UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE INVESTIGACIÓN

DETECCIÓN DE MALWARE A TRAVÉS DE REDES NEURONALES

PREVIO A LA OBTENCIÓN DEL TÍTULO:

INGENIERO EN TELECOMUNICACIONES

PRESENTADO POR:

21511024 ABNER GERARDO VENTURA PALACIOS

ASESOR: ANA CRISTINA REYES LEIVA

CAMPUS: SAN PEDRO SULA; OCTUBRE, 2020

DEDICATORIA

El presente proyecto va dedicado a mi padre quien fue un gran ejemplo a mi vida, de valentía y el poder realizar objetivos sin importar las circunstancias. Cada día de mis estudios mediante recuerdos y a través de ese imagen en mi mente me inspiro al pensar cuán orgulloso estaría si pudiera ver que he llegado hasta este punto en mi desarrollo como profesional.

Asimismo a mi madre quién ha dado todo su esfuerzo todos estos años, gracias a ella y la gracia de Dios es posible poder culminar mis estudios de pregrado. Por último a cada uno de los docentes y compañeros quienes han sido de mucho apoyo inundándome de su conocimiento y habilidades prácticas e intelectuales.

AGRADECIMIENTO

Agradezco a Dios por regalarme talentos y habilidades y sentirme feliz de poder de desarrollarlas cada vez más. A UNITEC quién con su programa de BECAS pude ingresar a la universidad, recibiendo mucho apoyo de cada uno de los programas académicos.

Todos los docentes y personas no mencionadas, muchas gracias.

Epígrafe

Lo que sabemos es una gota de agua; lo que ignoramos es el océano.

~ *Isaac Newton*

RESUMEN EJECUTIVO

En la actualidad múltiples empresas de desarrollo dedicadas a la detección de software malicioso utilizan técnicas de aprendizaje mejorando los métodos comunes que provocan infecciones a un sistema de computadores. Este trabajo muestra una técnica que detecta software malicioso analizando el comportamiento de resultado final de un clasificador, definiendo parámetros y así conocer si se trata de un programa malicioso. Se mostrarán varios escenarios en el cual tendremos muestras de datos y así poder comprobar si se trata de un componente que realmente realice daño o una falsa alarma de un objeto que no tendrá ningún efecto en el sistema. De esta forma a través de esta investigación se logrará diseñar una red neuronal para la predicción de malware trabajando con diferentes familias para la clasificación de estas mismas, logrando un porcentaje de entrenamiento adecuado para la detección de software malicioso. Asimismo por recomendar ciertos aspectos de mejora y desarrollo para este tipo de investigaciones a través de redes neuronales.

Palabras clave: redes, clasificación de software malintencionado, seguridad, aprendizaje automatizado

ABSTRACT

En la actualidad, múltiples empresas de desarrollo dedicadas a la detección de software malicioso detectan técnicas de aprendizaje mejorado, los métodos comunes que provocan infecciones en un sistema de computadores. Este trabajo muestra una técnica que detecta software malicioso analizando el comportamiento del resultado final de un clasificador, definiendo parámetros y así conocer si se trata de un programa malicioso. Se mostrarán varios escenarios en el cual tendremos muestras de datos y así poder verificar si se trata de un componente que realmente se dañe o una falsa alarma de un objeto que no tenga ningún efecto en el sistema.

Palabras clave: redes, clasificación de software malintencionado, seguridad, aprendizaje automatizado

ÍNDICE DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

LISTA DE SIGLAS Y GLOSARIO

ANN	*Artificial Neural Network* (Red Neuronal Artificial)
CNN	*Convolutional Neural Network* (Red Neuronal Convolucional)
DL	*Deep Learning* (Aprendizaje Profundo)
DNN	*Deep Neural Network* (Red Neuronal Profunda o Densa)
ML	*Machine Learning* (Aprendizaje Automatizado)
NN	*Neural Network* (Red Neuronal)

CAPÍTULO I. INTRODUCCIÓN

En los últimos años numerosos programas de infección por software han llevado a cabo diferentes ataques reportados debido a software etiquetado como troyanos, ransomwares, virus, etc., todo esto referido a un solo concepto nombrado malware. Compañías dedicadas a la protección y seguridad contra estas amenazas reportan cada semana, mes y año informes de nuevos ataques o amenazas comúnmente conocidas desde hace muchos años las cuales aún siguen en función y se ha podido controlar debido al análisis y comportamiento de sus antecedentes. El gran problema es la evolución de nuevos ataques que no logran ser reconocidos por estos sistemas y se convierten así de cierta manera indetectables hasta que se ven afectados usuarios, teniendo una detección tardía y como consecuencia dando una imagen ineficiente. Nombrando algunas de estas empresas conocidas de servicios contra malware como ser Symantec, Mc Afee, Virus Total, Norton antivirus, entre otras compañías comprometidas a la seguridad de un dispositivo portador de un sistema operativo determinado.

CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA

En el capítulo 1 de esta investigación se introdujo a una problemática en un sistema de computador, la cual ha tenido un crecimiento exponencial año tras año. Por lo que en este capítulo se presenta de una mejor manera la idea de esta investigación, como ser el enfoque que lleva, presentar antecedentes, definir la problemática, preguntas de investigación, objetivos y la justificación para la clasificación por de detección de malware.

2.1 PRECEDENTES DEL PROBLEMA

El concepto de malware surge en los años 90 por un Israelí de nombre Yisrael Radai, antes de este término se le definía como "virus informático" nombrado por Fred Cohen en 1983. Debido el crecimiento exponencial del malware el poder distinguir una anomalía que puede realizar daños mínimos o severos en un sistema operativo se torna cada vez más difícil. Esto lo podemos comparar en la vida real con el mundo biológico en el que conocemos infecciones y hemos creado medidas para poder realizar una defensa contra estas, asimismo sucede en el área digital o de computadores.

Como ser formas comunes de una infección, comportamientos y/o con diferentes propósitos por lo que una alerta para distinguir y prevenir aporta mayor facilidad de desinfección contra el malware.No existía algo concreto en la distinción de malware, por lo que la clasificación, diferenciación y nombres de diferentes tipos de malware se torna algo complejo.

El crecimiento masivo de malware se ha reflejado de manera exponencial, para el año 2006 existían alrededor de 1 millón de amenazas y en 2016 hubo un crecimiento hasta de 500 millones de variedad en múltiples formas de malware. Un estudio realizado por McAfee en el año 2007 detectó que los motores de búsqueda a través de internet conducen a páginas web altamente peligrosas. Para el 2010 Norton demostró que más del 10% de búsquedas web tenían una tendencia con resultados maliciosos.

2.2 Definición del Problema

La detección de software malintencionado se basa en métodos como ser: en firmas, lo cuales llevan a cabo bases de datos de malware conocido y así detectar malware. También se basan en heurística, utilizando el malware con patrones y un conjunto de reglas de detección de malware conocido o de nuevos. Para apoyar a esta tarea de detección de malware nuevo y conocido el uso de learning machine ha dado resultados mucho mayores, logrando una detección sin dificultad. Múltiples compañías dedicas a la detección de amenazas lo clasifican de diferentes maneras por lo cual no genera una vista clara en la clasificación del malware. La carencia de propiedades o métricas concretas que distingan de manera única un software benigno requiere un mayor esfuerzo al momento de preparar un conjunto de datos.

2.3 Justificación

La detección de malware es algo esencial en cuanto a la seguridad informática, hoy en día se cuenta con múltiples dispositivos electrónicos los cuales hacen referencia al Internet de las cosas IOT. Todos estos se encuentran conectados en una red LAN ya sea con o sin acceso a Internet logrando realizar tareas de manera presencial hasta remotamente. Se debe garantizar la seguridad para cada uno de los equipos que utilizamos, por lo que se debe tomar conciencia que lo que se encuentra en juego es toda la información la cuál a estas alturas significa dinero y es por eso que hoy existe una gran éxito en el análisis de datos nombrado "Big Data". El uso de nuevas técnicas y la aplicación de redes neuronales impulsan a lograr una seguridad más robusta en los dispositivos los cuales podrían estar expuestos a amenazas desde el momento que tienen acceso a una red. Cada uno de estos debe tener un constante análisis de prevención contra infecciones o vulnerabilidades, así evitar la contaminación de estos y tomar medidas lo más pronto posible evitando daños severos en nuestros equipos.

2.4 Preguntas de Investigación

1) ¿Qué tamaño tendrá la base de datos de entrenamiento y verificación de la red neuronal?
2) ¿A qué tipo de datos deben ser codificados la base de datos de entrenamiento para lograr entrenar la red?

3) ¿Cuáles son los parámetros de la red neuronal artificial?

2.5 Objetivos

En esta sección establezco lo que busco realizar a través de este trabajo, dando respuestas a las preguntas de investigación y el problema generado.

2.5.1 Objetivo General

Diseñar una red neuronal artificial capaz de determinar si un código es un código malicioso (malware) o no, a través del aprendizaje automático.

2.5.2 Objetivos Específicos

1) Establecer la magnitud del conjunto de datos para el entrenamiento y verificación de la red para la detección de malware.
2) Codificar el conjunto de datos para el entrenamiento de la red neuronal para la detección de malware.
3) Determinar los parámetros de la red neuronal artificial para la detección de malware.

CAPÍTULO III. MARCO TEÓRICO

En el capítulo anterior se estableció cuál era el problema objeto de investigación con la finalidad de lograr el análisis de manera completa en un orden que sea cronológico y lógico. En el desarrollo del presente capítulo se pretende sustentar teóricamente la investigación, recopilando información de diversas fuentes primarias y secundarias que nos permitirán explicar y examinar las teorías, conceptos, metodologías, etc. Así como lo menciona Baca Urbina (2010), que un buen marco teórico es aquel que vincula de manera lógica y coherente los conceptos y las proposiciones existentes de estudios anteriores.

3.1 ANÁLISIS DE LA SITUACIÓN ACTUAL: CYBER-ATAQUES

A lo largo de los últimos años, la industria del cibercrimen ha ido incrementando sustancialmente. Los ataques cibernéticos son considerados como un tipo de ataque de internet realizado ya sea por un individuo u organización que va dirigido hacia los sistemas de información, redes o infraestructura. Estos ataques cibernéticos podrían afectar a cualquiera, tanto a organizaciones como a la población en general. De hecho, se dice que sucede un ataque cibernético cada 39 segundos, por lo tanto, es un promedio de 2,244 ataque cibernético al día, además, estos tipos de ataques han aumentado en un 11 % en comparación del año 2018 y en un 67 % en comparación del año 2014 (Bissell et al., 2019).

Por esta razón, (WEFORUM, 2020) ha establecido a los ataques cibernéticos, como uno de los diez principales riesgos que vamos a enfrentar a nivel mundial durante el transcurso del presente año 2020, aumentando tanto los ataques cibernéticos hacia las infraestructuras como en la búsqueda de dinero o datos. Esto se debe principalmente a la globalización del panorama digital y a su constante cambio, donde han evolucionado los objetivos de estos ataques cibernéticos, ya que no solo se busca robo de información, sino que, además, buscan obtener control de los sistemas centrales, apuntando hacia la debilidad humana para solicitar demandas, según los intereses del atacante. Y mientras esto avanza, todos se ven amenazados por algún tipo de ataque cibernético.

En consecuencia, los ataques cibernéticos han causado un gran impacto negativo (Saini et al., 2012), especialmente, sobre la economía tanto de las compañías como de la población general.

Expertos estimaron que en el año 2016, fue depositado, a Bitcoin wallets, alrededor de 1 billón de dólares americanos, asociados solamente por un tipo de ataque cibernético, el ransomware (Conti et al., 2018), de hecho, el costo promedio del delito cibernético para una organización fue de 1.4 millones de dólares americanos y este aumentó a 13 millones de dólares americanos, representando un incremento de más de 9 veces, y para los próximos cinco años, se calculó un valor total en riesgo de 5.2 trillones de dólares americanos por ataques cibernéticos (Bissell et al., 2019).

A causa de esto, es de vital importancia tomar cartas sobre el asunto. Para esto se debe estar preparado, al tener conocimientos sobre los diferentes tipos de ataques cibernéticos y que se debe realizar para prevenir, detectar, predecir y/o neutralizar estos ataques. En la actualidad, se posee conocimiento sobre muchos tipos de ataques cibernéticos (Chapman & Leblanc, 2011; Simmons et al., 2014), sin embargo, se considera que los tipos de ataques cibernéticos más comunes son; el Distributed Denial of Service (DDoS), Phishing attack, Man-in-the-middle (MitM) attack y malware. Por lo tanto, es indispensable tener un vasto conocimiento sobre estos ataques para determinar la acción a seguir para contrarrestar uno de estos ataques cibernéticos.

