas de cruza: Cruza de un punto, Cruza de dos puntos, Cruza uniforme. La cruza utilizada es la cruza uniforme.

2.4.2.1. Cruza uniforme

Esta técnica fue propuesta originalmente por Ackley [24], aunque se le suele atribuir a Syswerda [25].

En este caso, se trata de una cruza de n puntos, pero en la cual el número de puntos de cruza no se fija previamente.

La cruza uniforme tiene un mayor efecto disruptivo que cualquiera de las otras cruzas. A fin de evitar un efecto excesivamente disruptivo, suele usarse con una probabilidad de cruza = 0.5 (PC). Algunos investigadores, sin embargo, sugieren usar valores más pequeños de PC [26].

Cuando se usa PC = 0.5, hay una alta probabilidad de que todo tipo de cadena binaria de longitud L (rango en consideración) sea generada como máscara de copiado de bits.

En la cruza uniforme se genera una máscara de copiado de bits, en la cual se copian los valores de los padres para generar dos hijos descendientes, la primera es generada aleatoriamente, mientras que la segunda es el complemento a dos de la primera.

Un ejemplo de la cruza uniforme se puede ver en la Figura 2.5, donde la máscara de copiado de bits está representada por las líneas de los padres hacia los hijos.

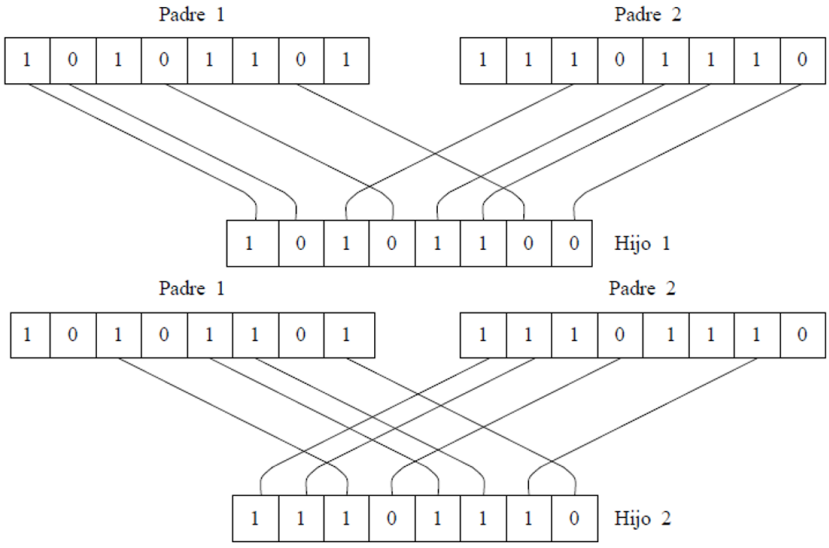

Figura 2.5: Ejemplo de Cruza Uniforme.

2.4.3. Mutación

En la práctica, se suelen recomendar porcentajes de mutación de entre 0.001 y 0.01 para la representación binaria.

Algunos autores sugieren que 1/L como porcentaje de mutación (donde L es la longitud de la cadena cromosómica) es un límite inferior para el porcentaje óptimo de mutación [27].

El papel que juega la mutación en el proceso evolutivo, así como su comparación con la cruza, sigue siendo tema frecuente de investigación y debate en la comunidad de computación evolutiva.

2.5. Resumen del capítulo

En este capítulo se presentaron los conceptos necesarios para la elaboración de la tesis. Se comenzó con los conceptos generales de las botnets, definición, taxonomía y principales características, después se mencionaron los conceptos generales del método de clasificación utilizado y su funcionamiento, finalizando con los conceptos generales sobre algoritmos genéticos y sus características utilizadas.

Capítulo 3

Estado del arte

En este capítulo se presentan trabajos relacionados identificados en el estado del arte. Se mencionan tipos de métodos utilizados para la detección de botnets, algoritmos, donde en algunos de ellos se utiliza una selección arbitraria de características.

Los trabajos en el estado del arte están relacionados con los algoritmos basados en conexiones con intervalos de tiempo para detectar las botnets en la fase de C&C. Sólo dos de ellos realizan una selección de características.

Los algoritmos de detección utilizados en el estado del arte son: *Support Vector Machine (SVM)*, *Artificial Neural Network (ANN)*, *Nearest Neighbors Classifier (NNC)*, *Gaussian-Based Classifier (GBC)*, *Naives Bayes Classifier (NBC)*, árboles de decisión *Rep*, árboles de decisión C4.5, *K-means* y el algoritmo de colonia de hormigas

A continuación, se muestran los trabajos del estado del arte y la evolución de éstos en los últimos años. Estos son presentados desde el más antiguo, hasta el más actual.

3.1. Año 2011

Saad et al. [46] realizaron una comparación entre cinco técnicas de aprendizaje automático usadas para detectar botnets descentralizadas. Los resultados de los experimentos con base en el conjunto de datos ISOT muestran que es posible detectar botnets con precisión durante la fase de comando y control (C&C). Las técnicas de clasificación utilizados para detectar botnets fueron: *Support Vector Machine (SVM)* con un 97.9 % de tasa de detección, *Naive Bayes* con un 89.8 % de tasa de detección, *Gauss classifier* con un 96 % de tasa de detección, *Artificial Neural Network* con un 94.6 % de tasa de detección y *Nearest Neighbor* con un 92.1 % de tasa de detección, donde el algoritmo SVM obtuvo la mejor tasa de detección con casi el 98 %. Las características usadas en ese trabajo se pueden observar en la Tabla 3.1.

Tabla 3.1: Vector de características usado por Saad et al. [46].

Nombre	Descripción
SrcIP	IP de origen
SrcPort	Puerto de origen
DstiP	IP de destino
DstPort	Puerto de destino
Protocol	Protocolo
Pack length	Tamaño del *payload*
APL	Promedio del *payload* por conexión
FPL	Largo del primer paquete
TPC	Número total de paquetes
TBT	Bytes totales transmitidos
lOP	El radio entre el número de paquetes entrantes sobre el número de paquetes salientes
DPL	El número total de paquetes del mismo tamaño sobre el número total de paquetes
PL	El número total de bytes sobre el número total de paquetes

3.2. Año 2012

B. Piyush et al. [47] analizaron el algoritmo Support Vector Machine para detectar botnets. Su investigación hace énfasis en que existe una diferencia significativa entre los valores de las características de las conexiones de las botnets descentralizadas y las del tráfico normal, en este artículo además combinaron tráfico web normal y tráfico P2P normal. Evaluaron el modelo haciendo una selección de características, donde eliminaban una por una y las clasifican dependiendo de la tasa de detección que obtenían al quitarla/agregarla. Su método de selección de características obtiene un 99.88 % de tasa de detección. El conjunto de datos utilizado fue obtenido del Departamento de Ciencias de la Computación de la Universidad de Texas en Dallas, el cual contiene la botnet *Nugache*. El tráfico normal fue obtenido de forma aleatoria de computadoras usando Windows incluyendo HTTP, FTP, SMTP. Además, su conjunto de datos contiene tráfico P2P proveniente de *Bit Torrent*, *Skype* y *e-Donkey*. Las características usadas en este trabajo se pueden observar en la Tabla 3.2.

Tabla 3.2: Vector de características usado por D. Zhao et al. [47].

Nombre	Descripción
lrgst_pck	Tamaño del paquete más grande
bytes_lrgst_pkt	Bytes totales transmitidos con los paquetes más largos
total_bytes	Bytes totales transmitidos
ratio_of_lrgst_pkt	Radio de los paquetes más largos
avg_iat	El promedio del tiempo de llegada de los paquetes
var_iat	Varianza del tiempo de llegada de los paquetes
avg_pktl	El promedio del tamaño de los paquetes
var_pktl	Varianza del tamaño de los paquetes

3.3. Año 2013

D. Zhao et al. [48] utilizaron el algoritmo *árboles REP* para detectar botnets utilizando el conjunto de datos ISOT. Los resultados muestran una tasa de detección del 98.1 % para un conjunto reducido de datos y una tasa de

detección del 98.3 % para un conjunto completo con ventanas de tiempo de 300 segundos, con 8.58 segundos de entrenamiento y 29.4 de entrenamiento respectivamente. Analizaron la tasa de detección y falsos positivos de las botnets con varias ventanas de tiempo, donde la mejor ventana de tiempo fue de 300 segundos. Además, se construyó un servidor para detectar botnets en tiempo real y se hicieron pruebas con 2 botnets centralizadas: *Black Energy* y *Weasel*, las cuales otorgaron 100 % de tasa de detección. Las características usadas en este trabajo se pueden observar en la Tabla 3.3.

Tabla 3.3: Vector de características usado por D. Zhao et al. [48].

Nombre	Descripción
SrcIP	IP de origen
SrcPort	Puerto de origen
DstiP	IP de destino
DstPort	Puerto de destino
Protocol	Protocolo
APL	Promedio del *payload* por conexión
PV	Varianza del *payload*
PX	Número de paquetes intercambiados en un intervalo de tiempo
PPS	Número de paquetes intercambiados por segundo en un intervalo de tiempo
FPS	El tamaño del primer paquete
TBP	El promedio del tiempo de llegada de los paquetes
NR	Número de reconexiones
FPH	El número total de paquetes sobre el número total de paquetes por hora

3.4. Año 2014

E. Beigi et al. [49] realizaron una selección de características para la detección de botnets. Utilizaron el algoritmo de aprendizaje automático C4.5. También utilizaron un algoritmo *greedy* llamado por *stepwise* para la selección de características. Los conjuntos de datos utilizados se obtuvieron de tres organismos diferentes: ISOT, ISCX y *Malware Capture Facility Project*.

En ese trabajo se recopilaron las características de trabajos relacionados con la detección de botnets para la selección de características. Se realizaron diferentes experimentos en los que se separaron características en cuatro grupos con base en su tipo: características con base en bytes, paquetes, tiempo y comportamiento. Los resultados del conjunto final de características muestran una tasa de detección del 99 % en un conjunto de datos con un número limitado de botnets. En otro experimento que contenía algunas botnets para la fase de entrenamiento y un conjunto mucho más diverso de botnets para la fase de prueba, la tasa de detección fue del 75 %. Las características usadas en este trabajo se pueden observar en la Tabla 3.4.

Tabla 3.4: Vector de características usado por E. Beigi et al. [49].

Nombre	Descripción
SrcIP	IP de origen
SrcPort	Puerto de origen
DstiP	IP de destino
DstPort	Puerto de destino
Protocol	Protocolo
PX	Número de paquetes intercambiados
NNP	No. de paquetes nulos intercambiados (tamaño 0 *payload*)
NSP	No. de paquetes pequeños intercambiados (tamaño 63-400 bytes)
PSP	Porcentaje de paquetes pequeños intercambiados
lOPR	El radio entre el número de paquetes entrantes sobre el número de paquetes salientes
Reconexión	Número de reconexiones
Duración	Duración de la conexión
FPS	Largo del primer paquete
TBT	Número total de bytes
APL	Promedio del *payload* por conexión
DPL	El número total de paquetes del mismo tamaño sobre el número total de paquetes
PV	Varianza del *payload*
BS	Promedio de bits por segundo

PS	Promedio de paquetes por segundo en una ventana de tiempo
AIT	El promedio del tiempo de llegada de los paquetes
PPS	Promedio de paquetes por segundo

K. Huseynov et al. [50] realizaron una comparación entre el algoritmo de *K-means* y el algoritmo de colonia de hormigas para detectar botnets descentralizadas. Utilizaron características con base en el host y propusieron un método capaz de detectar los botnets de manera rápida y precisa. Sus resultados muestran que el algoritmo de *K-means* tiene una mejor tasa de detección y falsos positivos más bajos que el algoritmo de colonia de hormigas. El algoritmo *K-means* obtuvo una tasa de detección 82.1 % y falsos positivos del 2.4 %. Mientras que el algoritmo de colonia de hormigas obtuvo una tasa de detección muy baja, con una tasa de detección de 67.8 % y falsos positivos de 23.5 %. El conjunto de datos utilizado fue ISOT. Las características usadas en este trabajo fueron obtenidas del trabajo de D. Zhao et al. [48] y se pueden observar en la Tabla 3.3.

P. Narang et al. [51] en lugar de utilizar una 5-tupla para la detección de botnets, utilizaron una 2-tupla con base en conversaciones, la cual no necesita de los puertos y protocolos. Llamaron a su detector *PeerShark*, el cual clasifica las diferentes aplicaciones P2P como *Emule* y *Utorrent*, por otra parte, también detectan el tráfico P2P de las botnets descentralizadas Storm y Waledac con una tasa de detección de más de un 95 %. Se ejecutaron pruebas con 3 diferentes algoritmos: *Red Bayesiana*, *C4.5* y *Adaboost con árboles REP* para detectar las botnets descentralizadas, donde el mejor algoritmo fue el C4.5. El conjunto de datos utilizado fue ISOT. Las características usadas en este trabajo se pueden observar en la Tabla 3.5.