En 2019, se encontró una escalada de ataques de ransomware sofisticados y específicos. Industrias específicas fueron fuertemente victimizadas, incluidos el gobierno estatal y local y las organizaciones de salud. La nueva y cruda realidad es que los atacantes están dedicando más tiempo a recopilar información sobre sus víctimas, logrando la máxima interrupción y rescates a mayor escala. Los ataques se han vuelto tan dañinos que el FBI ha suavizado su postura anterior sobre el pago de rescates. Ahora reconocen que, en algunos casos, las empresas pueden necesitar evaluar sus opciones para proteger a sus accionistas, empleados y clientes. (Symantec, 2019)

A medida que se desarrollen las redes 5G, el uso de dispositivos IoT conectados se acelerará drásticamente. Aumentarán la vulnerabilidad de las redes a los ciberataques de Gen V multivectoriales a gran escala. Los dispositivos de IoT y sus conexiones a redes y nubes son un eslabón débil de la seguridad. Es difícil obtener visibilidad de estos dispositivos que pueden tener requisitos de seguridad complejos. Lo que se necesita es un enfoque más holístico de la seguridad de IoT, que combine controles tradicionales y nuevos para proteger estas redes en constante crecimiento en todos los sectores industriales y comerciales. La nueva generación de seguridad

se basará en nano agentes de seguridad. Estos micro-complementos pueden funcionar con cualquier dispositivo o sistema operativo en cualquier entorno, controlando todos los datos que fluyen hacia y desde el dispositivo y brindando seguridad siempre activa. (Check Point Research, 2020)

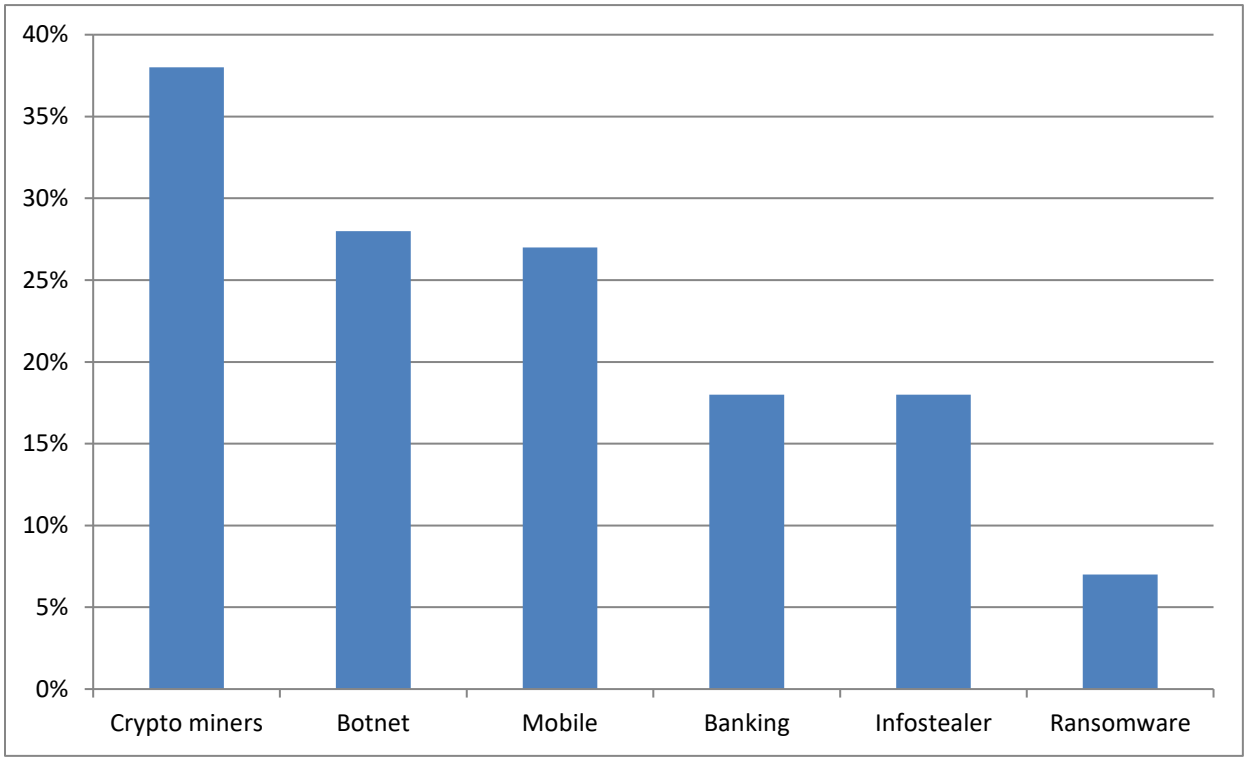

Ilustración 1-Categorías de ataques cibernéticos a nivel global

Fuente: (Check Point Research, 2020)

De acuerdo a la ilustración 1, 38% de los ataques cibernéticos son de tipo crypto miners. Por otro lado 28% de los ataques son botnet. 27% representan ataques cibernéticos a dispositivos móviles, esto incluye a los teléfonos inteligentes. En relación a temas financieros, en especial con cuentas bancarias, representan un 18% de los ataques a nivel global, de igual manera, los ataques para el robo de información. Los ataques ransomware representan el 7% de los ataques cibernéticos a nivel global.

3.2 MALWARE

El malware es un código malicioso, siendo un fragmento de código que es añadido, cambiado o removido en algún sistema, con la finalidad de lograr obtener información delicada, acceso a los sistemas privados o tomar el control sobre todo el sistema (Khlaif Abuzaid et al., 2013). El malware es considerado como el tipo de ataque cibernético más utilizado, y es una ataque que ha tenido un gran incremento, y el mayor impacto económico (Bissell et al., 2019). Este tipo de ataque cibernético, ocurrió por primera vez en el año 1970, desde entonces ha evolucionado para realizar el mayor daño posible (Milošević, 2013).

1) El malware puede manifestarse a través de varios comportamientos aberrantes. Estos son algunos signos reveladores de que tiene malware en el sistema:
2) El ordenador se ralentiza. Uno de los efectos principales del malware es reducir la velocidad del sistema operativo, tanto si navega por Internet como si sólo utiliza sus aplicaciones localmente.
3) La pantalla se llena de oleadas de publicidad fastidiosa que no tendría que mostrarse. Los anuncios emergentes inesperados son un signo típico de infección por malware. Están asociados especialmente con una forma de malware conocida como adware. Es más, los mensajes emergentes suelen ir unidos a otras amenazas de malware ocultas.
4) El sistema se bloquea constantemente o muestra una pantalla azul BSOD (Blue Screen of Death), que puede aparecer en los sistemas Windows cuando se encuentra un error grave.
5) Existe una pérdida misteriosa de espacio disponible en disco, probablemente debido a un ocupante indeseado de malware que se oculta en su disco duro.
6) Se produce un aumento extraño de la actividad del sistema en Internet.
7) La utilización de recursos del sistema es anómalamente elevada y el ventilador del equipo comienza a funcionar a toda velocidad, lo cual señala que la actividad del malware se ha apropiado de recursos del sistema en segundo plano.
8) La página de inicio del navegador cambia sin su permiso. Igualmente, los enlaces en los que hace clic lo llevan a un destino web no deseado. Esto significa normalmente que hizo clic en aquel mensaje emergente de "enhorabuena", que descargó algún software no deseado. También es posible que el navegador responda muy lentamente.

9) El navegador se llena inesperadamente de nuevas barras de herramientas, extensiones o complementos.
10) Su producto antivirus deja de funcionar y no puede actualizarlo, dejándolo desprotegido contra el malware tramposo que lo deshabilitó.
11) También puede producirse un ataque de malware obviamente dañino e intencionadamente provocador. Este es el caso del ransomware, que se anuncia sin disimulo, le dice que tiene sus datos y exige un rescate para devolverle sus archivos.
12) Incluso si todo parece funcionar bien en su sistema, no se confíe, porque no conocer el problema no significa necesariamente que no existe. El malware potente puede ocultarse en lo más profundo de su ordenador y husmear sus datos sin disparar ninguna alarma mientras se apodera de sus contraseñas, roba archivos confidenciales o utiliza su PC para expandirse por otros equipos.

3.2.1 CLASIFICACIÓN

De acuerdo a (Amamra et al., 2012) estos son los malware más comunes que se pueden presentar:

1) El adware es un software no deseado diseñado para mostrar anuncios en su pantalla, normalmente en un explorador. Suele recurrir a un método subrepticio: bien se hace pasar por legítimo, o bien se adosa a otro programa para engañar al usuario e instalarse en su PC, tableta o dispositivo móvil.
2) El spyware es malware que observa las actividades del usuario en el ordenador en secreto y sin permiso, y se las comunica al autor del software.
3) Un virus es malware que se adjunta a otro programa y, cuando se ejecuta —normalmente sin que lo advierta el usuario—, se replica modificando otros programas del ordenador e infectándolos con sus propios bits de código.
4) Los gusanos son un tipo de malware similar a los virus, que se replica por sí solo con el fin de diseminarse por otros ordenadores en una red, normalmente provocando daños y destruyendo datos y archivos.
5) Un troyano, o caballo de Troya, es uno de los tipos de malware más peligrosos. Normalmente se presenta como algo útil para engañar al usuario. Una vez que está en el sistema, los

atacantes que se ocultan tras el troyano obtienen acceso no autorizado al ordenador infectado. Desde allí, los troyanos se pueden utilizar para robar información financiera o instalar amenazas como virus y ransomware.

6) El ransomware es un tipo de malware que bloquea el acceso del usuario al dispositivo o cifra sus archivos y después lo fuerza a pagar un rescate para devolvérselos. El ransomware se ha reconocido como el arma preferida de los delincuentes informáticos porque exige un pago rápido y provechoso en criptomoneda de difícil seguimiento. El código que subyace en el ransomware es fácil de obtener a través de mercados ilegales en línea y defenderse contra él es muy difícil.
7) El rootkit es un tipo de malware que proporciona al atacante privilegios de administrador en el sistema infectado. Normalmente, también se diseña de modo que permanezca oculto del usuario, de otro software del sistema y del propio sistema operativo.
8) Un registrador de pulsaciones de teclas es malware que graba todas las pulsaciones de teclas del usuario, almacena la información recopilada y se la envía al atacante, que busca información confidencial, como nombres de usuario, contraseñas o detalles de la tarjeta de crédito.
9) La minería de criptomonedas maliciosa, denominada también minería fortuita o cryptojacking, es un malware cada vez más prevalente instalado por un troyano. Permite que otras personas utilicen su ordenador para hacer minería de criptomonedas como bitcoin o monero. Los programas maliciosos de minería de criptomonedas utilizan los recursos de su ordenador pero envían los coins obtenidos a sus propias cuentas, no a las del propietario del equipo. En pocas palabras, un programa de minería de criptomonedas malicioso, le roba recursos para hacer dinero.
10) Los exploits son un tipo de malware que aprovecha los errores y vulnerabilidades de un sistema para que el creador del exploit pueda asumir el control. Los exploits están vinculados, entre otras amenazas, a la publicidad maliciosa, que ataca a través de un sitio legítimo que descarga contenido malicioso inadvertidamente desde un sitio peligroso. A continuación, el contenido dañino intenta instalarse en el ordenador tras una descarga involuntaria. Ni siquiera es necesario hacer clic. Todo lo que tiene que hacer es visitar un sitio bueno el día equivocado.

En la actualidad, la cantidad de dispositivos móviles ha aumentado masivamente, por lo tanto, se han convertido en un objetivo de los ataques cibernéticos. En consecuencia, los ataques malware han evolucionado, para el año 2018, se encontraron más de 2000 variantes y más de 20 familias de mobile malware, además, 1 de cada 36 dispositivos móviles posee aplicaciones de alto riesgo instaladas (Symantec, 2019). De hecho, para la primera mitad del año 2019, la cantidad de ataques por mobile banking malware a dispositivos móviles aumento en un 50 % (Check Point Research, 2020; Imran et al., 2018). El mobile malware ataca tanto a los dispositivos móviles of people based and organization based, se estimó que alrededor de un 27 % de todas las organizaciones a nivel mundial, fueron objetivos de un ataque cibernético que involucraba un dispositivo móvil (Check Point Research, 2020). Por lo tanto, los mobile attacks han evolucionado consistentemente, y los tipos de ataques más encontrados son banking Trojans, cryptominers, Remote Access Trojans (RAT), adware and ransomware (Check Point Research, 2020). El tipo de mobile malware más común es el adware, ya que se puede encontrar en aplicaciones como Google Play and App Store, algunos ejemplos de los mobile malware son Necro, Hiddad, and xHelper (Feizollah et al., 2015).