Tabla 3.5: Vector de características usado por P. Narang et al. [51].

Nombre	Descripción
Duración	Duración
TPC	Número total de paquetes intercambiados
TBT	Bytes totales transmitidos
MIP	La media del tiempo de llegada de los paquetes

3.5. Año 2015

Ritu et al. [52] realizaron una comparación entre el algoritmo de *árboles REP* y el algoritmo de *Redes Bayesianas* para detectar botnets descentralizadas. Utilizaron características que fueron extraídas de varios trabajos del estado del arte. Su principal aportación fueron 2 características que ellos propusieron, las cuales están basadas en el *handshaking*. Sus resultados muestran una mejora sustancial en la tasa de detección cuando esas características son utilizadas, mejorando en casi 2 % la tasa de detección. Además, de los 2 tipos de algoritmos comparados el que mayor tasa de detección y menores falsos positivos obtuvo fue el algoritmo de *árboles REP*. La tasa de detección obtenida utilizando el conjunto de características propuesto y el algoritmo de *árboles REP* fue de 99.9 %. El conjunto de datos utilizado fue ISOT. Las características usadas en este trabajo se pueden observar en la Tabla 3.6, donde las ultimas 2 características fueron las propuestas en este trabajo.

Tabla 3.6: Vector de características usado por Ritu et al. [52].

Nombre	Descripción
SrcPort	Puerto de origen
DstPort	Puerto de destino
Protocol	Protocolo
APL	Promedio del *payload* por conexión
FPL	Largo del primer paquete
TPC	Número de paquetes intercambiados
TBT	Número total de bytes
OUT	Número de paquetes salientes
IN	Número de paquetes entrantes
DPL	El número total de paquetes del mismo tamaño sobre el número total de paquetes
PV	Varianza del *payload*
Dur	Duración
MIAT	El promedio del tiempo de llegada de los paquetes
MaxPkt	Tamaño del paquete más grande
MinPkt	Tamaño del paquete más pequeño

PacksLen (Propuesto)	Largo del segundo, tercer, cuarto y quinto paquete
IAT (Propuesto)	Tiempo de llegada de paquetes consecutivos desde el primer al quinto paquete

3.6. Discusión del estado del arte

Primero que nada, es importante mencionar que el trabajo del 2015 no fue considerado para la propuesta en la tesis, debido a que fue obtenido después de realizarse. En el estado del arte sólo dos trabajos realizan una selección de características. El primero usando el algoritmo *greedy stepwise*. "Un algoritmo *greedy* es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima" [53] con la esperanza de encontrar un óptimo global. Una estrategia *greedy* en general, no producirá una solución óptima. Por otra parte, en ese trabajo separan características en grupos y se obtienen las mejores características de cada grupo, por lo que no se realiza una evaluación de cada característica. El segundo trabajo que realiza selección de características elimina característica por característica y las clasifica dependiendo de la tasa de detección que obtenían al quitarla/agregarla, esto no evaluaba todo el espacio de soluciones en la selección de características y por lo tanto no otorgara la mejor solución. En el resto de los trabajos relacionados propusieron sus propias características con base en su experiencia sin necesidad de utilizar un método que las evalúe. Este trabajo pretende evaluar y seleccionar el conjunto de características propuestas en los trabajos del estado del arte considerando todas las posibles combinaciones utilizando un método automatizado para obtener una tasa de detección mayor en las botnets.

3.7. Resumen del capítulo

En este capítulo se presentaron los trabajos relacionados en el estado del arte. Se mencionaron los tipos de métodos utilizados para la detección de botnets, algoritmos, donde en algunos de ellos se utiliza una selección arbitraria de características. Se presentó la evolución del estado del arte or-

denada por año y las principales características utilizadas en cada trabajo, detallando la tasa de detección obtenida, los conjuntos de datos utilizados, las botnets utilizadas y su tipo, así como la principal aportación de cada uno de ellos. Además, al final del capítulo se presentó una discusión de los trabajos presentados, donde se menciona sus carencias que la propuesta es capaz de solucionar.

Capítulo 4

Propuesta de selección de características

En este capítulo se presenta la propuesta de selección de características en la tesis, los algoritmos utilizados y cómo interactúan para obtener el conjunto de características con la mayor tasa de detección de las botnets.

La propuesta es capaz de evaluar las características para determinar el conjunto con la tasa de detección más alta, este método mejora la tasa de detección que los trabajos del estado del arte.

El vector inicial de características es recopilado de los trabajos del estado del arte relacionados con la detección de botnets en la fase de C&C por medio de conexiones con intervalos de tiempo. En general, esta propuesta utiliza un algoritmo genético (AG) para seleccionar el conjunto de características que mejor detecta las botnets. El algoritmo C4.5 es responsable de evaluar conjuntos de características potenciales generadas por el AG y calcular la tasa de detección obtenida. Como resultado de la interacción del AG y el algoritmo C4.5, se obtuvo el mejor conjunto de características para la detección de botnets en su etapa de C&C. El AG fue elegido debido a su capacidad para resolver problemas de optimización como lo es la selección de características con la mejor tasa de detección y el conocimiento que se tiene de él, además es de propósito general y no necesita conocimiento de la forma del espacio de búsqueda. Mientras que el algoritmo C4.5 fue elegido influenciado por el estado del arte, ya que presenta la mejor tasa de detección.
La Figura 4.1 muestra el proceso general de la propuesta de selección de ca-

racterísticas para obtener el conjunto de características.

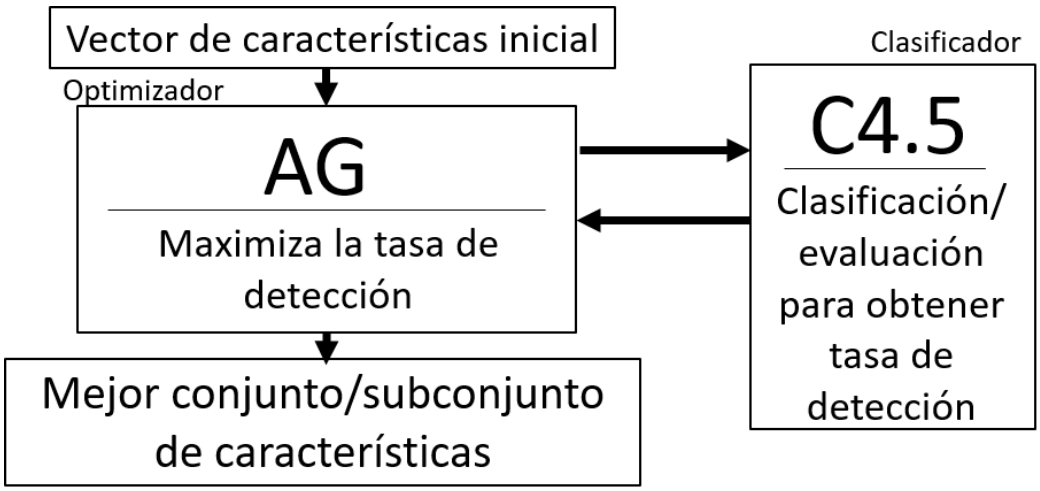

Figura 4.1: Proceso realizado de la propuesta de de selección de características.

4.0.1. Vector inicial de características

El conjunto inicial de 19 características utilizado en este trabajo fue tomado de los trabajos del estado del arte relativos a la detección de botnets en la etapa de C&C utilizando características sin intervalos de tiempo. Este vector de características se muestra en la tabla 4.1 donde, en la primera columna se muestra una enumeración de las características; en la segunda columna se refiere al nombre dado a la característica; la tercera columna es la descripción de la característica; y la cuarta columna muestra las referencias de donde se extrajo. Este conjunto inicial tiene en total 19 características.

Tabla 4.1: Vector inicial de características.

No.	Nombre	Descripción	Referencia
1	BytesAB	Bytes de origen a destino	[38]
2	BytesBA	Bytes de destino a origen	[38]

3	NPackets	No. Paquetes transmitidos	[46], [48], [51], [38]
4	NPacketsAB	No. Paquetes de origen a destino	[38]
5	NPacketsBA	No. Paquetes de destino a origen	[38]
6	Duración	Duración de la conexión	[49], [51], [38]
7	APL	Promedio del *payload* por conexión	[49], [50], [46], [48]
8	DPS	Cantidad de diferentes tamaños de paquete por conexión	[49]
9	Payload	Cantidad total de bytes por conexión excluyendo la cabecera	[46]
10	TBT	Bytes totales transmitidos	[49], [46], [51], [38]
11	Flen	Largo del primer paquete de la conexión	[49], [50], [46], [48]
12	NNP	No. de paquetes nulos intercambiados (tamaño 0 *payload*)	[49]
13	NSP	No. de paquetes pequeños intercambiados (tamaño 63-400 bytes)	[49]
14	PSP	Porcentaje de paquetes pequeños intercambiados	[49]
15	IPP	Paquetes de entrada sobre paquetes	[49], [46]
16	OPP	Paquetes de salida sobre paquetes	[49], [46]
17	PV	Desviación estándar de payload	[49], [50], [48]
18	BS	Promedio de bits por segundo	[49]
19	PPS	Promedio de paquetes por segundo	[49]

4.0.2. Algoritmo Genético (AG)

El AG busca el conjunto de características que mejor detectan las botnets en la etapa de C&C al tratar con diferentes conjuntos de características.

La representación de los individuos es mediante vectores binarios. Cada entrada representa una característica, donde un valor de 1 significa que la ca-

característica está incluida y un valor de 0 significa que la característica no está incluida en el conjunto solución/conjunto de características. La longitud del vector es 19.

La representación de los individuos se muestra en la figura 4.2. Donde la representación del individuo está en la parte superior de la figura y en la parte inferior está el vector inicial de características que representan el individuo. En este ejemplo, la primera característica BytesAB no está incluida, pero la segunda característica BytesBA si está incluida y así sucesivamente.

Figura 4.2: Representación de los individuos en el AG.

El espacio de solución en consideración para el AG es de 2^{19} posibles individuos (vectores solución). El tamaño del espacio de solución sugirió una idea de utilizar un AG.

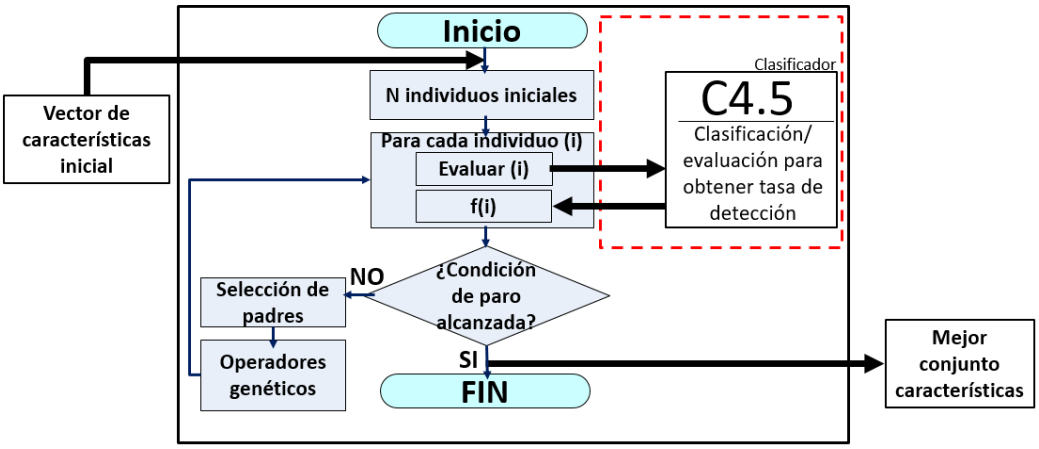

Figura 4.3: Diagrama de flujo del AG.

El AG utiliza un conjunto de 50 posibles soluciones en cada generación, es decir un conjunto de 50 cadenas binarias, donde cada cadena es de longitud 19. El conjunto de 50 cadenas se llama población y cada solución potencial (cadena binaria) se llama individuo. La población inicial es definida al azar. La evaluación de cada individuo es ejecutada por el algoritmo de clasificación C4.5. Este algoritmo calcula la tasa de detección que es la evaluación de cada

individuo. El valor de la aptitud de los individuos es, precisamente, la tasa de detección obtenida por el algoritmo C4.5. El proceso del AG considera 50 generaciones, las cuales son guiadas a través de los operadores genéticos a la convergencia del mejor conjunto de características.
La Figura 4.3 ilustra el proceso realizado por el AG.

4.0.3. Preprocesamiento de la información

Primero que nada, se debe hacer un preprocesamiento de la información con el fin de extraer los conjuntos de datos que servirán para la fase de entrenamiento y validación del clasificador. Para generar el **conjunto de entrenamiento** y el **conjunto de validación** es necesario extraer los datos de los flujos de comunicación pertenecientes a las botnets y al tráfico normal. A partir del flujo de comunicación son extraídas las 19 características del vector inicial con base en conexiones en la red, las cuales se utilizaron para definir el comportamiento de las botnets. Las conexiones organizan los paquetes del flujo de comunicación en un 5-tupla de la siguiente manera: < dirección IP de origen, dirección IP de destino, puerto de origen, puerto de destino, protocolo >. Una vez extraídas las características, el conjunto de datos generado es divido en 2 conjuntos llamados **conjunto de entrenamiento** y **conjunto de validación** los cuales son utilizados en el proceso de clasificación.
La Figura 4.5 ilustra el preprocesamiento de la información.

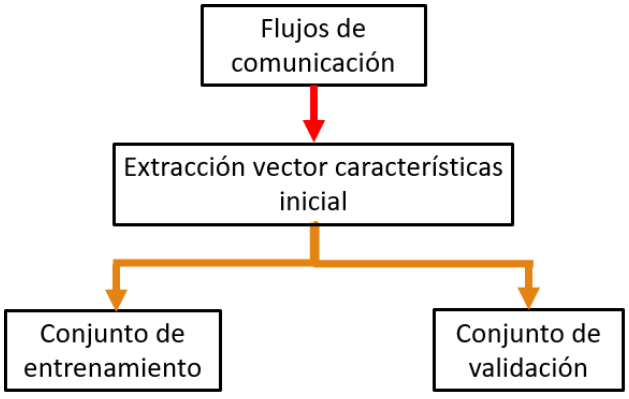

Figura 4.4: Preprocesamiento de la información.

4.0.4. Clasificador C4.5

El proceso de clasificación es realizado por el algoritmo C4.5, el cual evalúa los individuos que son posibles soluciones generadas por el AG.

El **conjunto de entrenamiento** y el un **conjunto de validación** reciben como entrada al individuo que indica un conjunto de características generadas por el AG, el cual es aplicado a ambos conjuntos. El algoritmo de clasificación C4.5, tiene como entrada, un **conjunto de entrenamiento**, el cual contiene las conexiones que pertenecen a una botnet y las conexiones normales, entonces el algoritmo de clasificación C4.5 es responsable de ejecutar la etapa de aprendizaje para generar un modelo con base en árboles de decisión. Este modelo obtiene en cada ciclo la característica que tiene una mayor ganancia de información (menor entropía). La característica $C_x = 1$ que tiene la mayor ganancia forma un nodo que divide a otras características ($C_y = 1, C_z = 1$, ... donde $x \neq y$, $x \neq z$, x, y, $z \in [1, ..,19]$), generando tantas ramas como características existan. Posteriormente este modelo recibe como entrada en la etapa de deducción un **conjunto de validación** que contiene las mismas características que el conjunto de entrenamiento, la etapa de deducción en el conjunto de validación generará como resultado la tasa de detección $f(i)$ que es empleada para medir el individuo i evaluado.
La Figura 4.5 ilustra el proceso de clasificación realizado para evaluar un individuo.

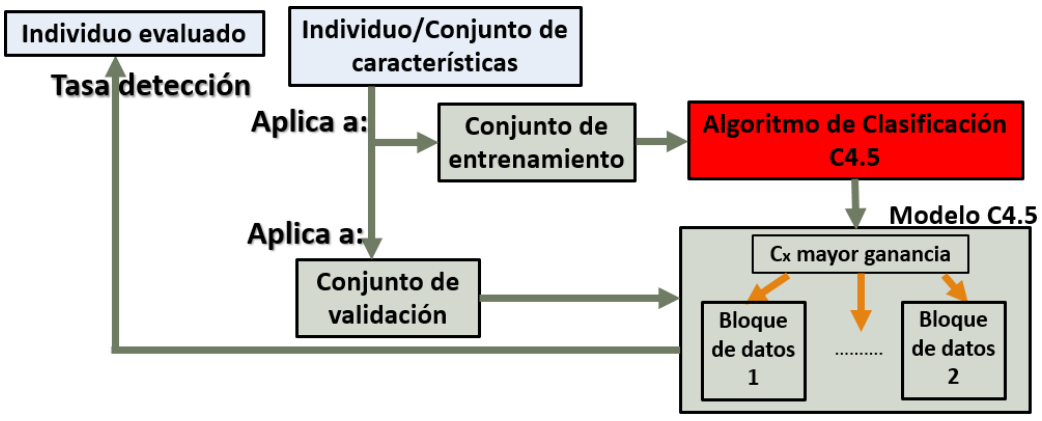

Figura 4.5: Proceso de clasificación para evaluar un individuo.

4.0.5. Los mejores parámetros iniciales en los algoritmos C4.5 y AG

Se realizaron pruebas para obtener los mejores parámetros en el AG y el algoritmo C4.5.
En el algoritmo C4.5 se utilizaron dos parámetros, el factor de confianza, con un valor en el intervalo [0, 1] y mínimo número de objetos en el cual sus valores comienzan en 1. En el factor de confianza se realizaron incrementos de 0.1 y en el mínimo número de objetos incrementos de 1. Los mejores parámetros fueron obtenidos por las pruebas de cada una de las combinaciones, por tanto, se realizaron 100 experimentos (10 valores de mínimo número de objetos por 10 valores en factor de confianza), utilizando el conjunto de entrenamiento y prueba ISOT manteniendo las 19 características del vector.

Los resultados de las pruebas para obtener los mejores parámetros para el clasificador C4.5 son mostrados en gris en la Tabla 4.2, los cuales son factor de confidencia igual a 2 y mínimo número de objetos igual a 1.

Tabla 4.2: Prueba de parámetros del algoritmo C4.5.

Factor de confidencia	Mínimo número de objetos
0.1	1
0.2	2
0.3	3
0.4	4
0.5	5
0.6	6
0.7	7
0.8	8
0.9	9
1	10

En el AG se utilizaron dos parámetros, la probabilidad de cruza (PC) en el intervalo [0, 1] y la probabilidad de mutación (PM) en el intervalo [0, 1]. En la probabilidad de cruza se realizaron incrementos de 0.1. Por otro lado, para la probabilidad de mutación mantuvieron valores bajos con sólo

dos valores de 1/L y 2/L, donde L = 19 y representa el número de características en el vector de características. Los mejores parámetros se obtuvieron mediante pruebas de cada una de las combinaciones, por tanto, se realizaron 10 experimentos (5 valores de probabilidad de cruza por 2 valores de probabilidad de mutación), utilizando el conjunto de entrenamiento y prueba de ISOT, manteniendo los mejores parámetros obtenidos del clasificador, con su factor de confianza igual a 0.2 y el mínimo número de objetos igual a 1.

Los resultados de las pruebas para obtener los mejores parámetros para el AG son mostrados en gris en la Tabla 4.3, los cuales son probabilidad de cruza = 0.5 y probabilidad de mutación = 2/L.

Tabla 4.3: Prueba de parámetros del AG.

Probabilidad de cruza	Probabilidad de mutación
0	1/L
0.25	2/L
0.5	
0.75	
1	

4.1. Resumen del capítulo

En este capítulo se presentó la propuesta se selección de características en la tesis, se detallaron cada uno de los componentes y el cómo interactúan para otorgar el mejor conjunto de características con la mejor tasa de detección. Los componentes utilizados en la propuesta fueron: el vector inicial de características que fue extraído de los trabajos representativos del estado del arte, el Algoritmo Genético (AG), su representación, su funcionamiento y la interacción con el algoritmo C4.5 encargado de evaluar las soluciones creadas por el AG, el algoritmo C4.5 y como utiliza los conjuntos de datos y las soluciones generadas por el AG para otorgar la tasa de detección que es recibida por el AG. Además, se realizaron pruebas en la propuesta para obtener los parámetros iniciales para los algoritmos. Se detalló el cómo se extrajo la información de los conjuntos de datos y la división de los mismos. Se

utilizaron figuras acordes a los componentes utilizados, con el fin de facilitar el entendimiento de ellos en cada una de las etapas.

Capítulo 5

Experimentos y resultados

En este capítulo se presenta el diseño de experimentos, las herramientas utilizadas para el desarrollo de éstos, los datos utilizados, los resultados obtenidos, la comparación y un análisis de los mismos.

5.1. Experimentos

Se realizaron experimentos específicos y experimentos generales utilizando dos conjuntos de datos que contienen botnets descentralizadas y centralizadas, con el fin de obtener el mejor conjunto de características con la mayor tasa de detección. Los experimentos llevados a cabo fueron:

Específicos. Que tienen como objetivo encontrar el conjunto de características de una botnet específica, estos experimentos se clasifican como:

· Storm (botnet tipo descentralizada).
· Waledac (botnet tipo descentralizada).
· Neris (botnet tipo centralizada).
· RBot (botnet tipo centralizada).

Generales. Que tienen como objetivo encontrar el conjunto de características de un tipo de botnet y botnets en general, estos experimentos se clasifican como:

· Experimento para botnet descentralizada.

· Experimento para botnet centralizada.
· Experimento para botnet centralizada descentralizada.

Los conjuntos de datos utilizados fueron los siguientes:

· ISOT. El cual contiene botnets descentralizadas.
· ISCX. El cual contiene botnets centralizadas.

En las siguientes subsecciones, en primer lugar, se menciona una descripción acerca de los conjuntos de datos utilizados para los experimentos, después, se mencionan las herramientas y finalmente, se menciona una descripción de los experimentos específicos y generales llevados a cabo.

5.1.1. Conjuntos de datos para experimentos

A continuación se menciona una descripción de los dos conjuntos de datos utilizados para la experimentación. Estos conjuntos de datos contienen datos representativos de las botnets centralizadas y descentralizadas, así como el tráfico normal, centrándose sólo en las conexiones de este tráfico. A partir de estos datos se extrae el conjunto de entrenamiento, así como el conjunto de validación. Estos dos conjuntos de datos son los siguientes:

ISOT Este conjunto de datos fue creado por *Information Security and Object Technology* (ISOT) en la Universidad de Victoria [54]. Básicamente, es una mezcla de muchos conjuntos de datos existentes (maliciosos y no maliciosos). El tráfico malicioso en el conjunto de datos ISOT se extrajo de *French chapter of the Honeynet Project* [55] e incluye tres diferentes botnets descentralizadas: Waledac, Storm y Zeus.

El tráfico no malicioso se extrajo de dos organismos: Uno fue extraído del Laboratorio de Investigación de Ericsson en Hungría [56]. El segundo fue extraído de *Lawrence Berkeley National Lab* (LBNL) [57].

Esta combinación de tráfico normal es importante debido a que el conjunto de datos del Laboratorio de Ericsson contiene tráfico de varias aplicaciones tales como motores de búsqueda con tráfico HTTP, juegos como World of Warcraft y el tráfico de clientes de bittorrent como Azureus. Por otro lado, el tráfico LNBL proviene de una red de negocios de tamaño medio que consta

de 5 conjuntos de datos.

En total, el conjunto de datos de ISOT contiene 14.1 GB de datos en formato pcap. La descripción de este conjunto de datos se muestran en la Tabla 5.1, en él que la primera columna representa la botnet; la segunda el número de conexiones; finalmente, la tercera el tipo de botnet.

Tabla 5.1: Descripción del conjunto de datos ISOT.

Botnet	Número de conexiones	Tipo
ISOT Storm	22,888	Descentralizada
ISOT Waledac	34,442	Descentralizada
ISOT NO Botnet	77,586	NO Botnet

ISCX Este conjunto de datos fue creado por el *Information Security Centre of Excellence* (ISCX) de la Universidad de New Brunswick [58]. Se ha generado en un entorno de prueba físico utilizando dispositivos reales que generan tráfico (SSH, HTTP y SMTP). Contiene las botnets centralizadas Neris y RBot. En total, el conjunto de datos ISCX contiene 5.6GB de datos en formato pcap. La descripción de este conjunto de datos se muestran en la Tabla 5.2, en él que la primera columna representa la botnet; la segunda el número de conexiones; finalmente, la tercera el tipo de botnet.

Tabla 5.2: Descripción del conjunto de datos ISCX.