La cantidad de dispositivos IoT han aumentado significativamente a lo largo de los últimos años, y cada vez más dispositivos se van incorporando, como ser tablets, routers, switches modern SCADA, PLC, EPOS, entre otros, además, con el surgimiento de 5G, el incremento de dispositivos IoT se ha ido acelerando dramáticamente, con la aparición de automotive systems, home devices, dispositivos médicos, entre muchos otros (Zhang et al., 2014). Esto hace que incremente la vulnerabilidad de las redes hacia los ataques cibernéticos a gran escala, como ser botnets, lo que convierte a los dispositivos IoT y sus conexiones a la red y a la nube como el eslabón débil con respecto a la seguridad, de hecho, el 67 % de las organizaciones ha experimentado un incidente de seguridad en un dispositivo o red IoT (Check Point Research, 2020). Se considera que los routers y cámaras IP son los principales objetivos de un ataque cibernético en una red IoT, y los dispositivos IoT, experimentan alrededor de 5,200 ataques mensualmente (Symantec, 2019; Wang et al., 2017). Uno de los malware más utilizados para atacar a dispositivos IoT, es el malware Mirai, es un botnet que infectas estos equipos, especialmente, routers y cámaras IP, para posteriormente, utilizar estos dispositivos para un ataque DDoS, por esta razón, el Mirai es

considerado el malware de IoT con capacidad DDoS más disruptivo visto hasta ahora (De Donno et al., 2018). Darlloz and BASHLITE son otros ejemplos de IoT malware.

3.2.2 DETECCIÓN

Como resultado del desarrollo de malware en la innovación, la información de la protección de malware oscuro es un tema fundamental en el reconocimiento de malware. Los enfoques de detección de malware se dividen en dos categorías principales que incluyen métodos basados en el comportamiento y basados en firmas. Además, hay dos análisis de malware estáticos y dinámicos que generalmente se realizan para encontrar aplicaciones maliciosas.

3.2.2.1 Signature-Based (Basado en Firmas)

Recientemente, la detección basada en firmas es el procedimiento más utilizado en la programación antivirus y destaca la correlación exacta. El reconocimiento de malware se ha centrado esencialmente en realizar investigaciones estáticas para revisar la marca de la estructura del código de las infecciones, en lugar de métodos de comportamiento de elementos. El sistema basado en firmas encuentra interrupciones utilizando una lista predefinida de asaltos conocidos. A pesar de que esta disposición tiene la capacidad de identificar malware en la aplicación versátil, requiere una revisión constante de la base de datos de firmas predefinida. Además, es menos eficaz en la identificación de ejercicios nocivos utilizando la técnica basada en firmas debido a la naturaleza cambiante del malware portátil. Las estrategias basadas en firmas dependen de ejemplos excepcionales de bytes crudos o articulaciones estándar, conocidas como marcas, creadas para coordinar el documento nocivo. Por ejemplo, los aspectos destacados estáticos de un registro se utilizan para decidir si es un malware. La principal ventaja de las técnicas basadas en firmas es su minuciosidad, ya que siguen todas las formas de ejecución concebibles de un documento determinado. (Souri & Hosseini, 2018)

Dentro de la estructura del malware, los objetos maliciosos existentes tienen características que pueden usarse para generar una firma digital única. El proveedor de anti-malware utiliza los algoritmos metaheurísticos que pueden escanear de manera eficiente el objeto malicioso para controlar su firma. Después de identificar el objeto malicioso, la firma detectada se agrega a la base de datos existente como el malware reconocido. Las fuentes de la base de datos incluyen

una gran cantidad de las diversas firmas que clasifican los objetos maliciosos. En la detección de malware basada en firmas, hay varias cualidades, incluida la identificación rápida, fácil de ejecutar y ampliamente accesible.(Feizollah et al., 2015)

Dado que los planes de firma digital se obtienen a partir de malware conocido, estos planes también son conocidos en general. Posteriormente, los programadores pueden eludirlos eficazmente mediante procedimientos sencillos de confusión. Por lo tanto, se puede modificar el código de malware y evitar la identificación basada en firmas. Dado que los proveedores de antimalware se basan en la premisa del malware conocido, no pueden distinguir el malware oscuro o incluso las variaciones de malware conocido. De esta forma, sin una firma digital exacta, no pueden distinguir adecuadamente el malware polimórfico. En este sentido, el reconocimiento basado en firmas no ofrece un seguro de día cero. Además, dado que un indicador basado en firmas utiliza una firma aislada para cada variación de malware, la base de datos de firmas se desarrolla a un ritmo exponencial. (Feizollah et al., 2015)

3.2.2.2 Behavior-Based

Esta subsección ilustra los enfoques basados en el comportamiento en la detección de malware.. Las metodologías basadas en el comportamiento requieren la ejecución de un ejemplo dado en una situación de espacio aislado y los ejercicios en tiempo de ejecución se verifican y registran. Los sistemas de investigación dinámica utilizan tanto la virtualización como las condiciones de imitación para ejecutar un malware y eliminar sus prácticas. La principal ventaja del enfoque basado en el comportamiento es que proporciona una comprensión superior de cómo se produce e implementa el malware. (Souri & Hosseini, 2018)

Behavior-based utiliza un análisis dinámico para le detección de un código malicioso. Por esto, en los procesos de detección de malware en donde se utiliza el behavior based approach, es necesario ejecutar el código entro de un sandbox, en donde los ejercicios en tiempo de ejecución se verifican y registran (Tang & Qian, 2019). Esto permite que tenga más eficiencia al momento de detectar la intención real del código que está siendo analizando, ya que logra contrarrestar la evasión porque observa que es lo que realiza el código en vez de determinar como este se ve (Chanana et al., 2017).

En base a esto, signature-based approach, permite detectar variantes de malware que no se tenía conocimiento con anterioridad, siendo esta, uno de sus principales ventajas, además, de que permite la detección de malwares polimórficos. A causa de la constante evolución de los malwares, estos poseen una porción de código que logran detectar si se encuentran en un ambiente controlado. Convirtiendo a este método incapaz de tratar con estos tipos de malware, ya que cuento estos detectan que se encuentran en un ambiente controlado, estos limitan sus acciones. En consecuencia, estos procedimientos de análisis de malware se están quedando atrás, por lo tanto, es necesario reajustar estos métodos para logren adaptarse a los ataques de malware y lograr detectarlo, de lo contrario, los sistemas de detección perderán su rendimiento. (Watson et al., 2016)

3.3 REDES NEURONALES

Las redes neuronales artificiales son consideras para de los sistemas de aprendizaje automático, el cual es una rama de la inteligencia artificial. Esta es considerada una herramienta que ayuda a convertir los datos en conocimiento, esto a causa que actualmente la humanidad produce y obtiene información del mundo que le rodea por millones de bites al día, pero en si estos datos no generan conocimiento si no son analizados. (Raschka, 2015)

Por lo tanto, se utilizará esta herramienta para obtener el máximo beneficio de ella, para realizar el análisis respectivo de los datos obtenidos del ambiente. Todo esto sucde, gracias que las redes neuronales poseen las habilidades de realizar un aprendizaje automático que le permite aprender de su entorno y de los datos que se les provee.

Según Ponce Cruz (2010) las redes neuronales artificiales son como aproximadores no lineales a la forma en que funciona el cerebro. Por lo tanto, no deben compararse directamente con el cerebro ni confundir los principios que fundamentan el funcionamiento de las redes neuronales artificiales y el cerebro, ni pensar que las redes neuronales se basan únicamente en las redes biológicas ya que solo emulan en una parte muy simple el funcionamiento del cerebro humano. Además, se debe tener en cuenta que las redes biológicas son generadoras de procesos neurobiológicos en que se establecen relaciones de complejidad muy alta, las cuales no se puede lograr con redes monocapa ni con redes multicapas.

3.3.1 Fundamentos Biológicos

La neurona, como toda célula, consta de una membrana exterior M, que la limita y le sirve de órgano de intercambio con el medio exterior, de un citoplasma C, que es el cuerpo principal de la célula donde radica el grueso de sus funciones y de un núcleo N, que contiene el material genético de la célula, como lo muestra la ilustración 2. (Ramon & Cajal, 1911)

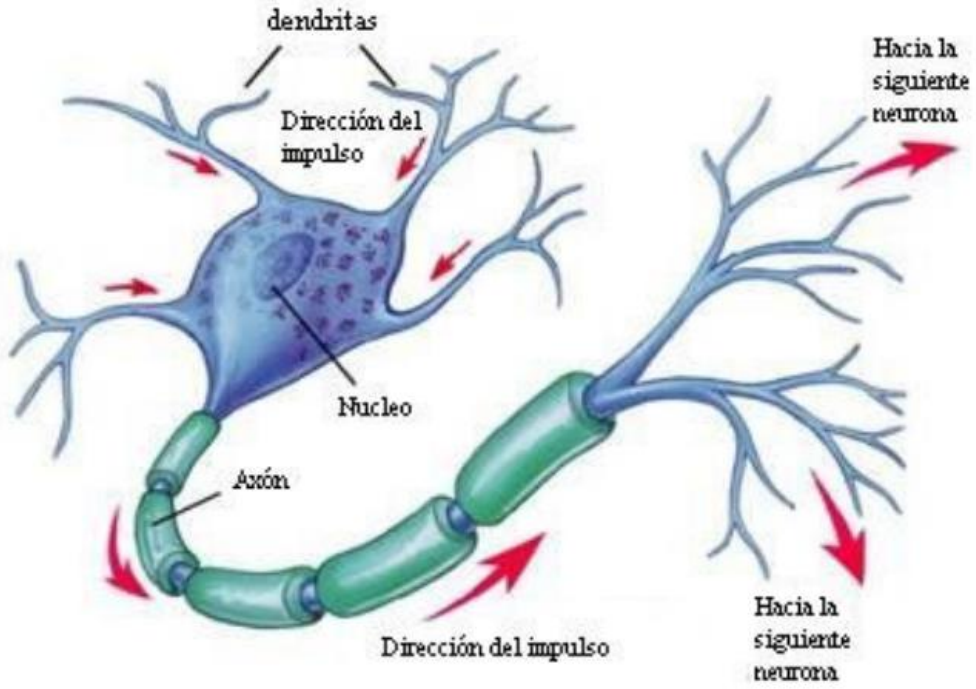

Ilustración 2-Esquema de una neurona biológica

Fuente: (Ramon & Cajal, 1911)

El citoplasma presenta unos alargamientos D, llamados dendritas, que son órganos de recepción. En las dendritas termina un gran número de fibras F que son conductores que llevan la señal o impulso nervioso de los receptores o de otras neuronas hacia la neurona. Estas fibras terminan en un pequeño corpúsculo llamado sinapsis, que constituye un relevador bioquímico y que sirve para transferir la señal de una neurona a otra. Existen dos clases de sinapsis: actuadoras, que favorecen el disparo de la neurona receptora e inhibidora, que dificultan éste. Cuando se presenta un cierto desbalance entre las sinapsis actuadoras y las inhibidoras activas, la neurona dispara un impulso de salida, que constituye la respuesta de la neurona. Este impulso nervioso de salida es conducido

por una prolongación cilíndrica alargada (hasta de varios decímetros de largo) de la neurona, que se llama cilindro eje o axón A, que en su extremo se divide en varias fibras para comunicarse con otras neuronas o con órganos efectores o motores como glándulas o músculos. (Ramon & Cajal, 1911)

Una de las características de las neuronas es su capacidad de comunicarse. En forma concreta, las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite señales de salida. El axón transmite dichas señales a las terminales axónicos, los cuales distribuyen la información. Se calcula que en el cerebro humano existen aproximadamente 1015 conexiones. Las señales que se utilizan son de dos tipos: eléctricas y químicas. La señal generada por la neurona y transportada a lo largo del axón es un impulso eléctrico, mientras que la señal que se transmite entre los terminales axónicos de una neurona y las dendritas de la otra es de origen químico. Para establecer una similitud directa entre la actividad sináptica y la analogía con las redes neuronales artificiales podemos considerar que las señales que llegan a la sinapsis son las entradas a la neurona; éstas son ponderadas (atenuadas o simplificadas) a través de un parámetro denominado peso, asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo). El efecto es la suma de las entradas ponderadas. Si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da salida). Esta es una de todo o nada; cada neurona se activa o no se activa. (Ponce Cruz, 2010)

3.3.2 Red Neuronal Artificial

El investigador Rosenblatt (1962) desarrolló el perceptrón que fue la primera red neuronal artificial especificada con toda precisión y orientada computacionalmente. Como era una máquina que podía aprender y demostrar comportamiento adaptativo complejo, atrajo de inmediato la atención de los investigadores. Desechó el enfoque de teóricos anteriores, que veían al cerebro como una computadora lógica. En vez de ello, lo consideró como un asociador y clasificador, cuya misión era asociar respuestas de clasificación a estímulos específicos.

Los perceptrones se aplicaron rápidamente a resolver problemas tales como la predicción climatológica, la interpretación de electrocardiogramas y otros. Tal parecía que se había hallado

la clave para comprender el funcionamiento cerebral, emulando las Redes Neuronales naturales mediante redes complejas de perceptrones. Sin embargo, pronto se comprobó que las redes con una capa de perceptrones eran incapaces de resolver problemas tan simples como la simulación de una compuerta lógica de tipo O exclusivo y, tras una investigación sobre las limitaciones de los perceptrones, Minsky y Pappert publicaron el libro Perceptrons (Minsky & Pappert, 1969) donde se hacían patentes estas limitaciones. Por lo que, emergió un nuevo cuerpo teórico alrededor de las redes neuronales multicapas, que superó las limitaciones y dio un nuevo impulso al desarrollo de redes neuronales artificiales.

Consecuentemente, Ponce Cruz, (2010) define las redes neuronales artificiales como sistemas de mapeaos no lineales cuya estructura se basa en principios observados en los sistemas nerviosos de humanos y animales. Constan de un número grande de procesadores simples ligados por conexiones con pesos. Las unidades de procesamiento se denominan neuronas. Cada unidad recibe entradas de otros nodos y genera una salida simple escalar que depende de la información local disponible, guardada internamente o que llega a través de las conexiones con pesos. Pueden realizarse muchas funciones complejas dependiendo de las conexiones. La neurona artificial es una unidad procesadora con cuatro elementos funcionales que se presentan es la ilustración 3.

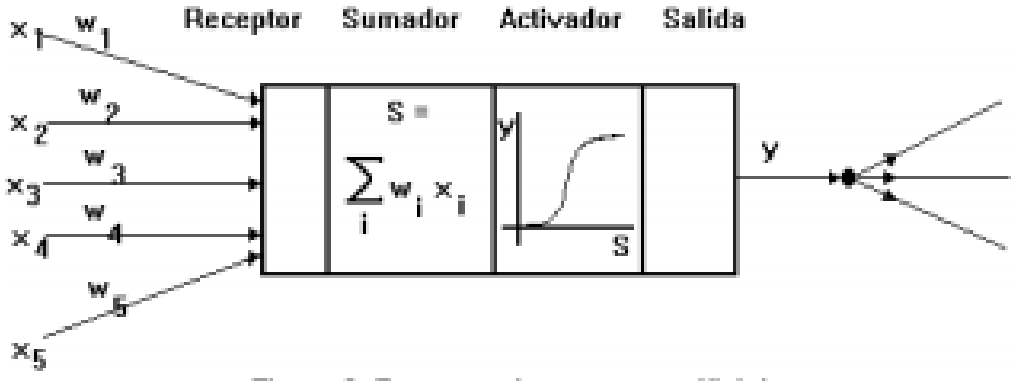

Ilustración 3-Esquema de una neurona artificial

Fuente: (Raschka, 2015)

El elemento receptor, a donde llegan una o varias señales de entrada xi, que generalmente provienen de otras neuronas y que son atenuadas o amplificadas cada una de ellas con arreglo a

un factor de peso wi que constituye la conectividad entre la neurona fuente de donde provienen y la neurona de destino en cuestión. El elemento sumador, que efectúa la suma algebraica ponderada de las señales de entrada, ponderándolas de acuerdo con su peso.

Donde wi son los pesos de las ligas de conexión y xi son señales de la salida de otros nodos o entradas externas. El elemento de función activadora, que aplica una función no lineal de umbral (que frecuentemente es una función escalón o una curva logística) a la salida del sumador para decidir si la neurona se activa, disparando una salida o no. El elemento de salida que es el que produce la señal, de acuerdo con el elemento anterior, que constituye la salida de la neurona. Este modelo neuronal es el utilizado en casi todas las Redes Neuronales artificiales, variando únicamente el tipo de función activadora.

3.3.3 Red Neuronal Convolucional (CNN)

Las CNN, son redes del aprendizaje profundo, que se utiliza comúnmente para la clasificación de imágenes, la cual es la red neuronal convolucional, sin embargo, también puede ser utilizadas para el análisis y clasificación de datos, que es lo que se busca en la presente investigación. Las redes neuronales convolucionales son consideradas un tipo de RNA en donde las neuronas se basan biológicamente sobre las neuronas que se encuentran en la corteza visual primaria del cerebro humano, además, es una variante de una RNA normal de perceptrón multicapa, ya que su procesamiento lo hace a través de matrices bidimensionales, siendo redes muy efectivas para la clasificación, también se debe porque cada parte de la red convolucional es entrenada para realizar una tarea en específico, en consecuencia, se reduce grandemente el número de capas ocultas, obteniendo una reducción sustancial del proceso de entrenamiento de la red neuronal convencional. (LeCun et al., 2015)

3.3.3.1 Arquitectura

LeCun *et al*. (2015) mencionan que la arquitectura general de una CNN es una red multicapa, la cual está compuesta de capas de convoluciones o capas convolucionales y de reducción alternadas y finalmente tiene capas de conexión total, al igual que una red perceptrón multicapa, tal como se muestra en la ilustración 4.

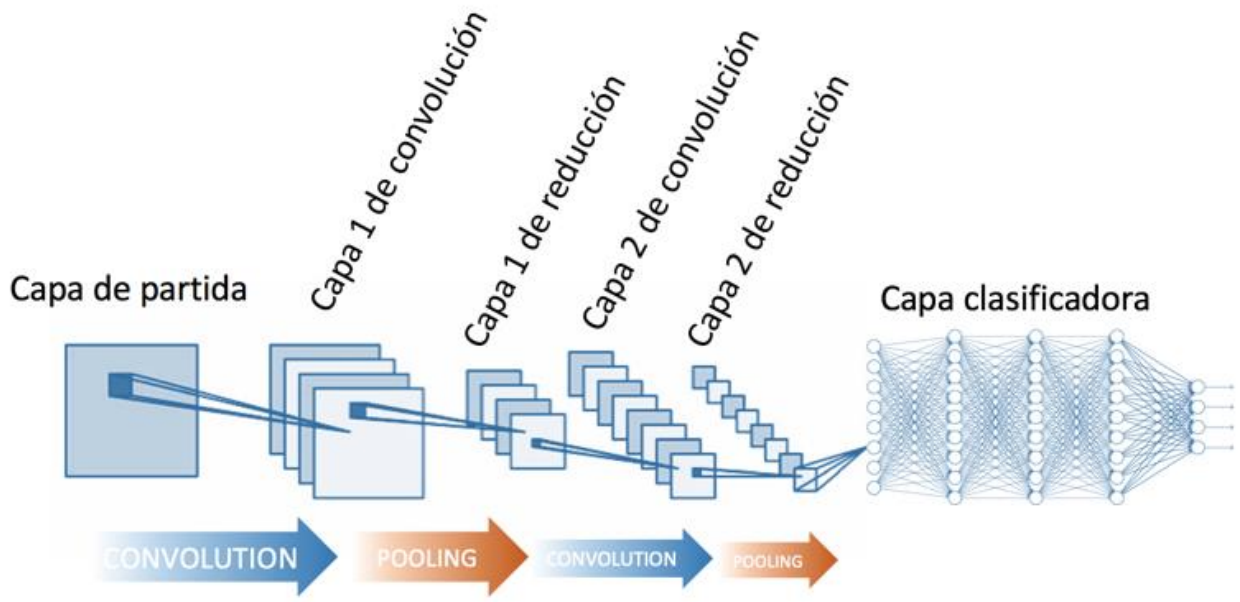

Ilustración 4-Arquitectura de una CNN

Fuente: (LeCun et al., 2015)

La CNN es una RNA utilizada comúnmente para clasificación, por lo tanto, como cualquier red con este tipo de aplicación, al inicio se componen por una fase de extracción de características de los datos entregados, la cual está compuesta por las neuronas o capas convolucionales, seguidamente existe una reducción por muestro y para finalizar, se encuentran neuronas de perceptrón para realizar la clasificación sobre las características extraídas, tal como se detalla gráficamente en la ilustración 36. (LeCun et al., 2015)

3.3.3.2 Proceso de Aprendizaje de una CNN

Ngiam *et al.* (2019) mencionan que el proceso de aprendizaje de una CNN es un poco diferente a la de una RNA normal de perceptrón multicapa, sin embargo, siempre sigue siendo ya sea aprendizaje supervisado. Como ya se mencionó, la CNN utiliza como datos de entrada matrices bidimensionales, por esta razón son muy utilizadas para el análisis de imágenes, ya que la información de una imagen está representada en una matriz de pixeles. Por lo tanto, el proceso de aprendizaje de una CNN se compone de diversos pasos que permiten la clasificación de datos, estos procesos son:

1) Matriz de Datos y Neuronas

Primeramente, se deben ingresar los datos en forma de matriz, cada datos dentro de la matriz de datos representa una neurona, es decir, si se tiene una matriz de datos de 28x28, entonces se utilizan 784 neuronas, en el caso de una imagen, si es una imagen de una resolución de 28x28 y solamente es de 1 color de escala de grises son 784, sin embargo, si la imagen es a color, se utilizarían 3 canales RBG, por lo tanto, serian un total 2352 neuronas, 784 para cada canal. Estas neuronas representan la capa de entrada de la CNN.

2) Pre-Procesamiento

Antes de entregar los datos de entrada a la red, se deben convertir los valores, en el caso de una imagen, cada dato dentro de la matriz representa un valor entre 0 a 255, por lo tanto, estos valores se deben normalizar a los valores entre 0 y 1, para que sean entregados a la red. Como se muestra en la ilustración 5, se muestra este preprocesamiento, para el caso de una imagen a color que está compuesta por tres canales de RGB, es una imagen de 6x6 pixeles.

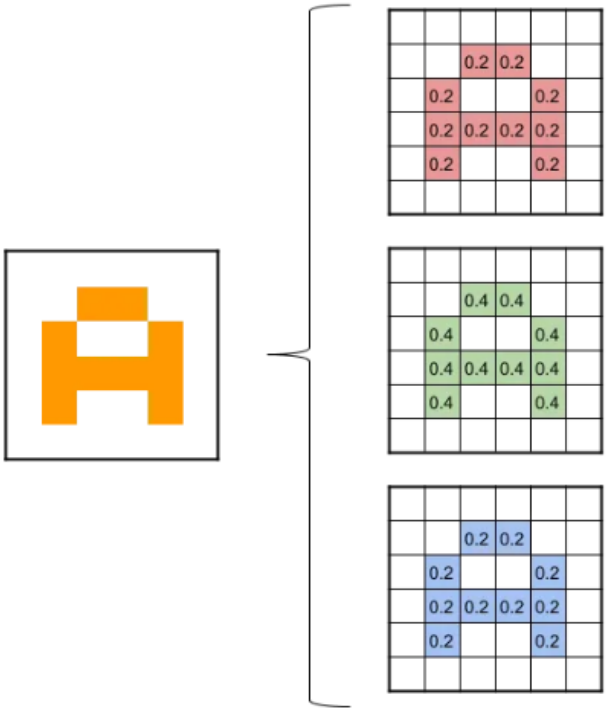

Ilustración 5-Pre-procesamiento en la CNN

Fuente: (Ngiam et al., 2019)

3) Convolución

Luego, de que los datos son alimentados a la red, se procede al proceso de convolución realizado por las capas de convolución, donde su función consiste en realizar un proceso distintivo de los datos. Para realizar esto, en esta capa se ejecutan la operación matemática del producto escalar contra una pequeña matriz que se le denomina *kernel*, ver apartado B de la ilustración 36, que posee un menor tamaño a la matriz de entrada, y permite visualizar todas las neuronas de entrada, generando una nueva matriz de salida, la cual será la siguiente capa de neuronas ocultas. Generalmente, no se aplica solamente 1 *kernel*, por el contrario, se aplican muchos *kernels*, de esta manera se tienen los filtros que consisten en un conjunto de *kernels*. Inicialmente, el *kernel* toma valores aleatorios.

Por lo tanto, si para la primera convolución se tienen 32 filtros, se tendrá como resultado 32 matrices de salida, a estas matrices de salida se le conoce como *feature mapping* o mapas de características, ver apartado C de la ilustración 36. Estas matrices de salidas representan ciertas características de la imagen original. En el caso de que la matriz de datos de entrada sea 28x28, luego de que pase por la primera convolución, la cantidad de neuronas sería 28x28x32 siendo un total de 25,088 neuronas para la primera capa oculta de neuronas.

Seguidamente, de este proceso, se les aplica a las matrices de salidas o mapas de características una función de activación. Generalmente, la función de activación más utilizada para este tipo de redes neuronales es la función de activación ReLU, que solamente permite pasar los valores positivos, como se muestra en el apartado D, de la ilustración 6.