Botnet	Número de conexiones	Tipo
ISCX Neris	33,084	Centralizada
ISCX RBot	34,217	Centralizada
ISCX NO Botnet	76,175	NO Botnet

5.1.2. Herramientas usadas en los experimentos

Se utilizaron una serie de herramientas y programas para el desarrollo de los experimentos. Estas herramientas y programas juegan un papel importante para llevar a cabo la tarea de detección de botnets. Por lo tanto, es importante dar una breve introducción a estas herramientas y programas, discutiendo sus ventajas y limitaciones.

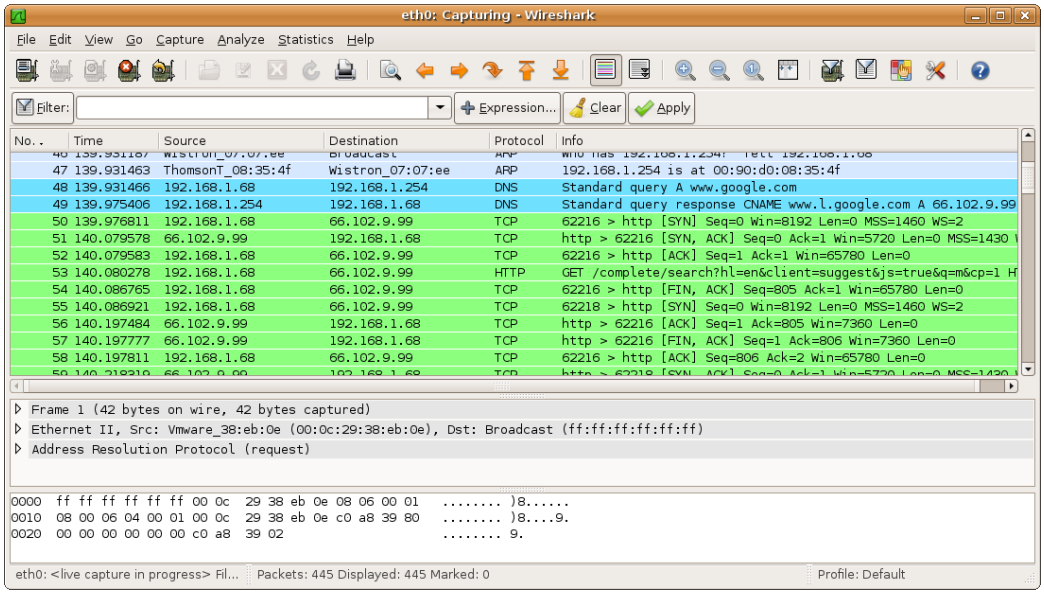

Figura 5.1: Ejemplo de captura de tráfico de la herramienta Wireshark.

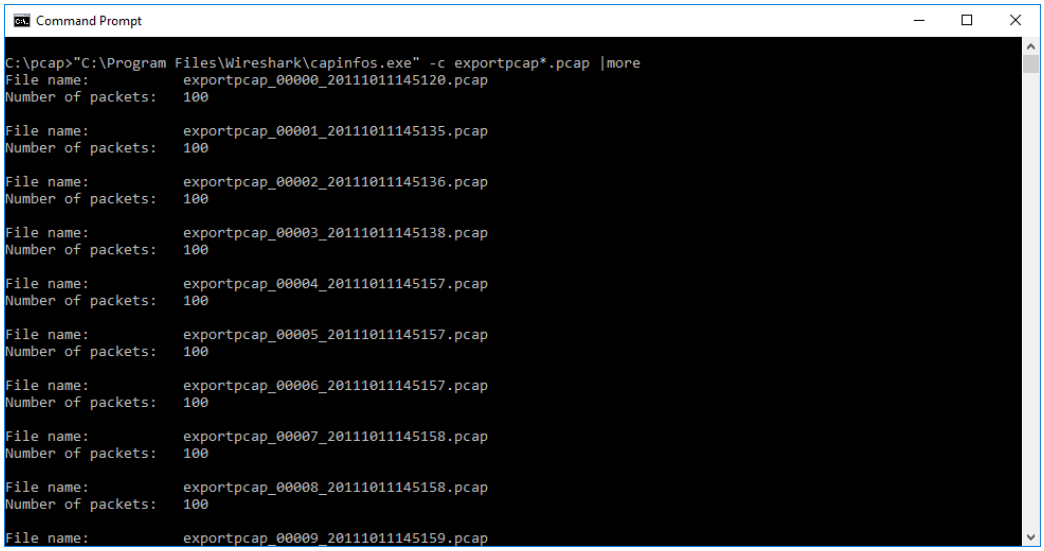

Figura 5.2: Ejemplo de la división de un conjunto .pcap con la herramienta Tshark.

Wireshark [60] fue utilizado para leer los conjuntos de datos en formato

.pcap y con él se obtuvieron las conexiones a las que posteriormente se le extrajeron las características. Un ejemplo de captura de tráfico de Wireshark en la red se muestra en la Figura 5.1.

Tshark fue utilizado para separar el tráfico en los conjuntos de datos en formato .pcap de las distintas botnets que se encuentran en cada conjunto de datos de ambos organismos ISOT e ISCX. Un ejemplo de la división de un conjunto .pcap con la herramienta Tshark se muestra en la Figura 5.2.

```
@relation ISOT
@attribute BytesAB NUMERIC
@attribute BytesBA NUMERIC
@attribute NPackets NUMERIC
@attribute NPacketsAB NUMERIC
@attribute NPacketsBA NUMERIC
@attribute Duracion NUMERIC
@attribute MPL NUMERIC
@attribute MDPS NUMERIC
@attribute Payload NUMERIC
@attribute TPF NUMERIC
@attribute Flen NUMERIC
@attribute NNP NUMERIC
@attribute NSP NUMERIC
@attribute PSP NUMERIC
@attribute IPR NUMERIC
@attribute OPR NUMERIC
@attribute PV NUMERIC
@attribute BS NUMERIC
@attribute PPS NUMERIC
@attribute 'class' { Anormal, Normal}
@data
73.0,139.0,2.0,1.0,1.0,42.077,106.0,1.0,144.0,212.0,73.0,0.0,2.0,1.0,0.0,0.0,23.334524,40307.056111,47.531906, Anormal
86.0,118.0,2.0,1.0,1.0,0.42,102.0,1.0,136.0,204.0,86.0,0.0,2.0,1.0,0.0,0.0,11.313708,3885714.286404,4761.904763, Anormal
68.0,201.0,2.0,1.0,1.0,698.745,134.5,1.0,201.0,269.0,68.0,0.0,2.0,1.0,0.0,0.0,47.022601,3079.807369,2.862275, Anormal
5361.0,2391.0,119.0,82.0,37.0,1426945.516,65.142857,0.042017,3402.0,7752.0,67.0,0.0,119.0,1.0,0.0,0.0,2.052052,43.460664,0.083395, Anormal
76.0,102.0,2.0,1.0,1.0,717.844,89.0,1.0,110.0,178.0,76.0,0.0,2.0,1.0,0.0,0.0,9.192388,1983.717911,2.786121, Anormal
636,416,9,5,4,528.306,116.888889,0.444444,520,1052,62,7,2,0.222222,0,0,38.160731,15930.161687,17.035582, Normal
213,1484,2,1,1,52.769,848.5,1,1589,1697,213,0,1,0.5,0,0,449.366359,257272.262117,37.90104, Normal
842,12025,21,9,12,146.596,612.714286,0.285714,11663,12867,62,11,1,0.047619,0,0,144.418114,702174.684166,143.250839, Normal
1583,2256,21,11,10,14.37,182.809524,0.714286,2669,3839,62,5,15,0.714286,0,0,50.365823,2137230.340989,1461.377871, Normal
2194,19348,61,25,36,23636.311,353.147541,0.114754,18090,21542,62,26,3,0.04918,0,0,33.140977,7291.154698,2.580775, Normal
```

Figura 5.3: Ejemplo del formato arff de Weka.

Weka The Waikato Environment for Knowledge Analysis (Weka) es un conjunto de herramientas bien conocidas de código abierto de aprendizaje automático para tareas de minería de datos [63]. Está escrito en el lenguaje de programación Java y se puede ejecutar en casi todas las plataformas. El formato que se utiliza en los pasos de pre-procesamiento y clasificación que utiliza Weka, es el formato de atributo-relación de archivos (ARFF), un archivo de texto ASCII, que describe una lista de instancias que comparten una serie de características. El formato ARFF contiene encabezados que describen los atributos. Esto significa que las estructuras de datos internas pueden configurarse correctamente antes de leer los datos. Un ejemplo del formato ARFF se representa en la Figura 5.3., el formato ARFF se divide en dos

partes, @relation y @data como se muestra en la Figura 5.3. La parte @relation contiene una lista de nombres y tipos de los atributos y la parte @data contiene todas las instancias de atributos declarados (datos correspondientes a los atributos). De acuerdo con [63], los diferentes formatos soportados son numéricos, nominales (discreto o un conjunto de valores predefinidos), cadenas y fechas [6].

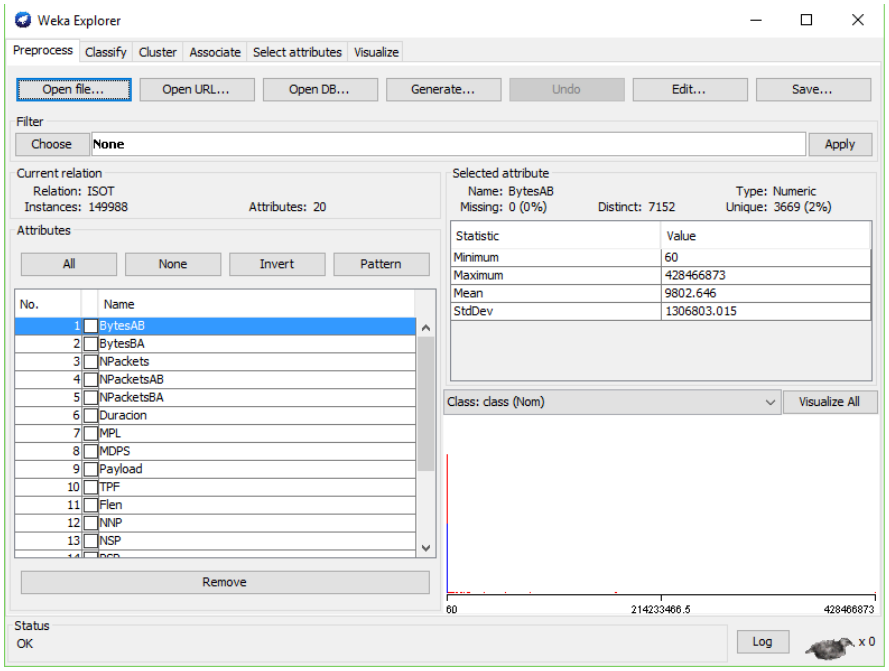

Figura 5.4: Ejemplo de la herramienta Weka explorer.

Weka contiene una serie de interfaces de usuario diferentes para el procesamiento de los datos utilizados para entrenar el sistema [6]. Un ejemplo del Explorador de Weka que contiene los datos de entrenamiento se muestra en la Figura 5.4. Además, existe una serie de bibliotecas que son utilizadas por Weka explorer, las cuales pueden ser utilizadas para la programación en el lenguaje de programación Java, importando estas mismas bibliotecas y utilizando los métodos obtenidos.

5.1.3. Parámetros iniciales usados en los experimentos

El AG fue programado en el lenguaje de programación Java. Los parámetros que se aplicaron en el AG son los siguientes:

· Generaciones: 50
· Individuos: 50
· PM: 2/19
· PC: 0.15
· Selección: Torneo
· Cruza: Uniforme

El algoritmo de clasificación C4.5 fue implementado en weka [61].

Los parámetros que se aplicaron en el algoritmo C4.5 son los siguientes:

· Factor de confianza: 0.2
· Mínimo número de objetos: 1

Para dividir el conjunto de datos en cada experimento en un conjunto de entrenamiento y prueba se utilizó una validación cruzada Weka 3-*folds*.

Los experimentos fueron realizados en una computadora portátil Sony Vaio modelo SVS15125PLB con las siguientes especificaciones:

· 8GB de memoria ram.
· Procesador Intel Core I7-3520M a 2.90GHz con Turbo Boost hasta 3.60GHz.
· Disco duro de 750gb a 7200rpm.
· Sistema operativo Windows 10 con los procesos básicos del sistema.

Cada evaluación de individuo generado por el AG fue de 65-120 segundos aproximadamente. Por lo tanto, puesto que se utilizaron 50 generaciones y 50 individuos como parámetros del AG, el tiempo total de cada corrida del AG fue de 50 x 50 x 65 = 162,500 segundos = 45 hrs. hasta 50 x 50 x 120 = 300,000 segundos = 83 hrs. aproximadamente. Cabe destacar que el espacio de soluciones es de $2^{19} = 524,288$ evaluaciones y como cada evaluación toma de 65-120 segundos, resolver este problema por fuerza bruta tomaría 524,288 x 65 = 34,078,720 segundos = 394 días hasta 524288 x 120 = 62,914,560

segundos = 728 días aproximadamente y todo esto sólo para cada uno de los experimentos. Es por esto que esta propuesta es capaz de usar un método inteligente como el AG, que encuentra buenas soluciones realizando pocas evaluaciones y obteniendo la mejor tasa de detección para cada uno de los experimentos. Además, si el vector inicial de características aumentará, la cantidad de evaluaciones necesarias por el método de fuerza bruta también incrementaría exponencialmente.

5.1.4. Experimentos específicos

El objetivo principal de estos experimentos es obtener el conjunto de características que maximiza la tasa de detección de una botnet específica. Etiquetando los datos de la botnet específica que se desea detectar y los demás datos con una etiqueta para datos que no se desean detectar. Se llevaron a cabo doce experimentos diferentes.

Las botnets evaluadas en cada experimento fueron Storm, Waledac, Neris y RBot. Básicamente, estos doce experimentos se pueden dividir en 3 diferentes grupos de 4 experimentos cada uno.

El primer grupo contiene un conjunto de entrenamiento y prueba con una sola de las botnets (Storm, Waledac, Neris y RBot) y conexiones que no son botnets. El objetivo de estos experimentos es el encontrar las características de una botnet específica para maximizar la tasa de detección de las botnets, donde se diferencie de las conexiones que no son botnets. Estos experimentos se muestran en las Tablas 5.3a, 5.3b, 5.3c y 5.3d respectivamente.

El segundo grupo contiene un conjunto de entrenamiento y prueba con botnets de un solo tipo, descentralizadas (Storm y Waledac) o centralizadas (Neris y RBot), combinado con conexiones que no son botnets. El objetivo de estos experimentos es el encontrar las características de una botnet específica para maximizar la tasa de detección de las botnets, donde se diferencie de las demás botnets del mismo tipo (centralizadas o descentralizadas) y de las conexiones que no son botnets. Estos experimentos se muestran en las Tablas 5.4a, 5.4b, 5.4c y 5.4d para las botnets Storm, Waledac, Neris y RBot respectivamente.

El tercer grupo contiene un conjunto de entrenamiento y prueba con bot-

nets con los dos tipos de botnets descentralizadas (Storm y Waledac) y centralizadas (Neris y RBot), combinado con conexiones que no son botnets. El objetivo de estos experimentos es el encontrar las características de una botnet específica para maximizar la tasa de detección de las botnets, donde se diferencie de las demás botnets de cualquier tipo (centralizadas y descentralizadas) y de las conexiones que no son botnets. Estos experimentos se muestran en las Tablas 5.5a, 5.5b, 5.5c y 5.5d para las botnets Storm, Waledac, Neris y RBot respectivamente.

Todas estas tablas muestran los datos utilizados en los experimentos. La primera columna corresponde a la botnet en el conjunto de datos; la segunda a la clase o la etiqueta para identificar los datos; finalmente, la tercera el número de conexiones que se utilizan para cada botnet.

Tabla 5.3a: Grupo 1. Experimento específico 1 para Storm.

Botnet	Clase	Número de conexiones
Storm	Storm	22,888
NO Botnet	No Storm	22,888

Tabla 5.3b: Grupo 1. Experimento específico 2 para Waledac.

Botnet	Clase	Número de conexiones
Waledac	Waledac	34,442
NO Botnet	No Waledac	34,442

Tabla 5.3c: Grupo 1. Experimento específico 3 para Neris.

Botnet	Clase	Número de conexiones
Neris	Neris	33,084
NO Botnet	No Neris	33,084

Tabla 5.3d: Grupo 1. Experimento específico 4 para RBot.

Botnet	Clase	Número de conexiones
RBot	RBot	34,217
NO Botnet	No RBot	34,217

Tabla 5.4a: Grupo 2. Experimento específico 1 para Storm.

Botnet	Clase	Número de conexiones
Storm	Storm	22,888
Waledac	No Storm	11,444
NO Botnet	No Storm	11,444

Tabla 5.4b: Grupo 2. Experimento específico 2 para Waledac.

Botnet	Clase	Número de conexiones
Storm	No Waledac	17,221
Waledac	Waledac	34,442
NO Botnet	No Waledac	17,221

Tabla 5.4c: Grupo 2. Experimento específico 3 para Neris.

Botnet	Clase	Número de conexiones
Neris	Neris	33,084
RBot	No Neris	16,542
NO Botnet	No Neris	16,542

Tabla 5.4d: Grupo 2. Experimento específico 4 para RBot.

Botnet	Clase	Número de conexiones
Neris	No RBot	17,109
RBot	RBot	34,217
NO Botnet	No RBot	17,108

Tabla 5.5a: Grupo 3. Experimento específico 1 para Storm.

Botnet	Clase	Número de conexiones
Storm	Storm	22,888
Waledac	No Storm	5,722
Neris	No Storm	5,722
RBot	No Storm	5,722
NO Botnet	No Storm	5,722

Tabla 5.5b: Grupo 3. Experimento específico 2 para Waledac.

Botnet	Clase	Número de conexiones
Storm	No Waledac	8,610
Waledac	Waledac	34,442
Neris	No Waledac	8,610
RBot	No Waledac	8,610
NO Botnet	No Waledac	8,610

Tabla 5.5c: Grupo 3. Experimento específico 3 para Neris.

Botnet	Clase	Número de conexiones
Storm	No Neris	8,271
Waledac	No Neris	8,271
Neris	Neris	33,084
RBot	No Neris	8,271
NO Botnet	No Neris	8,271

Tabla 5.5d: Grupo 3. Experimento específico 4 para RBot.

Botnet	Clase	Número de conexiones
Storm	No RBot	8,554
Waledac	No RBot	8,554
Neris	No RBot	8,554
RBot	RBot	34,217
NO Botnet	No RBot	8,554

5.1.5. Experimentos generales

El objetivo principal de estos experimentos es obtener el conjunto de características que maximiza la tasa de detección de un tipo de botnet y botnets para en general. Se llevaron a cabo tres experimentos diferentes.

El primer experimento general denominado ISOT contiene las botnets descentralizadas Storm y Waledac que se muestra en la Tabla 5.6a.

El segundo experimento general denominado ISCX contiene las botnets centralizadas Neris y RBot que se muestra en la Tabla 5.6b.

El tercer experimento general denominado ISOT+ISCX contiene ambas botnets centralizadas Storm y Waledac, así como botnets descentralizadas Neris y RBot que se muestra en la Tabla 5.6c.

Estas tablas muestran los datos utilizados en los experimentos. La primera columna corresponde a la botnet en el conjunto de datos; la segunda muestra a la clase o la etiqueta para identificar los datos; finalmente, la tercera muestra el número de conexiones que se utilizan para cada botnet.

Tabla 5.6a: Experimento general 1 para ISOT.

Botnet	Clase	Número de conexiones
Storm	Botnet	22,888
Waledac	Botnet	34,442
NO Botnet	No Botnet	77,586

Tabla 5.6b: Experimento general 2 para ISCX.

Botnet	Clase	Número de conexiones
Neris	Botnet	33,084
RBot	Botnet	34,217
NO Botnet	No Botnet	76,175

Tabla 5.6c: Experimento general 3 para ISOT+ISCX.

Botnet	Clase	Número de conexiones
Storm	Botnet	22,888
Waledac	Botnet	34,442
NO Botnet	No Botnet	77,586
Neris	Botnet	33,084
RBot	Botnet	34,217
NO Botnet	No Botnet	76,175

5.2. Resultados

En esta sección se presentan los resultados de la experimentación, la comparación y el análisis de los resultados. Se muestra el conjunto de características y la tasa de detección obtenido para cada experimento, también la comparación contra el estado del arte.

5.2.1. Resultados de las características específicas

Para verificar la robustez de la propuesta se repitieron los experimentos específicos del grupo dos (experimento 1, 2, 3 y 4) 10 veces, los cuales se muestran en la Tabla 5.7. Esta Tabla muestra en su primera columna la botnet utilizada; la segunda la mejor tasa de detección obtenida de las 10 corridas; la tercera la peor tasa de detección obtenida de las 10 corridas; la cuarta muestra las tasas de detección promedio de las 10 corridas; finalmente, la quinta muestra la desviación estándar de las tasas de detección de las 10 corridas.

Tabla 5.7: Estadísticas específicas de las 10 corridas.

Botnet	Mejor	Peor	Promedio	Desviación estándar
Storm	97.58 %	96.17 %	96.61 %	0.607036
Waledac	97.12 %	97.04 %	97.09 %	0.024701
Neris	84.92 %	84.56 %	84.74 %	0.106575
RBot	99.58 %	99.57 %	99.58 %	0.005782

Las estadísticas de los resultados específicos de correr los experimentos 10 veces de la Tabla 5.7 de los experimentos del grupo 2 (experimento 1, 2, 3 y 4) muestran que la diferencia del peor resultado obtenido comparado con el mejor fue baja, con diferencias de 1.41 %, 0.08 %, 0.36 %, 0.01 % para Storm, Waledac, Neris y RBot respectivamente. Además, el promedio obtenido fue similar a la mejor detección, así como la desviación estándar por debajo del 0.61, con lo cual se puede observar que la propuesta otorgó tasa de detección alta y en la mayoría de las corridas los resultados muestran poca dispersión, por lo que es capaz de entregar la mejor tasa de detección en la mayoría de las corridas.

En la Tabla 5.8 se muestra el mejor conjunto de características obtenidas en cada uno de los experimentos específicos, los resultados son los mejores de las 10 corridas. Esta Tabla muestra en la primera columna el grupo y número de experimento; la segunda muestra la botnet; la tercera muestra la mejor tasa de detección; la cuarta muestra los falsos positivos; la quinta muestra los individuos del AG correspondiente al mejor conjunto de características obtenido que debe ser interpretado de acuerdo a la siguiente lista: [BytesAB, BytesBA, NPackets, NPacketsAB, NPacketsBA, Duración, APL, DPS, Payload, TBT, Flen, NNP, NSP, PSP, IPP, OPP, PV, BS, PPS]; finalmente, la sexta muestra la cantidad de características.

Tabla 5.8: Resultados de las características específicas.

Experimento	Botnet	Tasa de detección	Falsos positivos	Conjunto de características	No.
G.1 Exp. 1	Storm	99.14 %	1.11 %	[1101110110110001000]	10
G.1 Exp. 2	Waledac	99.69 %	0.53 %	[1101101111110011110]	14
G.1 Exp. 3	Neris	93.02 %	4.43 %	[1101111100101101110]	13
G.1 Exp. 4	RBot	99.82 %	0.30 %	[0111111110110000000]	10
G.2 Exp. 1	Storm	97.58 %	2.08 %	[1010100100111100000]	8
G.2 Exp. 2	Waledac	97.12 %	1.36 %	[0011010110110000100]	8
G.2 Exp. 3	Neris	84.92 %	5.11 %	[1101011110100110100]	11
G.2 Exp. 4	RBot	99.58 %	1.04 %	[1101101110101100000]	10
G.3 Exp. 1	Storm	94.39 %	6.95 %	[1001111100101000001]	9
G.3 Exp. 2	Waledac	98.01 %	7.40 %	[1001011011110011100]	11
G.3 Exp. 3	Neris	94.90 %	17.88 %	[1111100010111100101]	12
G.3 Exp. 4	RBot	99.53 %	4.39 %	[1111001111110111100]	14

Con los experimentos específicos se obtuvo el conjunto de características de una botnet descentralizada y una botnet centralizada. Los resultados de la botnet descentralizada Storm se muestran en los resultados de los experimentos 1, 5 y 8 de la Tabla 5.8 y muestran 4 características similares BytesAB, NPacketsBA, DPS y FLen, de un total de 10, 8 y 9 características respectivamente, obteniendo así casi un 50 % de similitudes de las características en cada uno de los experimentos de Storm. Las tasas de detección de estos resultados van desde el 94 % hasta el 99 % y los falsos positivos desde el 1.11 %

hasta el 6.95 %.