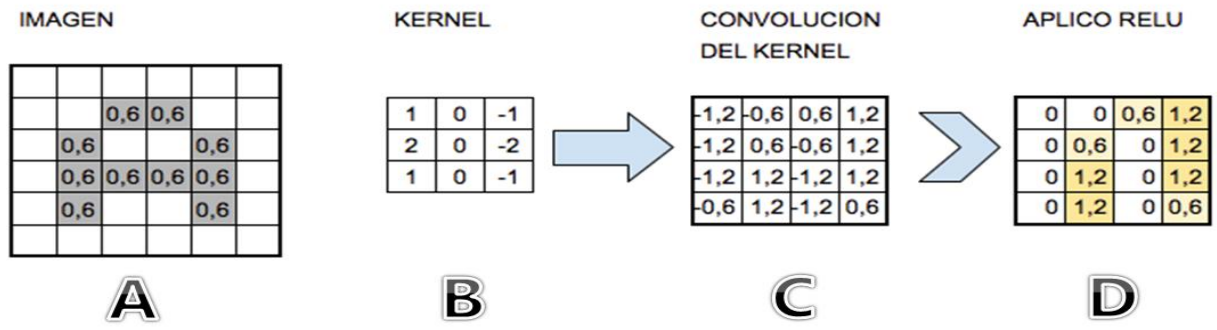

Ilustración 6-Proceso de convolución

Fuente: (Ngiam et al., 2019)

4) Reducción o Muestreo

Luego de obtener los mapas característicos, se realiza el proceso de reducción, muestreo o también conocido como *pooling*. Este paso consiste en disminuir la cantidad de parámetros al quedarse con las características más comunes. Es proceso es de vital importancia, debido a lo siguiente, en el caso de que la entrada sea una matriz de 28x28, en la primera capa oculta o convolución se apliquen 23 filtros se tendría 28x28x32 neuronas, luego, si se pasa directamente a la siguiente convolución 28x28x32x32 neuronas en caso de que igualmente posea 32 filtros, generando una gran cantidad de neuronas y para lograr procesar esto se requeriría un gran poder computacional de procesamiento, por esta razón es muy importante reducir el tamaño de la próxima capa de neuronas, haciendo un muestreo y conservando las características más importante de cada mapa característico.

Existen diferentes tipos de muestreo, sin embargo, el más utilizado es el *Max-Pooling*. Este proceso consiste en reducir los parámetros mediante la extracción del valor máximo de una región fija del mapa característico, al reducir características el método pierde precisión, sin embargo, mejora significativamente su compatibilidad, en la ilustración 7, se detalla con matrices este proceso.

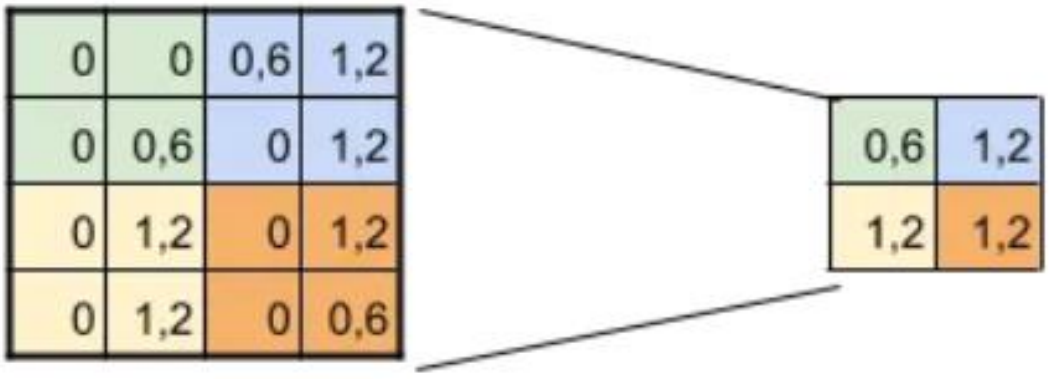

Ilustración 7-Muestreo con max-pooling

Fuente: (Ngiam et al., 2019)

El proceso de convolución y de muestreo se repite en las convoluciones subsecuentes que depende de la cantidad de convoluciones asignada por el diseñador de la red neuronal.

5) Clasificación

Finalmente, se procede con la respectiva clasificación de la información de entrada con respecto a una salida en específico. Para esto se utiliza una red neuronal tradicional, una red multicapa, el número de neuronas en la capa de entrada de esta red, depende de la última convolución junto con su respectivo muestreo, por ejemplo, el último muestreo termina con 3x3x28 (alto, ancho, mapas), entonces esa es la cantidad de neuronas para la capa de entrada de la red multicapa.

Esta capa de neuronas en la red multicapa, generalmente, se le aplica la función de activación SoftMax, la cual se encuentra conectada con la capa de salida final, que posee la cantidad de neuronas correspondientes con las clases que se están clasificando. En la ilustración 8, se muestran todo el procedimiento que se detalló.

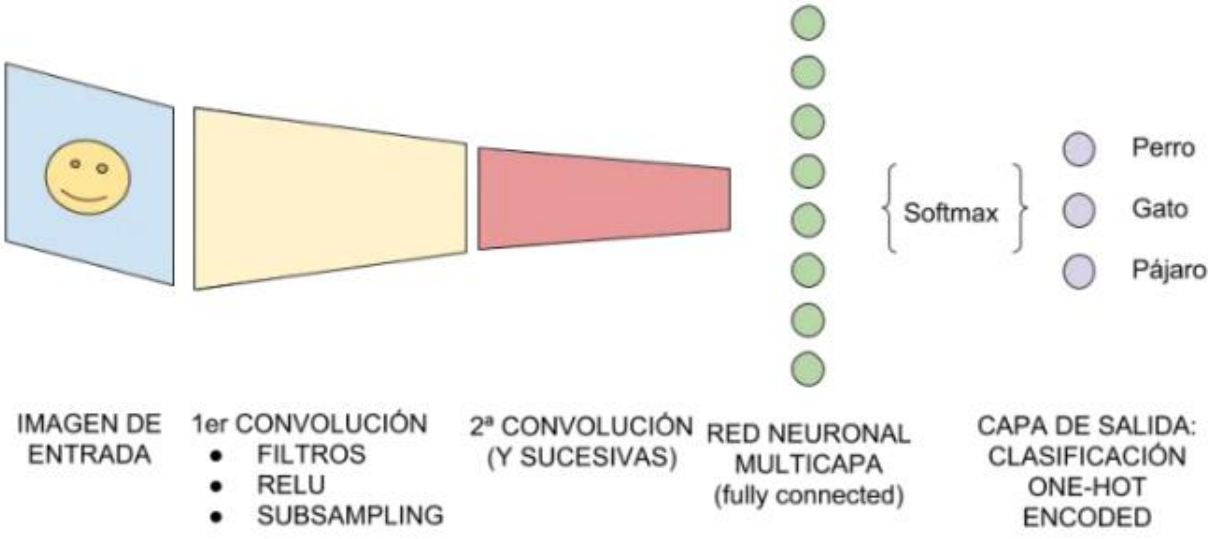

Ilustración 8-Proceso de aprendizaje de una CNN

Fuente: (Ngiam et al., 2019)

3.3.3.3 Backpropagation en CNN

El algoritmo la rectificación de pesos *backpropagation* también es utilizado en las CNN, el proceso es similar a las redes multicapa, en el caso de la multicapa, se ajusta hacia adelante y hacia atrás,

mejorando el valor de los pesos en cada neurona interconectada en las capas, a medida se repite el proceso, los pesos son ajustados hasta obtener los óptimos. (Schmidhuber, 2015)

Con respecto a las CNNs, el valor ajustado es el valor de los pesos de los distintos *kernels* dentro de cada convolución según la cantidad de filtros. Esto se convierte en una gran ventaja en el momento del proceso de aprendizaje, debido a que el tamaño de los *kernels* es reducido una matriz pequeña, por lo tanto, es mucho más eficiente este algoritmo que en una red multicapa, ya que aquí se ajusta el peso de cada neurona, y como se mencionó, en una CNN existe una gran cantidad de neurona. (Schmidhuber, 2015)

CAPÍTULO IV. METODOLOGÍA

La presente investigación científica es un proyecto aplicado en la industria de ciberseguridad, referente al área del análisis de códigos maliciosos. La metodología a seguir para el desarrollo de este proyecto de investigación jugó un papel muy importante. Ya que en esta, se detalló el seguimiento realizado a las diferentes variables que fueron analizadas a lo largo de toda la investigación. Permitiendo su manipulación con el fin de obtener los mejores resultados. Para esto se utilizaron diversas técnicas, instrumentos y materiales, además, se detalló este seguimiento a través de diferentes actividades que se presentaron en el cronograma de actividades, todo esto se presenta en este capítulo.

4.1 ENFOQUE

En el presente proyecto de investigación se analizaron diversas variables numéricas, encontradas en el análisis de los códigos con el uso de una red neuronal artificial. Por lo tanto, se establece que la presente investigación posee un enfoque cuantitaitvo. Hernández Sampieri, Fernández Collado, & Baptista Lucio (2010) mencionan que una investigación cuantitativa se da por aludida al ámbito del análisis numérico de las variables de investigación, se analiza una realidad objetiva con base en las mediciones numéricas y el análisis de estas, con la finalidad de determinar predicciones o patrones de comportamiento del problema planteado.

Debido a que las variables de investigación fueron manipuladas para emplear experimentos y realizar un análisis de causa-efecto para obtener la resolución del problema planteado y obtener el alcance estipulado, se determina que la presente investigación es de tipo experimental. Con esto se obtiene una investigación la cual conlleva un proceso secuencial y deductivo

4.2 VARIABLES DE INVESTIGACIÓN

Las variables de investigación juegan un papel muy importante en la investigación, ya que son manipuladas y observadas de manera experimental para posteriormente analizar su comportamiento y determinar los efectos de la manipulación. Las variables de investigación se clasificaron en variables dependientes e independientes. En la ilustración 9, se presentan las variables de estudio de la presente investigación.

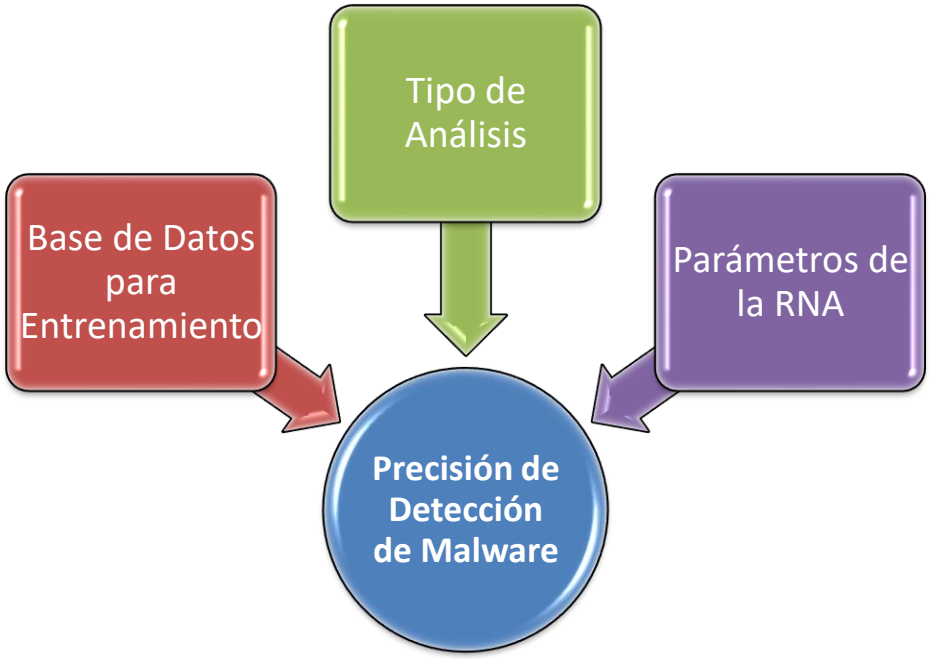

Ilustración 9-Variables de investigación

Fuente: Elaboración Propia

4.2.1 Variable Dependiente

La variable dependiente de la presente investigación fue la precisión de la red neuronal artificial para la detección acertada de un código malicioso. La precisión se refiere a la dispersión del conjunto de valores obtenidos con base en las mediciones repetidas de una magnitud, entre menor es la dispersión mayor es la precisión. Sin embargo, la precisión refleja la proximidad de distintas medidas entre sí, y es función exclusiva de los errores accidentales, además, es un parámetro relevante, especialmente en la investigación de fenómenos físicos.

4.2.2 Variables Independientes

Se establecieron un total de 4 variables independientes de investigación las cuales afectan directamente la precisión de detección de malware, la variable dependiente. Las 3 variables independientes de investigación se detallan a continuación.

4.2.2.1 Base de Datos para Entrenamiento

Una de las variables que es la magnitud de la base de datos para el entrenamiento de la red neuronal artificial. Esto se debe principalmente, a que la cantidad de información o la base de datos utilizada para el entrenamiento de la red neuronal afecta considerablemente sobre el buen resultado de la RNA. Ya que con mayor información en la base de datos para entrenamiento la RNA puede llegar a aprender más y diferentes patrones que le permita detectar un malware con una mayor precisión.

4.2.2.2 Tipo de Análisis

El tipo de análisis se refiere enfoque o el método de detección utilizado para el análisis de un código para determinar si este es o no malicioso. Como se detalló en el marco teórico, se pueden utilizar ya sea un de dos métodos disponibles. Un análisis estática, basado en firmas. O un análisis dinámico que está basado en un el comportamiento que presenta el código. Donde la mayor diferencia entre estos dos métodos, es que con un análisis estático, no es necesaria la ejecución del código. Por otro lado, un análisis dinámico si es necesario ejecutar el código para determinar su comportamiento.

4.2.2.3 Parámetros de la RNA

El tipo de red neuronal artificial a utilizar es una red neuronal convolucional. Para el diseño de una CNN es necesario definir ciertos parámetros para su funcionamiento. Por lo tanto, la manipulación de estos parámetros afecta sobre la precisión final de que poseerá la RNA para la detección de malware. Estos parámetros incluyen el número de entradas, el número de salidas, la cantidad de filtros, el número de convoluciones, entre otras.

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

Para poder ejecutar el proyecto se realizó un proceso investigativo para el análisis y clasificación de las imágenes, por medio de la adquisición de información confiable en libros, revista académicas y el conocimiento y experiencia de especialistas en el área.

Para el diseño de la red neuronal convolucional fue necesario el uso de diversos conjuntos de librerías. Principalmente, fue utilizado el lenguaje de programación Python para el desarrollo de

la RNA. Además, fueron utilizadas librerías CUDNN7 y NCCL 2 para el desarrollo de la red neuronal convolución, ya que permite el desarrollo de RNAs basadas en el aprendizaje profundo.

4.4 Metodología de Estudio

En la presente sección se lleva a cabo un recuento de todas las formas de análisis aplicadas para la elaboración del presente proyecto; aplicando todos los conocimientos, herramientas y métodos en la detección de un código malicioso, para ello se desarrolló la investigación en un proceso esquematizado y secuencial de un total de 4 etapas, las cuales se muestran en la ilustración 10.

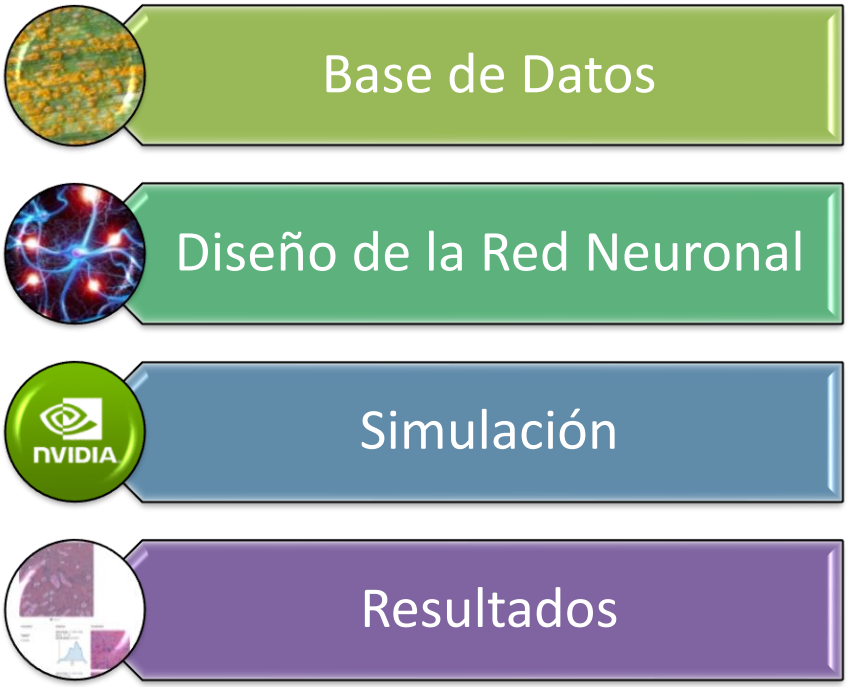

Ilustración 10-Metodología de estudio

Fuente: Elaboración Propia

En la ilustración 10, se muestran los 4 etapas realizadas para la resolución de la problemática establecida y el alcance investigativo. Se inició con la etapa de la recolección de datos, durante esta etapa se obtuvo información de todos los medios posibles. Se obtuvo una base de datos de diversos códigos que permitieron el enteramiento de la RNA. De igual manera, en esta etapa fue seccionada la base de datas entre códigos para el entrenamiento, verificación y pruebas de la RNA.

Posteriormente, en la segunda etapa de la metodología de estudio fue diseñada la RNA. Se procedió al diseño y entrenamiento de la RNA para la detección de un código malicioso. Se realizó un entrenamiento supervisado, en donde se le entregó la información a la RNA con su respectiva etiqueta del código si este es maliciosos o no. Con la finalidad de que la RNA detecte diferentes patrones que se presentan los códigos que si son maliciosos y de igual manera, reconocer patrones que presentan los códigos que no son maliciosos. Aquí fueron seleccionados los parámetros de la RNA.

Después, la RNA fue puesta a prueba con el conjunto de la base de datos de verificación y de prueba. Todo esto para determinar la precisión de la RNA después del entrenamiento supervisado ejecutado. Finalmente, se presenta todos los resultados obtenidos del proyecto de investigación.

4.5 Cronograma de Actividades

El presente proyecto de investigación se realizó de manera secuencial, siguiendo diferentes actividades para la culminación de este, estas actividades se presentan en la tabla 6.

Se inició con plantear el problema a investigar, los objetivos y alcance del proyecto, además, de los precedentes respecto a la misma temática de investigación. Luego, se recolectó toda la información y conceptos que da sustento a la investigación. Posteriormente, dio inició a la recolección de datos para la construcción de la base de datos para el entrenamiento de la RNA. Seguidamente, se diseñó la metodología utilizada para el proceso investigativo. Una vez, con la base de datos para el entrenamiento de la red neuronal, se diseñó y creó la RNA y se entrenó de manera supervisada. Ya finalizado el entrenamiento, se simuló y se puso a prueba la red neuronal para la detección de malware.

Tabla 1-Cronograma de actividades

Actividades Desarrolladas	**Semana**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

Plantear el problema de investigación										
Recopilación de la teoría de sustento										
Recolección de datos										
Presentación de la metodología										
Diseño de la red neuronal										
Simulación de la red neuronal										
Análisis de los resultados obtenidos										

Fuente: Elaboración Propia

CAPÍTULO V. RESULTADOS Y ANÁLISIS

En el presente capítulo, se detallan tanto los resultados obtenidos como su respectivo análisis, demostrando el origen y efecto de los resultados, se muestra el diseño de la red neuronal artificial que permitió la clasificación de malware.

5.1 CREACIÓN DE BASE DE DATOS

Para el entrenamiento de la RNA fue necesario crear una base de datos. Para ello fueron seleccionadas familias específicas de malware. Estas han sido de las más comunes en los últimos ataques cibernéticos, estos se describen a continuación:

1) Cerber

Cerber se vio por primera vez a mediados de 2015. Cerber se clasifica como Ransomware-as-aService (RaaS), donde el desarrollador alquilará la funcionalidad de ransomware con beneficios divididos entre los compradores y los vendedores. Entre muchos otros, es un tipo de ransomware que muestra un mensaje de rescate después de que ha infectado la computadora del usuario. Cifra archivos y agrega una extensión ".cerber", lo que significa que determinados archivos se han cifrado. Además, como cualquier otro malware de alto perfil, tiene una forma de "fábrica de malware"; lo que significa que utiliza una técnica de ofuscación de código que puede generar automáticamente grandes volúmenes de variantes de malware de hash único a partir del código de malware original.

2) Cryptowall

CryptoWall es una familia de ransomware que está diseñada para utilizar un algoritmo de cifrado sofisticado (RSA-2048) para hacer que los archivos sean inaccesibles en las computadoras de destino. Los investigadores de malware detectaron la primera versión de ransomware en 2013. Desde entonces, el virus criptográfico se actualizó varias veces y se agregaron funcionalidades adicionales (como el modo sigiloso y la eliminación de instantáneas de volumen) para evitar detecciones continuas de antivirus.

3) GandCrab

El ransomware GandCrab se descubrió como Ransomware-as-a-Service (RaaS) a finales de enero de 2018 y pronto se convirtió en el ransomware más popular y ampliamente utilizado del año. GandCrab nació potencialmente de la necesidad de monetizar aún más los datos cifrados de las organizaciones mediante la personalización de las notas de rescate en función del perfil de la víctima y el tipo de datos cifrados. Como resultado, la demanda por el rescate de GandCrab podría estar entre $ 600 y $ 700,000 por víctima. Este cambio de comportamiento probablemente ha llevado a un aumento significativo en los ingresos de los ciber delincuentes, especialmente desde que comenzaron a ofrecerlo como servicio.

4) Petya

Petya es una familia de cifrado de ransomware que se descubrió por primera vez en marzo de 2016. El malware se dirige a los sistemas Microsoft Windows, que infecta el registro de arranque maestro para ejecutar una carga útil que cifra la tabla del sistema de archivos en un disco duro y evita que Windows arranque. Petya ha evolucionado de varias formas. Un ejemplo de este tipo se puede ver en junio de 2017, donde una nueva variante de Petya (apodada NotPetya por los investigadores) se utilizó en un ciberataque a escala global y podría propagarse a través de la técnica del gusano informático.

5) Sality

Sality es una familia de malware que infecta archivos en sistemas Microsoft Windows y debutó por primera vez en 2003. Ha pasado por varias evoluciones y avances a lo largo de los años para convertirse en una forma dinámica, duradera y con todas las funciones de código malicioso. Debido a su continuo desarrollo y capacidades, Sality es una de las formas más complejas y formidables de malware hasta la fecha.

6) Wannacrypt

WannaCrypt es un gusano informático ransomware que se dirige a la familia de sistemas operativos Microsoft Windows. Se descubrió por primera vez en mayo de 2017, apuntando al sistema Microsoft Windows y ha afectado a más de 230,000 computadoras en 48 horas en más

de 150 países. El ransomware exigió un pago para desbloquear el sistema infectado. WannaCrypt provocó el desvío de ambulancias, el cierre de servicios que no eran de emergencia, la captura de máquinas en Telefónica en España y las aerolíneas y el ministerio afectados.

5.1.1 Codificación de Base de Datos a Bits

La base da datos de cada malware es una código. Sin embargo, la entrada de la RNA debe ser en bits. Por lo tanto, es necesario transformar el conjunto de datos ejecutables de malware en un formato de cadena de bits que sea adecuado para la entrada de aprendizaje automático. En la ilustración 11 se presenta el proceso a seguir para transformar estos archivos ejecutables a cadena de bits.

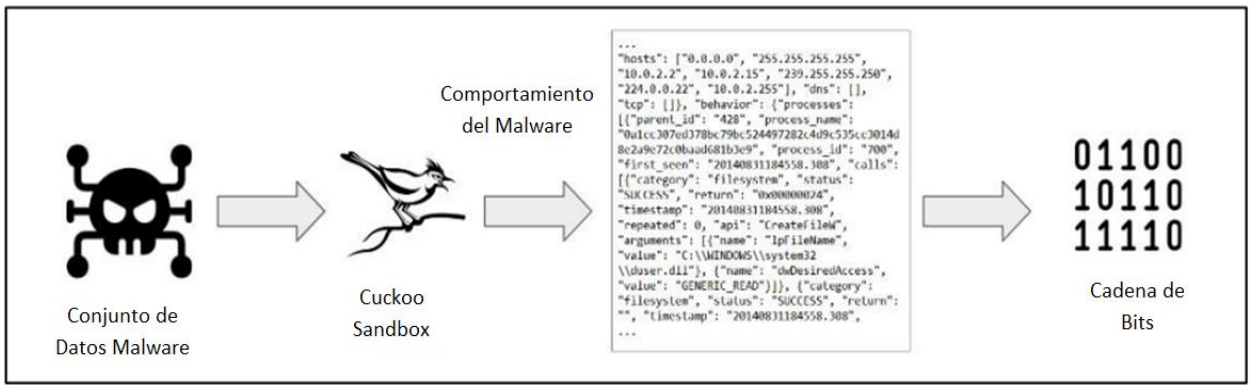

Ilustración 11-Codificación de datos malware a cadena de bit

Fuente: Elaboración Propia

5.1.1.1 Registro de datos de comportamiento de malware

Para obtener los comportamientos del malware en forma de datos de texto, se ha elegido un entorno de pruebas informático para articular esta tarea. Una caja de arena o sandbox es un entorno controlado por computadora en el que puede monitorear el comportamiento del programa a través de varios mecanismos de registro, como el rastreo de API, el registrador de llamadas del sistema y el registrador de tráfico de red de intermediarios. Entre todos los sandbox que se han lanzado a Internet, Cuckoo Sandbox es el que ha demostrado tener más potencial, ya que se actualiza constantemente con nuevas funciones a lo largo del tiempo. Además, Cuckoo Sandbox podría monitorear los comportamientos ejecutables hasta el nivel del kernel, donde es

el punto más bajo de todos los sistemas operativos. Al usar esta función, ayuda a la caja de arena de la computadora a monitorear cualquier comportamiento ejecutable incluso si el malware intenta esconderse inyectando en cualquier módulo del kernel. Para lograr estos pies, Cuckoo Sandbox se ha configurado junto con la ayuda de un software supervisor alojado en x86. Se ha elegido Oracle VirtualBox porque es de código abierto y cuenta con un gran apoyo de los propios desarrolladores de Cuckoo Sandbox. Para utilizar completamente los recursos del host para un análisis más eficiente y que consume menos, se ha configurado un grupo de tres entornos sandbox. Estos tres entornos sandbox tienen las especificaciones de hardware que son Windows 7 x64 bit y 2 GB de acceso a la memoria. Un sistema operativo en el que se ha elegido Windows 7 ya que este sistema operativo es popular entre los analistas de malware y puede ejecutar la mayor parte del malware moderno sin muchos problemas, ya que la versión actual de Windows incluye muchas protecciones para frustrar las infecciones de malware.