Los resultados de la botnet descentralizada Waledac se muestran en los resultados de los experimentos 2, 6 y 9 de la Tabla 5.8 y muestran 5 características similares NPacketsAB, Payload, Flen, NNP y PV, de un total de 14, 8 y 11 características respectivamente, obteniendo así casi un 50 % de similitudes de las características en cada uno de los experimentos de Waledac. Las tasas de detección de estos resultados van desde el 98.01 % hasta el 99.69 % y los falsos positivos desde el 0.53 % hasta el 7.40 %.

Los resultados de la botnet centralizada Neris se muestran en los resultados de los experimentos 3, 7 y 10 de la Tabla 5.8 y muestran 6 características similares BytesAB, BytesBA, NPacketsAB, Flen, PSP, PV, de un total de 13, 11 y 12 características respectivamente, obteniendo así casi un 50 % de similitudes de las características en cada uno de los experimentos de Neris. Las tasas de detección de estos resultados van desde el 84.92 % hasta el 93.02 % y los falsos positivos desde el 0.53 % hasta el 17.88 %.

Los resultados de la botnet centralizada RBot se muestran en los resultados de los experimentos 4, 8 y 12 de la Tabla 5.8 y muestran 6 características similares BytesBA, NPacketsAB, APL, DPS, Payload y Flen, de un total de 10, 10 y 14 características respectivamente, obteniendo así casi un 50 % de similitudes de las características en cada uno de los experimentos de RBot. Las tasas de detección de estos resultados van desde el 99.53 % hasta el 99.82 % y los falsos positivos desde el 0.30 % hasta el 4.39 %.

Se obtuvo casi un 50 % de similitudes en características en general para cada experimento en cada grupo para la misma botnet.

5.2.2. Resultados de las características generales

Para verificar la robustez de la propuesta, se repitieron los experimentos generales 10 veces, los cuales se muestran en la Tabla 5.9. Esta Tabla muestra en su primera columna el experimento; la segunda la mejor tasa de detección obtenida de las 10 corridas; la tercera la peor tasa de detección obtenida de las 10 corridas; la cuarta muestra las tasas de detección promedio de las 10 corridas; finalmente, la quinta muestra la desviación estándar de las tasas de detección de las 10 corridas.

Tabla 5.9: Estadísticas generales de las 10 corridas.

Experimento	Mejor	Peor	Promedio	Desviación estándar
ISOT	99.46 %	99.42 %	99.44 %	0.014946
ISCX	95.58 %	95.51 %	95.55 %	0.023309
ISOT+ISCX	96.52 %	96.51 %	96.52 %	0.003862

Las estadísticas de los resultados específicos de correr los experimentos 10 veces de la Tabla 5.9 muestran que la diferencia del peor resultado obtenido comparado con el mejor fue baja, con diferencias de 0.04 %, 0.07 %, 0.01 para ISOT, ISCX y ISOT+ISCX respectivamente. Además, el promedio obtenido fue similar a la mejor detección, así como la desviación estándar por debajo del 0.02, con lo cual se puede observar que la propuesta otorgó tasa de detección alta y en la mayoría de las corridas los resultados muestran poca dispersión, por lo que es capaz de entregar la mejor tasa de detección en la mayoría de las corridas.

En la Tabla 5.10 se muestra el mejor conjunto de características obtenidas en cada uno de los experimentos generales, los resultados son los mejores de las 10 corridas. La Tabla muestra en su primera columna el experimento; la segunda muestra la mejor tasa de detección; la tercera muestra los falsos positivos; la cuarta muestra los individuos del AG correspondiente al mejor conjunto de características obtenido que debe ser interpretado de acuerdo a la siguiente lista: [BytesAB, BytesBA, NPackets, NPacketsAB, NPacketsBA, Duración, APL, DPS, Payload, TBT, Flen, NNP, NSP, PSP, IPP, OPP, PV, BS, PPS]; finalmente, la quinta muestra la cantidad de características.

Tabla 5.10: Resultados de las características generales.

Experimento	Tasa de detección	Falsos positivos	Conjunto de características	No.
ISOT	99.46 %	0.57 %	[1010110011111100001]	11
ISCX	95.58 %	2.24 %	[1100011110111100111]	13
ISOT+ISCX	96.52 %	1.23 %	[1101011111111001111]	15

En los experimentos generales para detectar un tipo de botnet, se tiene un

conjunto de datos reducido de 10 y 11 para ISOT (botnets descentralizados) e ISCX (botnets centralizados), respectivamente. La similitud entre ISOT y ISCX es de 7 características: NPacketsAB, APL, Payload, OPP, Flen, NNP, PV, esto significa que ISOT consiguió 70 % y ISCX 63 % de similitud entre ellos. Las tasas de detección de estos experimentos muestran un 99,46 % para ISOT y 95,58 % para ISCX y podría ser mayor si Neris hubiese conseguido una mayor tasa de detección.

En el último experimento que muestra las características necesarias para detectar cualquier tipo de botnet, se combinaron los conjuntos de datos ISOT y ISCX, donde se llamó a este experimento ISOT+ISCX. En este experimento sólo 2 características no se incluyen en los otros experimentos generales y contiene 13 en total, aquellas que no fueron similares son: BytesBA y PSP. La tasa de detección de este experimento muestra un 96,52 % y podría ser mayor si Neris hubiese conseguido una mayor tasa de detección.

5.3. Análisis de resultados

Para el análisis de resultados en esta tesis se realizaron varios tipos de pruebas, con el fin de verificar la efectividad de la propuesta, estas pruebas son: comparación de resultados con los trabajos del estado del arte, utilización de un conjunto de prueba, una agrupación de datos usando algoritmos de agrupación (*clustering*), una prueba con un conjunto reducido de datos para explorar el espacio de solución por completo, entre otras.

5.3.1. Comparación de resultados

En la Tabla 5.11 se muestra una comparación entre los resultados con algunos trabajos del estado del arte, además se muestra una comparación del mejor conjunto/subconjunto de características obtenidas con la propuesta contra la evaluación incluyendo todas las características y una comparación del mejor conjunto de características obtenido con la propuesta, pero usando un algoritmo del estado del arte llamado *RepTree*, el cual es mostrado en los 2 últimos renglones de la Tabla 5.11. La Tabla muestra en su primera columna la referencia de los trabajos correspondientes; la segunda los algoritmos utilizados; la tercera el conjunto de datos utilizado; finalmente, la cuarta y quinta muestra la tasa de detección y falsos positivos.

Tabla 5.11: Comparación de resultados con los trabajos del estado del arte y la propuesta.

Referencia	Algoritmo	Conjunto de datos	Tasa de detección	Falsos positivos
K. Huseynov [50]	*K-Means*	ISOT	82.1 %	2.4 %
K. Huseynov [50]	Colonia de Hormigas	ISOT	67.8 %	23.5 %
S. Saad [46]	*SVM*	ISOT	97.8 %	5.1 %
D. Zhao [48]	*RepTree*	ISOT	98.3 %	0.01 %
P. Narang [51]	*C4.5*	ISOT	98.7 %	0.04 %
Esta propuesta de tesis	Mejor conjunto características	ISOT	99.46 %	0.57 %
	Considerando las 19 características + *C4.5*	ISOT	98.25 %	1.9 %
	Mejor conjunto características + *RepTree*	ISOT	99.2 %	0.8 %
	Considerando las 19 características + *RepTree*	ISOT	98.1 %	2.1 %

Los resultados obtenidos en la Tabla 5.11 muestran una mejora de más del 1 % en la tasa de detección con el mejor conjunto de características obtenido respecto a todas las características. Además, la propuesta con el mejor conjunto de características obtiene mayor tasa de detección que los resultados del estado del arte, donde obtuvo un 0.76 % de diferencia contra P. Narang [51] que fue el que mayor tasa de detección obtuvo en los trabajos del estado del arte. En la comparación del mejor conjunto de características obtenido con la propuesta pero usando el algoritmo de *RepTree* la tasa de detección con las mejores características e incluyendo todas, fue de 99.2 % y 98.1 % respectivamente, donde casi existe un 1 % de diferencia en la tasa de detección. Esto demuestra que el mejor conjunto de características obtenidas con la propuesta, puede ser aplicado a otros algoritmos como es el *RepTree*.

5.3.2. Prueba de propuesta utilizando conjunto de prueba

Se realizó una prueba de la propuesta utilizando un conjunto de prueba para cada uno de los experimentos tanto específicos como generales. El objetivo de esta prueba es simular datos provenientes de un entorno real, con la cual la propuesta no ha sido entrenada. El conjunto de prueba fue obtenido aleatoriamente y representa un 10 % de los datos en cada conjunto de datos de los experimentos, el 90 % restante, se utilizó para generar las características con la propuesta utilizando los mismos parámetros. Una vez generada las características se entrenó con el 90 % de los datos y se realizó la deducción sobre el conjunto de prueba. La Tabla 5.12 muestra los resultados de esta prueba. En su primera columna el experimento; la segunda muestra la botnet; la tercera muestra la tasa de detección; finalmente, la cuarta muestra los falsos positivos.

Tabla 5.12: Prueba de propuesta utilizando conjunto de prueba.

Experimento	Botnet	Tasa de detección	Falsos positivos
G.1 Exp. 1	Storm	99.68 %	0.49 %
G.1 Exp. 2	Waledac	99.84 %	0.23 %
G.1 Exp. 3	Neris	99.90 %	0.14 %
G.1 Exp. 4	RBot	99.92 %	0.15 %
G.2 Exp. 1	Storm	98.19 %	2.07 %
G.2 Exp. 2	Waledac	97.20 %	1.35 %
G.2 Exp. 3	Neris	85.97 %	4.42 %
G.2 Exp. 4	RBot	99.37 %	1.96 %
G.3 Exp. 1	Storm	96.26 %	4.15 %
G.3 Exp. 2	Waledac	98.22 %	5.23 %
G.3 Exp. 3	Neris	97.45 %	19.40 %
G.3 Exp. 4	RBot	99.66 %	5.31 %
General 1	ISOT	99.42 %	0.74 %
General 2	ISCX	95.78 %	2.36 %
General 3	ISOT+ISCX	96.29 %	1.22 %

Los resultados de la prueba de propuesta utilizando un conjunto de prueba muestran que la tasa de detección para cada uno de los experimentos tanto

específicos como generales se mantuvo muy alta y la tasa de detección, así como los falsos positivos son similares comparándolos con los resultados de las Tablas 5.10 y 5.8 que pertenecen a la propuesta utilizando el 100 % de los datos. Por lo tanto, se puede afirmar que las características resultantes mantienen una tasa de detección alta contra datos en un entorno real.

5.3.3. Agrupación de datos

Se realizó una agrupación de datos en un primer conjunto antes de que los datos sean separados en el **conjunto de entrenamiento** y el **conjunto de validación** con el fin de identificar que tan separables son los datos entre las clases definidas en cada experimento y un segundo conjunto de datos una vez obtenido el mejor conjunto de características para compararlo con el primero y observar si los datos son más separables entre sí una vez utilizado el conjunto con las mejores características. Se utilizó el algoritmo de agrupación *K-means* implementado en la herramienta weka [61] con un valor de 100 en el parámetro número de *clusters*. Cada cluster generado es identificado como Botnet o No Botnet dependiendo de la cantidad de datos existentes en el cluster que sean mayoría.

En la Tabla 5.13 se muestra la separabilidad de los datos en cada experimento (específico y general) usando el conjunto de datos con las 19 características; mientras que en la Tabla 5.14 se muestra la separabilidad de los datos en cada experimento (específico y general) usando el conjunto de datos con el mejor conjunto de características obtenido. Las Tablas muestra en su primera columna el experimento; la segunda la cantidad de clusters identificados como Botnet; la tercera la cantidad de clusters identificados como NO Botnet; la cuarta el promedio de datos correspondientes a los clusters; finalmente, la quinta la desviación estándar de datos correspondientes a los clusters.

Tabla 5.13: Separabilidad del conjunto de datos con 19 características.

Experimento	# Cluster Botnet	# Cluster NOBotnet	Promedio	Desviación estándar
G1. Exp. 1	37	61	85.61 %	37.76
G1. Exp. 2	31	66	92.43 %	41.20
G1. Exp. 3	42	57	78.60 %	32.72

G1. Exp. 4	30	68	90.83 %	38.31
G2. Exp. 1	42	56	82.49 %	37.92
G2. Exp. 2	30	67	89.99 %	38.62
G2. Exp. 3	38	39	75.58 %	31.06
G2. Exp. 4	39	59	87.15 %	38.88
G3. Exp. 1	35	58	82.40 %	34.28
G3. Exp. 2	28	62	89.60 %	36.33
G3. Exp. 3	49	46	75.76 %	30.12
G3. Exp. 4	31	69	90.62 %	37.10
ISOT	32	67	90.64 %	40.99
ISCX	28	54	81.13 %	33.24
ISOT+ISCX	40	56	76.86 %	30.06

Tabla 5.14: Separabilidad del conjunto de datos con el mejor conjunto de características.