Para permitir que Cuckoo Sandbox use estos tres entornos sandbox, se han escrito más archivos de configuración para señalar qué máquinas virtuales debe usar al procesar y analizar malware. Utilizando el programa de interfaz de línea de comandos `cuckoo`, todo el malware se ha enviado al sistema de Cuckoo Sandbox, en el que utiliza una técnica de cola para procesar el malware que se ha enviado primero al sistema, como se presenta en la ilustración 12.

Mediante el uso de estas técnicas de agrupación en clústeres y sandboxing, Cuckoo Sandbox pudo analizar 70 ejecutables en un período de una hora, acelerando el tiempo necesario para analizar el malware y sus comportamientos. Por cada ejecutable que haya sido analizado por Cuckoo Sandbox, producirá un archivo de informe que contiene los comportamientos ejecutables mientras se ejecuta dentro del entorno limitado de la computadora. El informe producido tiene el formato de notación de objetos JavaScript (JSON), en el que este formato se puede analizar fácilmente con un analizador JSON común.

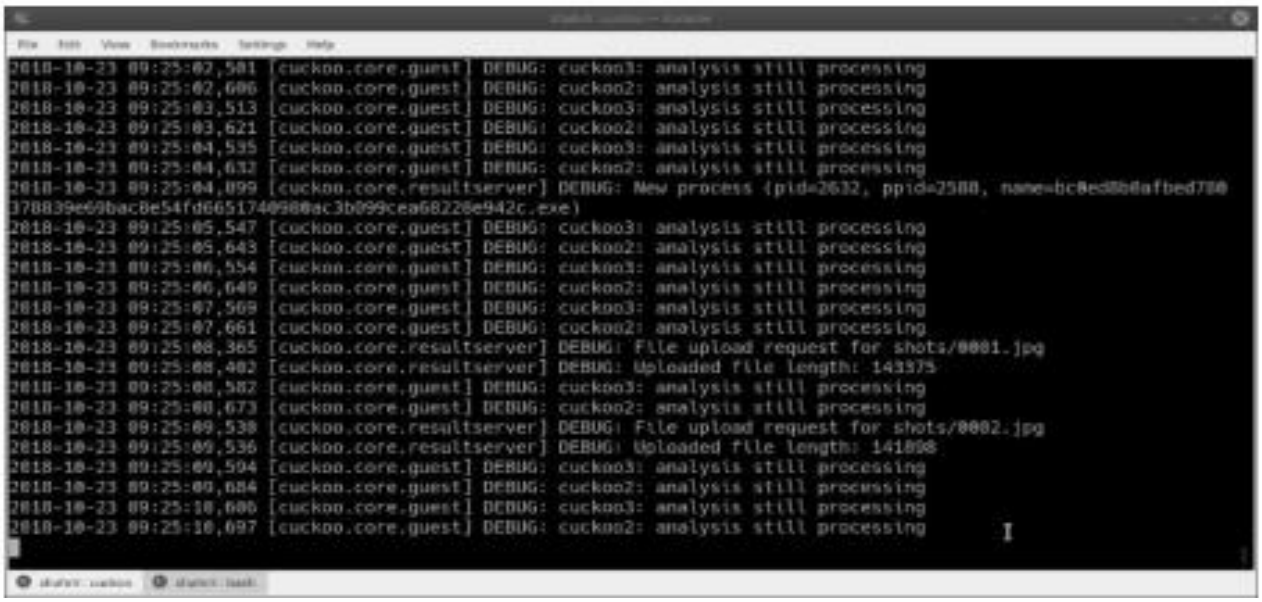

Ilustración 12-Cuckoo Sandbox analizando el ejecutable

Fuente: Elaboración Propia

Sin embargo, como el archivo de informe de comportamientos registrados está en formato JSON, todavía hay otro paso de preprocesamiento que debe realizarse. El problema con este archivo de informe es que para cada comportamiento registrado de malware, la salida del archivo de informe variará, dependiendo de la cantidad de comportamientos que escupe el malware. Esta representación de datos no se puede introducir en los algoritmos de aprendizaje automático, ya que la mayoría esperaba que el tamaño de la entrada fuera fijo. Un ejemplo de ello es la red neuronal; requiere que la entrada sea un vector de tamaño fijo o no puede continuar porque el número diferente de neuronas en la capa de entrada no coincide con el tamaño del vector de entrada.

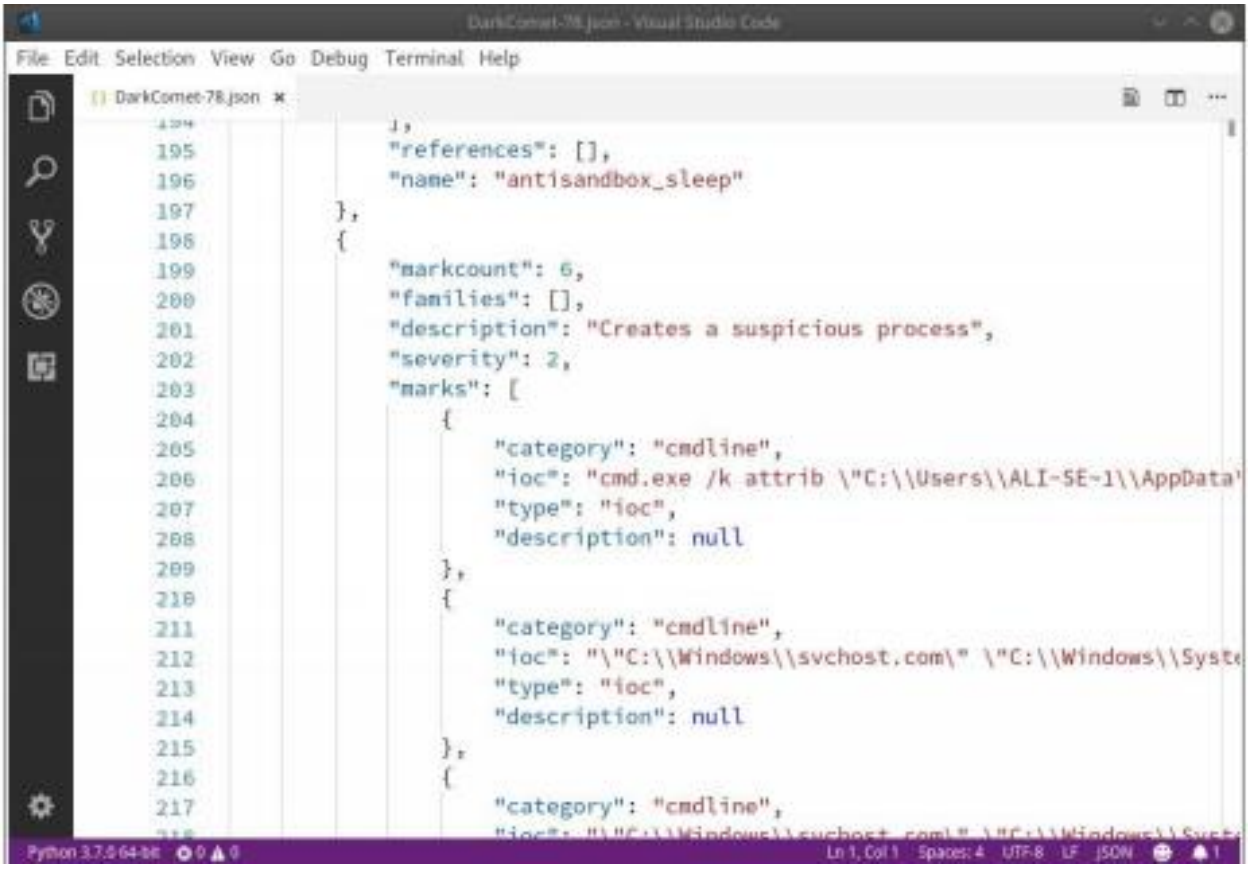

Ilustración 13- Comportamientos ejecutables en tiempo de ejecución producidos por Cuckoo Sandbox

Fuente: Elaboración Propia

5.1.1.2 Uso de n-Gramas

La representación de datos de malware sin procesar no se puede introducir en los algoritmos de aprendizaje automático tal como están. Un comportamiento de datos de malware sin procesar contiene mucha información basada en texto que no es adecuada para el proceso de capacitación de Machine Learning. Por lo tanto, se necesita un método para transformar un archivo de informe de comportamiento de tamaño variable en conjuntos de cadenas binarias de tamaño fijo. Para procesar los conjuntos de datos, se utiliza un método de procesamiento del lenguaje natural (NLP) para completar esta tarea. Uno de estos métodos es el uso de extracción de n-gramas. La forma más simple de extracción de n-gramo es la extracción de 1 gramo (o denominada unigramo). 1 gramo es un algoritmo que encuentra las palabras más frecuentes en la muestra de texto. A

continuación se muestran los pasos que se siguen para el proceso de transformación de datos del unigram. En la ilustración 14, se presente un ejemplo de este proceso.

1) Paso 1: Para todos los archivos de informe, cuente todas las apariciones de palabras unigrama.

Trate todo el archivo JSON informado como un archivo de texto normal. Divida el texto en unigramos haciendo tokenización de texto, como dividir por espacios y eliminar palabras cuyo tamaño sea menor o igual a tres. Construya un diccionario que por cada unigrama cuente sus ocurrencias.

2) Paso 2, ordena todas las palabras por su frecuencia.

Ordene el diccionario con respecto a sus ocurrencias, utilizando la función de clasificación de Python en la que utiliza el algoritmo de clasificación de Tim para acelerar la operación de clasificación.

3) Paso 3, elija las 10.000 palabras en unigramo más importantes.

Divida el diccionario y elija los 10,000 unigramas más frecuentes. Como primero se ha ordenado el diccionario, esta operación se puede realizar rápidamente.

4) Paso 4: Asigne los unigramas de cada ejecutable con los 10.000 unigramos superiores.

Al tener los 10.000 unigramos superiores, se puede crear un vector de cadena de bits mapeando cada unigramo dentro de cada ejecutable y marcarlo con 1 si el unigramo existe o 0 si no.

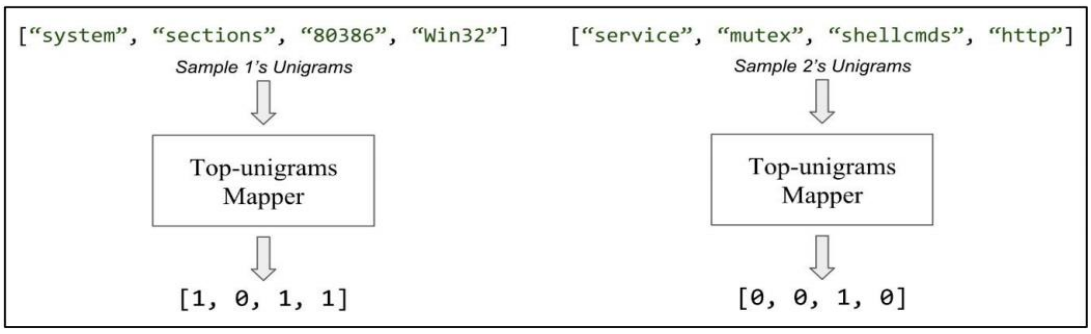

Ilustración 14-Conversión muestras unigramas a cadena de bits

Fuente: Elaboración Propia

Todos los procesos enumerados anteriormente se han realizado con la ayuda de los lenguajes de programación Python. La secuencia de comandos de Python analizará todos los informes, convertirá en token cada texto y encontrará unigrama de las 10,000 ocurrencias principales. Desarrollar la secuencia de comandos es un desafío técnico, ya que con cada informe con un tamaño promedio de más de 50 megabytes, se requiere mucho tiempo para procesar las 6.000 muestras ejecutables. Se han aplicado muchas optimizaciones, como el uso de la estructura de datos del diccionario de Python, que tiene una gran complejidad algorítmica para acelerar el proceso de recuento de ocurrencias.

Una vez que el script ha terminado de procesarse, para cada muestra ejecutable generará una representación de cadena de bits de unos y ceros como sus datos. Estos unos y ceros se tratan como características del ejecutable y tendrán un tamaño fijo de 10,000 de longitud. Al tener esta representación de la entrada, los algoritmos de aprendizaje automático, como la red neuronal, no tendrán problemas para manejarla con fines de aprendizaje.

5.1.1.3 Preprocesamiento de los Resultados

Como se explicó anteriormente, se han aplicado varios métodos a las muestras de malware para convertir la representación ejecutable del archivo en características que pueden ser entendidas por el algoritmo de aprendizaje automático.