Experimento	# Cluster Botnet	# Cluster NOBotnet	Promedio	Desviación estándar
G1. Exp. 1	42	57	87.69 %	40.17
G1. Exp. 2	32	63	92.67 %	41.21
G1. Exp. 3	47	52	80.25 %	34.16
G1. Exp. 4	33	63	93.31 %	43.32
G2. Exp. 1	39	54	84.39 %	33.76
G2. Exp. 2	28	72	90.22 %	37.10
G2. Exp. 3	38	42	76.30 %	30.57
G2. Exp. 4	17	83	92.01 %	30.36
G3. Exp. 1	32	59	85.02 %	35.48
G3. Exp. 2	30	64	90.75 %	37.94
G3. Exp. 3	48	52	79.16 %	33.27
G3. Exp. 4	17	83	94.27 %	31.64
ISOT	33	64	92.52 %	39.34
ISCX	35	61	81.26 %	32.88
ISOT+ISCX	45	55	77.95 %	29.18

Los resultados de la agrupación de los datos de la Tabla 5.14 muestra que los datos pertenecientes a las botnets o no son muy similares entre sí y existe

un promedio mayor y desviación estándar más pequeña comparándolo contra los resultados de la Tabla 5.13. Por lo tanto, se puede afirmar que el mejor conjunto de características que se obtuvo con la propuesta facilita la separación entre los datos pertenecientes a la botnet o no, de tal manera que se pueden clasificar con mayor facilidad.

5.3.4. Prueba de propuesta con conjunto de datos reducido

Se realizó una prueba con el conjunto de datos reducido proveniente del 1er experimento general, se tomó una muestra de 250 conexiones provenientes de Storm, 250 de Waledac y 500 de Neris. El objetivo de la prueba es explorar todo el espacio de solución y posteriormente usar la propuesta de selección de características para verificar si es capaz de entregar el mejor conjunto de características para este conjunto de datos reducido.

La Tabla 5.15 muestran los datos utilizados en ese conjunto reducido. La primera columna corresponde a la botnet en el conjunto de datos; la segunda a la clase o la etiqueta para identificar los datos; finalmente, la tercera el número de conexiones que se utilizan para cada botnet.

Tabla 5.15: Conjunto de datos reducido proveniente del experimento general 1.

Botnet	Clase	Número de conexiones
Storm	Botnet	22,888
Waledac	Botnet	34,442
NO Botnet	No Botnet	77,586

La Tabla 5.16 muestra las estadísticas de haber corrido esta prueba con la propuesta en 10 ocasiones. La primera columna la mejor tasa de detección obtenida de las 10 corridas; la segunda el mejor conjunto de características obtenido con la mejor tasa de detección obtenido que debe ser interpretado de acuerdo a la siguiente lista: [BytesAB, BytesBA, NPackets, NPacketsAB, NPacketsBA, Duración, APL, DPS, Payload, TBT, Flen, NNP, NSP, PSP, IPP, OPP, PV, BS, PPS]; la tercera la peor tasa de detección obtenida de las 10 corridas; la cuarta muestra las tasas de detección promedio de las 10

corridas; finalmente, la quinta muestra la desviación estándar de las tasas de detección de las 10 corridas.

Tabla 5.16: Estadísticas de la prueba con el conjunto de datos reducido.

Mejor	Características del mejor	Peor	Promedio	Desviación estándar
92 %	[1110100000110101001]	91.2 %	91.8 %	0.28284

Cabe destacar que el mejor de las 10 corridas con la propuesta se obtuvo en 6 de las 10 ocasiones. El espacio de solución también fue explorado por medio de búsqueda exhaustiva, el mejor conjunto en todo el espacio de solución para el conjunto reducido es 92 %, el mismo obtenido en 6 de las 10 ocasiones por la propuesta de selección. El mejor conjunto de características es [1110100000110101001] y contiene 9 características, de las cuales comparte 7 con el conjunto original.

5.3.5. Discusión de análisis de resultados

Las características propuestas en el estado del arte están enfocadas en detectar las botnets tipo descentralizadas, por esta razón es que puede verse afectada la tasa de detección en las botnets centralizadas.

Las botnets centralizadas son más fáciles de detectar que las descentralizadas debido a el tipo de comunicación entre sí, esto porque las centralizadas mantienen comunicación con un servidor central de C&C, mientras que las descentralizadas no tienen un servidor de C&C como tal.

La prueba de la propuesta utilizando un conjunto de prueba muestra que las características resultantes obtienen resultados similares comparado con la propuesta usando el 100 % de los datos en cada experimento.

La agrupación de los datos muestra que los datos pertenecientes a las botnets o no son muy similares entre sí y existe más separabilidad con el mejor conjunto de características.

La prueba del conjunto reducido obtiene el mejor conjunto de características de todo el espacio de solución por lo que debería funcionar de la misma manera en el conjunto original.

Un vector de características más grande y representativo ayudaría a mejorar la tasa de detección.

Los resultados de los experimentos específicos para cada grupo, muestran al grupo 1 como el grupo con mejor tasa de detección. Esto debido a que, en este grupo, las botnets se diferenciaron sólo contra las conexiones que no eran pertenecientes a las botnets, lo que permitió obtener una mejor tasa de detección y falsos positivos más bajos.

En los experimentos del grupo 2 adicionalmente de las conexiones que no eran pertenecientes a las botnets, cada botnet evaluada se diferenció contra las conexiones de una botnet del mismo tipo (descentralizada o centralizada), esto otorgó una tasa de detección más baja y falsos positivos más altos que en el grupo 1, pero detección más alta y falsos positivos más bajos que en el grupo 2.
En el grupo 3 se obtuvo una tasa de detección similar al grupo 2, pero los falsos positivos fueron los más altos. Esto debido a que en estos experimentos cada botnet evaluada adicionalmente de las conexiones que no eran pertenecientes a las botnets se diferenció de entre todas las demás botnets.

Las apariciones de las características específicas en los 12 experimentos específicos ordenadas de mayor a menor aparición se pueden observar en la Tabla 5.17.

Tabla 5.17: Aparición de características específicas.

Característica	Aparición
Flen	12
NPacketsAB	11
BytesAB	10
DPS	10
Payload	9
BytesBA	8

NPacketsBA	8
APL	8
NNP	8
Duración	7
PV	7
PSP	6
NPackets	5
NSP	5
OPP	5
IPP	4
TBT	3
BS	2
PPS	2

En la aparición de características, sólo Flen apareció en cada uno de los 12 experimentos, es por ello que esta característica es esencial para la detección especifica de las botnets, mientras que la aparición de BBS y PPS fue de 2 para cada una de ellos, por lo que estas dos características tienen muy poco impacto para la detección especifica de botnets.

Las apariciones de las características generales en los 3 experimentos generales ordenadas de mayor a menor aparición se pueden observar en la Tabla 5.18.

Tabla 5.18: Aparición de características generales.

Característica	Aparición
Flen	3
BytesAB	3
Duración	3
Payload	3
NNP	3
NSP	3
PPS	3
PSP	2
BytesBA	2

APL	2
DPS	2
TBT	2
PV	2
BS	2
NPackets	1
NPacketsAB	1
NPacketsBA	1
OPP	1
IPP	0

En la aparición de características, sólo Flen apareció en cada uno de los 3 experimentos generales y en cada uno de los 12 experimentos específicos, es por ello que esta característica es esencial para la detección de las botnets, mientras que la aparición de IPP fue de 0, lo que significa que la característica no fue necesaria para la detección general de las botnets.

Como se muestra en la comparación de los resultados, se obtuvo una mejora significativa en la tasa de detección sobre otros trabajos representativos del estado del arte, los cuales se muestran en la Tabla 5.11. Esto debido a que no utilizan una selección de características, ya que propusieron sus características en función de su experiencia y las características no fueron evaluadas. Por otro lado, los trabajos relacionados que sí realizaron una selección de características, en uno de ellos se utilizó un algoritmo *greedy*, estos algoritmos siguen una ruta de experimentos sin una evaluación de su ruta de experimentos, en otro de ellos se realizó un método de agregar/eliminar característica por característica, por lo cual no se evaluó el espacio de solución. La solución propuesta utiliza un AG para guiar los experimentos, creando y evaluando conjuntos de características, para obtener el mejor conjunto de características que tiene la mejor tasa de detección.

5.4. Resumen del capítulo

En este capítulo se presentó el diseño de experimentos, las herramientas utilizadas para el desarrollo de éstos, los datos utilizados, los resultados obtenidos y un análisis de los mismos. Los experimentos fueron divididos en

generales y específicos, los conjuntos de datos fueron extraídos de 2 organismos ISOT e ISCX, se detalló el uso de las herramientas, se detallaron los experimentos y los parámetros utilizados en ellos, los resultados fueron acorde a cada uno de los experimentos, los cuales otorgaron el conjunto de características con la mejor tasa de detección utilizando la propuesta con su respectiva tasa de detección, en el análisis de resultados se realizaron varias pruebas para comprobar la efectividad de la propuesta y la validez de los resultados.

Capítulo 6

Conclusiones y trabajo futuro

En esta investigación se presentó una propuesta para seleccionar un conjunto de características para la detección de botnets en la fase de C&C utilizando un Algoritmo Genético (AG) como algoritmo optimizador y un clasificador C4.5 para la evaluación de los individuos del AG, de lo cual se concluye:

Por medio de la metodología utilizada se permitió cumplir los objetivos y comprobar la hipótesis tras un ajuste en la primera iteración, en la que la hipótesis no se cumplía en todos los casos, resultando parcial, se replanteo la metodología para comprobar de manera exitosa una nueva hipótesis, la cual es abordada en este trabajo.

El objetivo en este proyecto de tesis se cumplió exitosamente debido a que con la propuesta se obtuvo el conjunto de características con una mayor tasa de detección que los trabajos en el estado del arte, aunque cabe resaltar que no se comprobó si el conjunto fue el óptimo.

La propuesta fue capaz de encontrar el mejor conjunto de características para la prueba con el conjunto reducido y evaluar el vector de características y pese a que no se comprobó con los conjuntos originales si obtuvo mayor tasa de detección que los trabajos presentados, por lo que la hipótesis puede considerarse verdadera.

Se señala que el problema de selección de características podría ser resuelto por búsqueda exhaustiva, pero las evaluaciones requeridas tomarían un tiempo considerable para cada uno de los experimentos propuestos y puesto que

se realizaron varios tipos de experimentos, la solución vía búsqueda exhaustiva no resultaría viable. Es por ello que la propuesta es capaz de resolver este problema usando un método inteligente, reduciendo así el número de evaluaciones necesarias para cada experimento. Además, sí el vector inicial de características incrementa, entonces la cantidad de evaluaciones necesarias por el método de búsqueda exhaustiva también incrementaría.

Los resultados muestran una mejora significativa respecto a los resultados mostrados en otros artículos representativos del estado del arte, donde se consiguió una mayor tasa de detección utilizando el mismo conjunto de datos. En el estado del arte sólo dos trabajos realizan una selección de características. El primero usando un algoritmo *greedy* y el segundo elimina característica por característica y las clasifica dependiendo de la tasa de detección que obtenían al quitarla/agregarla, por lo tanto, no se evalúa todo el espacio de soluciones y puede no otorgar la mejor solución.

Se encontró que hubo reducción de características del vector original propuesto, a pesar de que la propuesta no lo consideraba, las características de este consiguieron una tasa de detección más alta que el vector original.

No hay ninguna característica descartable, la cual no sea necesaria para la detección de las botnets, por lo que si son necesarias las 19 características consideradas en la literatura.

El diseño y la ejecución de los experimentos considero repetir en varias ocasiones los experimentos con el fin de comprobar la robustez de la propuesta, en la cual se obtuvieron características similares en cada una de las corridas y una tasa de detección muy similar, donde en varias ocasiones se obtuvo el mejor conjunto de características dentro de la capacidad de la propuesta.

Los experimentos específicos para cada botnet en cada grupo, muestran casi un 50 % de características similares en cada grupo, estas características similares son esenciales para la detección de esas botnets, mientras que en los experimentos generales se obtuvo más de un 63 % de características similares entre los 2 tipos de botnets, donde se observan las características esenciales para la detección de botnets en general. Además, al combinar ambas en el último experimento, sólo 2 características fueron añadidas, con lo cual se observa que se mantuvieron las propiedades de cada tipo de botnet.

En los experimentos específicos que fueron divididos en tres grupos, se observó que tiene una mejor tasa de detección cuando se compara cada botnet contra las conexiones que no son botnets como en el grupo 1, ya que en los demás grupos se obtuvo una tasa de detección más baja y falsos positivos más altos. Esto fue debido a que se mezcló la botnet que se deseaba detectar con más botnets, lo cual empeoró los resultados obtenidos, pero estos experimentos otorgan las características necesarias para la detección de la botnet específica, que diferencian está botnet de las demás y de las conexiones que no son botnets.

La comparación de resultados además de obtener una mejora significativa en la tasa de detección frente al estado del arte, otorgó una mejor detección usando el conjunto de características obtenido con la propuesta, pero evaluando un algoritmo diferente usado en el estado del arte incluyendo todo el vector de características, este algoritmo fue el *Reptree*, el cual obtuvo una diferencia de casi un 1 % en la tasa de detección.

Un posible trabajo futuro sería utilizar conjuntos de datos más grandes que incluyan más botnets para obtener el mejor conjunto de características para estos conjuntos de datos. Por otro lado, también podríamos realizar pruebas con un vector de características más grande para evaluar más características y obtener una mejor o peor tasa de detección dependiendo de las características a evaluar.

Bibliografía

[1] Leonard, S. Xu, y R. Sandhu, *A Framework for Understanding Botnets*, en International Workshop on Advances in Information Security (WAIS at ARES), Fukuoka, Japan, Instituto de Tecnología de Fukuoka, 16-19, Marzo, 2009.

[2] Symantec, *Informe de Symantec sobre las Amenazas a la Seguridad en Internet*, Abril, 2010, recuperado de: https://www.symantec.com/content/es/mx/enterprise/other_resources/ISTR_XV_LAM_Datasheet_SPA.pdf

[3] Symantec, *Internet Security Threat Report*, Abril, 2015, recuperado de: https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347932_GA-internet-security-threat-report-volume-20-2015-social_v2.pdf

[4] ¿Que es un botnet?, *Kaspersky lab. Daily*, recuperado el 30 de agosto de 2016, de https://blog.kaspersky.es/que-es-un-botnet/755/

[5] A. Orvalho, *Botnet Detection by Correlation Analysis*, Universidad Técnica de Dinamarca, informática y modelado matemático.

[6] N. Davis, *Botnet detection using correlated anomalies*, Universidad Técnica de Dinamarca, informática y modelado matemático.

[7] S. Garcia, A. Zunino y M. Campo, *Identifying, Modeling and Detecting Botnet Behaviors in the Network*, Universidad Nacional del Centro de la Provincia de Buenos Aires, tesis doctoral.

[8] F. Breto, P. García y I. Santos, *Detección de tráfico de control de botnets modelizando el flujo de los paquetes de red*, Universidad de Deusto, tesis doctoral.

[9] S. Agarwal, *Performance Analysis of Peer-To-Peer Botnets using "The Storm Botnet" as an Exemplar*, Universidad de Victoria, tesis para obtener el grado de ciencias de la computación.

[10] D. Jang, M. Kim, H. Jung y B. Noh, *Analysis of HTTP2P Botnet : Case Study Waledac* , en IEEE 9th Malaysia International Conference on Communications, 15 - 17, Diciembre, 2009 Kuala Lumpur Malaysia.

[11] Trend Micro, *Taxonomy of botnet threats*, 2006.

[12] D. Dagon, G. Gu, y C. P. Lee. *A taxonomy of botnets structures*, capítulo 8, p. 22, 2007.

[13] E. Cooke, F. Jahanian, y McPherson D., *The zombie roundup: Understanding, detecting, and disrupting botnets*, en Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), capítulo 8, p. 22, 2005.

[14] Igor Kononenko and Matjaz Kukar, *Machine Learning and Data Mining: Introduction to Principles and Algorithms*, Horwood Publishing Limited, West Sussex.

[15] J. R. Quinlan, *Induction of decision trees*, Machine learning, pp. 81–10, 1986.

[16] Steven L. Salzberg, *C4.5: Programs for machine learning*, Machine Learning, capítulo 16, pp. 235–240, 1993, ISSN0885-6125, 10.1007/BF00993309.

[17] J. R. Quinlan, *C4. 5: programs for machine learning*, publicado por Morgan kaufmann, 1993.

[18] C. E. Shannon, *Prediction and Entropy of Printed English*, Bell Systems Technical Journal, pp. 50–64, 1951.

[19] C. E. Shannon y W. Weaver, *The mathematical theory of communication*, 1949.

[20] John H. Holland. *Concerning efficient adaptive systems*, In M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, editors, Self-Organizing Systems, Spartan Books, Washington, pp. 215-230, D.C., 1962.

[21] John H. Holland. *Outline for a logical theory of adaptive systems*, Journal of the Association for Computing Machinery, pp. 297-314, 1962.

[22] A. Brindle, *Genetic Algorithms for Function Optimization*, tesis de doctorado, Department of Computer Science, Universidad de Alberta, Edmonton, Alberta, 1981.

[23] A. Wetzel, *Evaluation of the effectiveness of genetic algorithms in combinatorial optimization*, tesis de doctorado, University of Pittsburgh, Pittsburgh, Philadelphia, USA, 1983.

[24] David H. Ackley, *A Connectionist Machine for Genetic Hillclimbing*, Kluwer Academic Publishers, Boston, Massachusetts, 1987.

[25] Gilbert Syswerda, *AUniform Crossover in Genetic Algorithms*, In J. David Schaffer, editor, en Third International Conference on Genetic Algorithms, San Mateo, California, pp. 2-9, 1989.

[26] William M. Spears y Kenneth A. De Jong, *An Analysis of Multi-Point Crossover*, In Gregory E. Rawlins, editor, Foundations of Genetic Algorithms, San Mateo, California, pp. 301-315, 1991.

[27] Thomas Back, *Optimal Mutation Rates in Genetic Search*, en Stephanie Forrest, editor, en Fifth International Conference on Genetic Algorithms, San Mateo, California, pp. 2-8, Julio, 1993, publicado por Morgan Kaufmann.

[28] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, y S. Lim, *A survey and comparison of peer-to-peer overlay network schemes*, Communications Surveys Tutorials, IEEE, vol. 7, pp. 72-93, 2005.

[29] Craig A. Schiller, Jim Binkley, David Harley, Gadi Evron, Tony Bradley, Carsten Williams, y Michael Cros, *Botnets The Killer Web App*, Syngress capítulo 1, pp. 14-15, 2007.

[30] Ian H. Witten, Eibe Frank, y Mark A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, Tercera edición, capitulo 2, p. 40, 2011, publicado por Morgan Kaufmann.

[31] *Storm chaos prompts virus surg*, recuperado de: http://news.bbc.co.uk/1/hi/technology/6278079.stm

[32] T. Holz, M. Steiner, F. Dahl, E. Biersack, y F. C. Freiling, *Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm*, en LEET (F. Monrose, ed.), USENIX Association, 2008.

[33] I. Arce y E. Levy, *An analysis of the slapper worm*, Security Privacy, IEEE, vol. 1, pp. 82–87, Febrero, 2003.

[34] *Bot software looks to improve peer age*, recuperado de: http://www.securityfocus.com/news/11390

[35] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, y D. Dagon, *Peerto-peer botnets: overview and case study*, en The first conference on First Workshop on Hot Topics in Understanding Botnets, Berkeley, CA, USA, 2007.

[36] P. Maymounkov y D. Mazieres, *Kademlia: A peer-to-peer information system based on the xor metric*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 53-65, 2002.

[37] *edonkey network*, recuperado de: http://en.wikipedia.org/wiki/EDonkey

[38] Wireshark, *Conversations*, recuperado de: https://www.wireshark.org/docs/wsug_html_chunked/ChStatConversations.html

[39] *Aprendizaje supervisado*, En Wikipedia. recuperado el 30 de agosto de 2016, de https://es.wikipedia.org/wiki/Aprendizaje_supervisado

[40] B. González, *Compresión de datos para aprendizaje de máquina*, Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN), tesis de maestría.

[41] *C4.5*, En Wikipedia. recuperado el 30 de agosto de 2016, de https://es.wikipedia.org/wiki/C4.5

[42] C. Coello, *Introducción a la Computación Evolutiva (Notas de Curso)*, CINVESTAV-IPN, Departamento de Computación, Mayo, 2014.

[43] *Spam*, En Wikipedia. recuperado el 30 de agosto de 2016, de https://es.wikipedia.org/wiki/Spam

[44] *Lista negra*, En Wikipedia. recuperado el 30 de agosto de 2016, de https://es.wikipedia.org/wiki/Lista_negra

[45] *Captcha*, En Wikipedia. recuperado el 30 de agosto de 2016, de https://es.wikipedia.org/wiki/Captcha

[46] S. Saad, B. Sayed, J. Felix, I. Traore, D. Zhao, A. Ghorbani, W. Lu y P. Hakimian, *Detecting P2P botnets through network behavior analysis and machine learning*, en Privacy, Security and Trust (PST), Ninth Annual International Conference, Julio 19-21, 2011.

[47] B. Piyush, D. Manoj y K.G. Mrinal, *A Framework for P2P Botnet Detection Using SVM*, en Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 195-200, Octubre, 2012.

[48] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, y D. Garan, *Botnet detection based on traffic behavior analysis and flow intervals*, en 27th IFIP International Information Security Conference, Noviembre, 2013.

[49] E. Beigi, H. Jazi, N. Stakhanova y A. Ghorbani, *Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches*, en conferencia de la IEEE Communications and Network Security (CNS), San Francisco, CA, 29-31 Octubre, 2014.

[50] K. Huseynov, K. Kim y P. Yoo, *Semi-supervised Botnet Detection Using Ant Colony System*, en 31 Symposium on Cryptography and Information Security, Kagoshima, Japón, 21-24, Enero, 2014.

[51] P. Narang, S. Ray, C. Hota y V. Venkatakrishnan, *PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations*, en IEEE Security and Privacy Workshops (SPW), San Jose, CA, 17-18, Mayo, 2014.

[52] Ritu y R. Kaushal, *Role of handshaking packets in improving peer to peer Botnet detection*, en International Conference on Computing and Network Communications (CoCoNet), Departamento de Tecnología de la Información Indira Gandhi Universidad Técnica de Delhi para Mujeres Nueva Delhi, India, 16-19, Diciembre, 2015.

[53] Cormen, Leiserson, y Rivest, *Introduction to Algorithms*, capitulo 17 "Greedy Algorithms", p. 329, 1990.

[54] Information security and object technology (ISOT) research lab, *ISOT Botnet dataset*, University of Victoria, recuperado de: http://www.uvic.ca/engineering/ece/isot/datasets/index.php

[55] The Honeynet Project, *French Chapter of Honeynet*, recuperado de: http://www.honeynet.org/chapters/france.

[56] G. Szab'o, D. Orincsay, S. Malomsoky, y I. Szab'o, *On the validation of traffic classification algorithms*, in Proceedings of the 9th international conference on Passive and active network measurement (PAM'08), Berlin, Heidelberg, pp. 72–81, Springer-Verlag, 2008.

[57] LBNL and ICSI, *LBNL Enterprise Trace Repository*, recuperado de: http://www.icir.org/enterprise-tracing.

[58] A. Shiravi, H. Shiravi, M. Tavallaee, y A. A. Ghorbani, *Toward developing a systematic approach to generate benchmark datasets for intrusion detection*, en Computers & Security, vol. 31, no. 3, pp. 357–374, May, 2012.

[59] Tshark *tshark: Terminal-based Wireshark.*

[60] Wireshark, recuperado de: https://www.wireshark.org/docs/wsug_html_chunked/ChStatConversations.html

[61] E. Ian, H. Witten, L. Trigg, M. Hall, G. Holmes, y S. Cunningham, *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*, 1999.

[62] S. Garner, *Weka: The Waikato environment for knowledge analysis* en el New Zealand Computer Science Research Students Conference, pp. 57–64, 1995.

[63] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, y Ian H. Witten, *The weka data mining software: an update SIGKDD Explor. Newsl.*, 10-18, Noviembre, 2009.

Productos de la investigación

Se realizaron 2 artículos en este proceso de investigación, el primero fue con base en la primera iteración de la metodología y el segundo en la segunda iteración.

· F. Villegas, N. Cruz y E. Aguirre, *Botnets detection using clustering algorithms*, en Core 16, CIC IPN, 7-9, noviembre, 2016.

· F. Villegas, N. Cruz y E. Aguirre, *Feature selection to detect botnets using machine learning algorithms*, en Conielecomp 17, Cholula, Puebla, 22-24, febrero, 2017, (aceptado).