El primer proceso logra capturar todos los datos de comportamiento de las muestras ejecutables de malware. Utilizando 6.000 muestras ejecutables, todas ellas se han enviado a Cuckoo Sandbox para su análisis dinámico. Tomó aproximadamente un minuto para cada muestra ejecutable, por lo tanto, aproximadamente 100 horas de tiempo de procesamiento para examinar con éxito los programas maliciosos y capturar sus comportamientos.

Sin embargo, como se mencionó anteriormente, cada archivo de informe JSON variará según los comportamientos escupidos por el malware, por lo que es incompatible con el aprendizaje automático que espera que los datos de entrada sean todos iguales. Para resolver este problema, se deben aplicar pasos de preprocesamiento para todos estos archivos de comportamientos JSON.

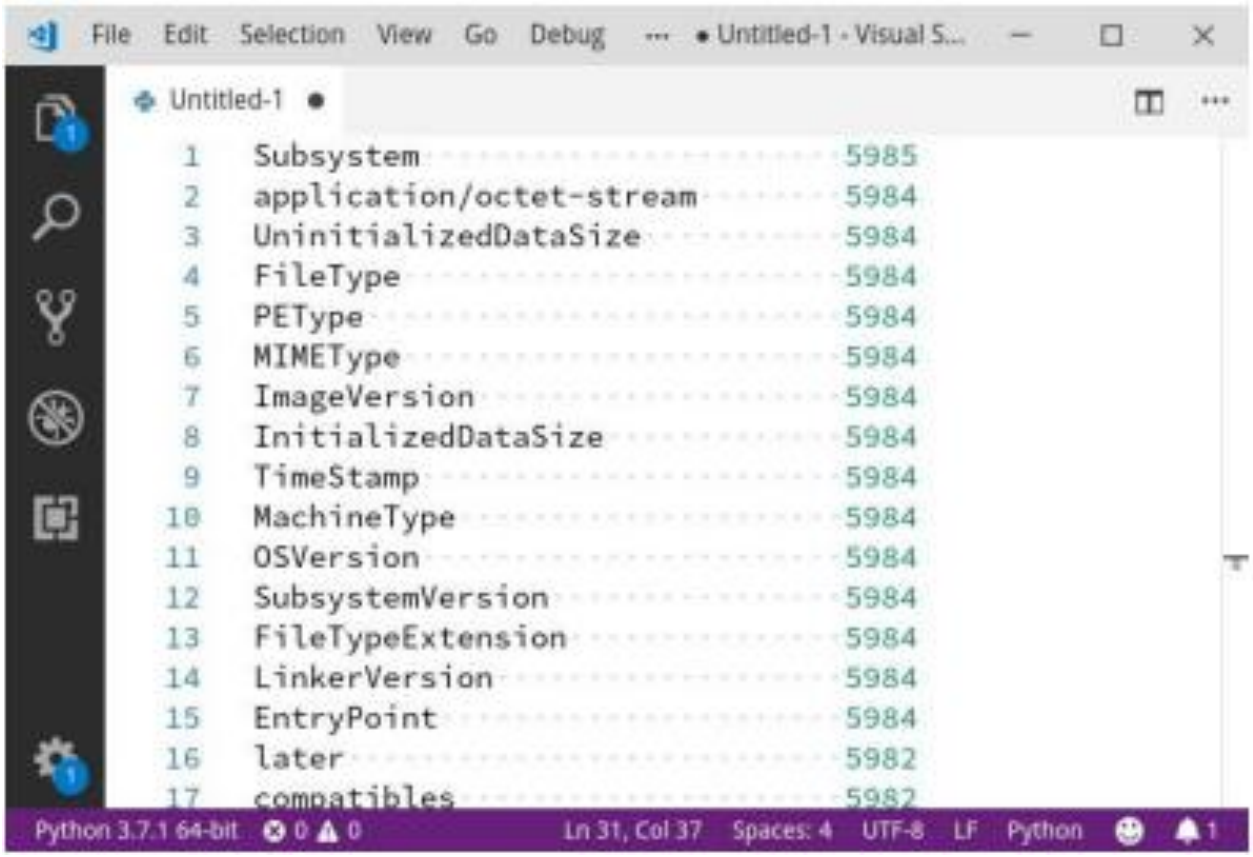

Ilustración 15-Archivo de lista de comportamientos producido por Cuckoo Sandbox

Fuente: Elaboración Propia

Al aplicar un script de Python codificado con las técnicas mencionadas anteriormente, se han muestreado los comportamientos de cada malware, se han extraído unigramo y se han elegido los 10.000 unigramos principales en función de su frecuencia de repetición. Estos 10,000 unigramas principales se marcan luego como un mapa de características, donde cada uno de los unigramas de los comportamientos de malware se ha asignado a uno o cero en función de su disponibilidad en los unigramas más frecuentes. Con estos archivos transformados, se pueden hacer diferencias visuales iniciales para tener una visión inicial de cómo se han transformado nuestros datos. A continuación se muestra un ejemplo de dos malwares completamente diferentes y sus familias en la representación a nivel de bits. La ilustración 16, muestra la visualización de datos que dan una idea sobre sus diferencias y cómo los algoritmos de aprendizaje automático, como los Autoencoders Denoising profundos, pueden explotar esta no

similitud para crear formas generalizadas de firmas de malware que pueden representar estos comportamientos de cadenas de bits en dimensiones mucho más bajas.

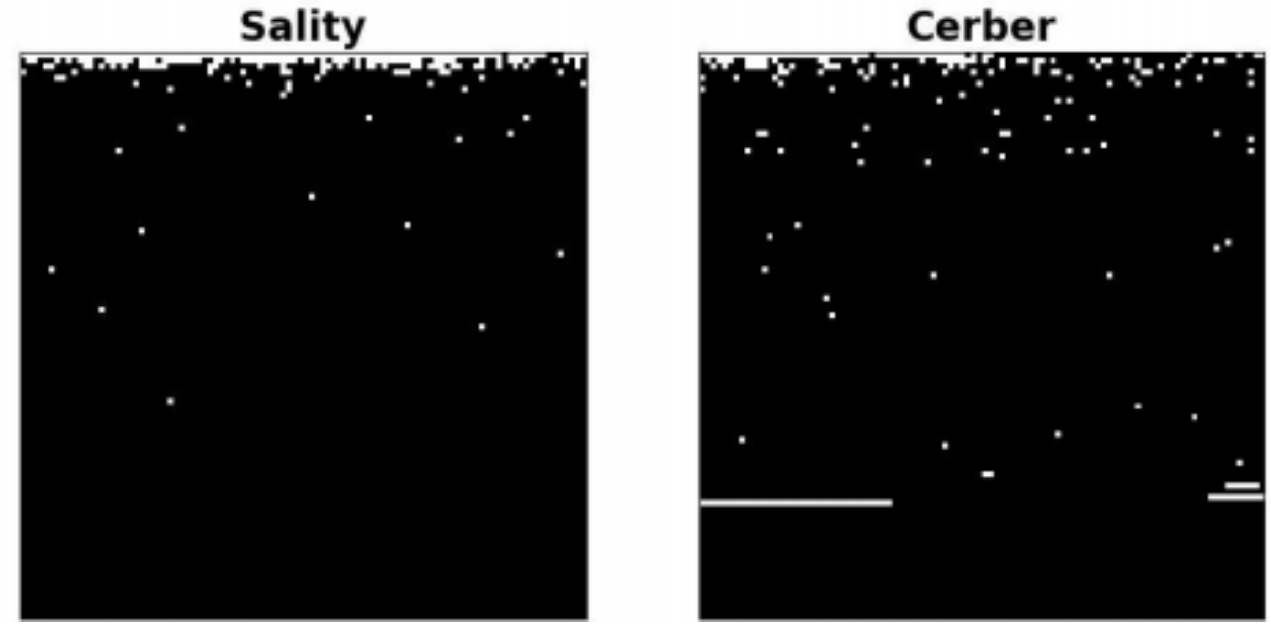

Ilustración 16-Visualización de cadena de bits de dos muestras de malware a nivel de bits

Fuente: Elaboración Propia

5.2 Diseño de la Red Neuronal

Del proceso de codificación anterior, con el uso de la técnica de 1 gram, cada comportamiento de malware ahora está codificado con una cadena binaria de 10,000 de tamaño fijo.

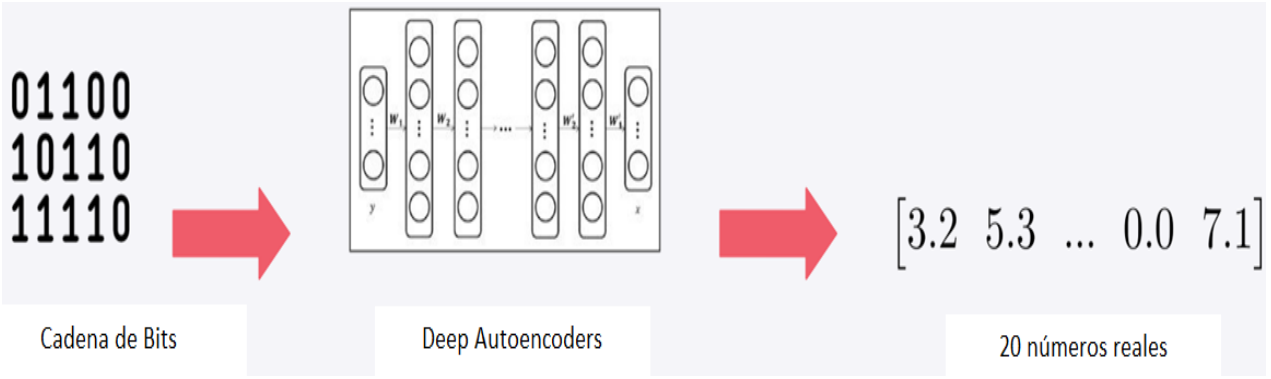

Ilustración 17-Reducción de dimensiones de entrada

Fuente: Elaboración Propia

Sin embargo, esto genera un problema para el entrenamiento. Ya que es una gran cantidad de entradas, 10,000, para la red neuronal. Entonces se debe buscar reducir sus dimensiones, como se muestra en la ilustración 17.

Para ello, se busca hacer una reducción de dimensión a los datos binarios. Se busca transformar los 10,000 datos binarios en solamente 20 números reales. Para ello se apoya de una arquitectura especial de aprendizaje profundo para realizar una reducción de dimensiones no lineales, una red de Deep Autoencoders, mostrada en la ilustración 18.

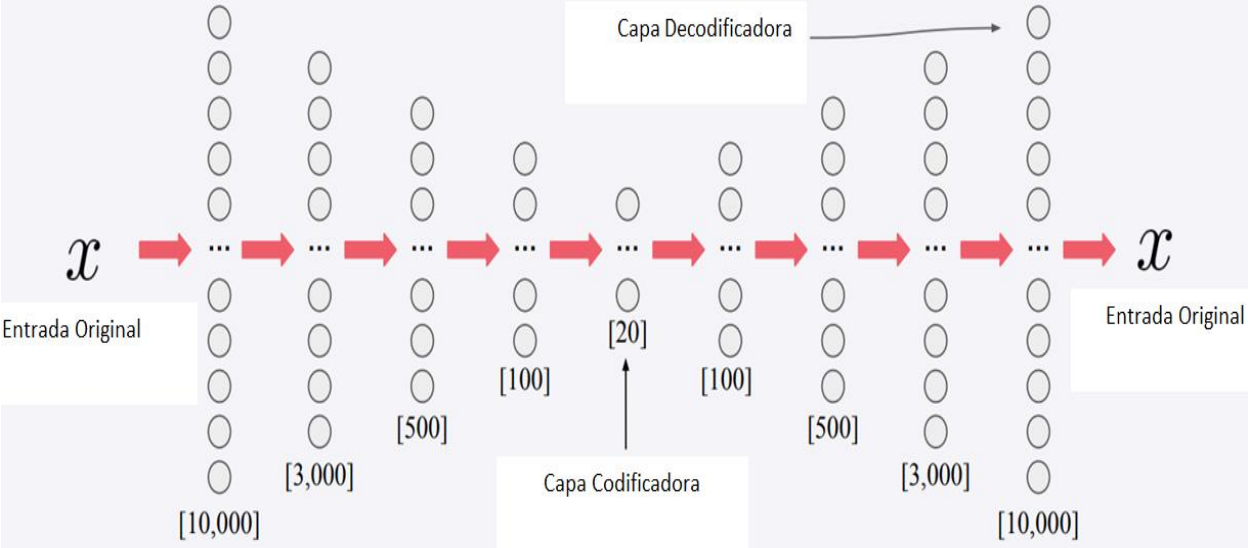

Ilustración 18-Deep Autoencoders

Fuente: Elaboración Propia

Este proceso de reducción de dimensiones es similar al entrenamiento de RNA. En donde se deben realizar diversas iteraciones o épocas para obtener el mejor funcionamiento de la misma. En la ilustración 19, se muestra el proceso de entrenamiento de la red de autoencoders en donde se observó la pérdida promedio en cada época. Se logra observar como esta pérdida fue disminuyendo con cada época.

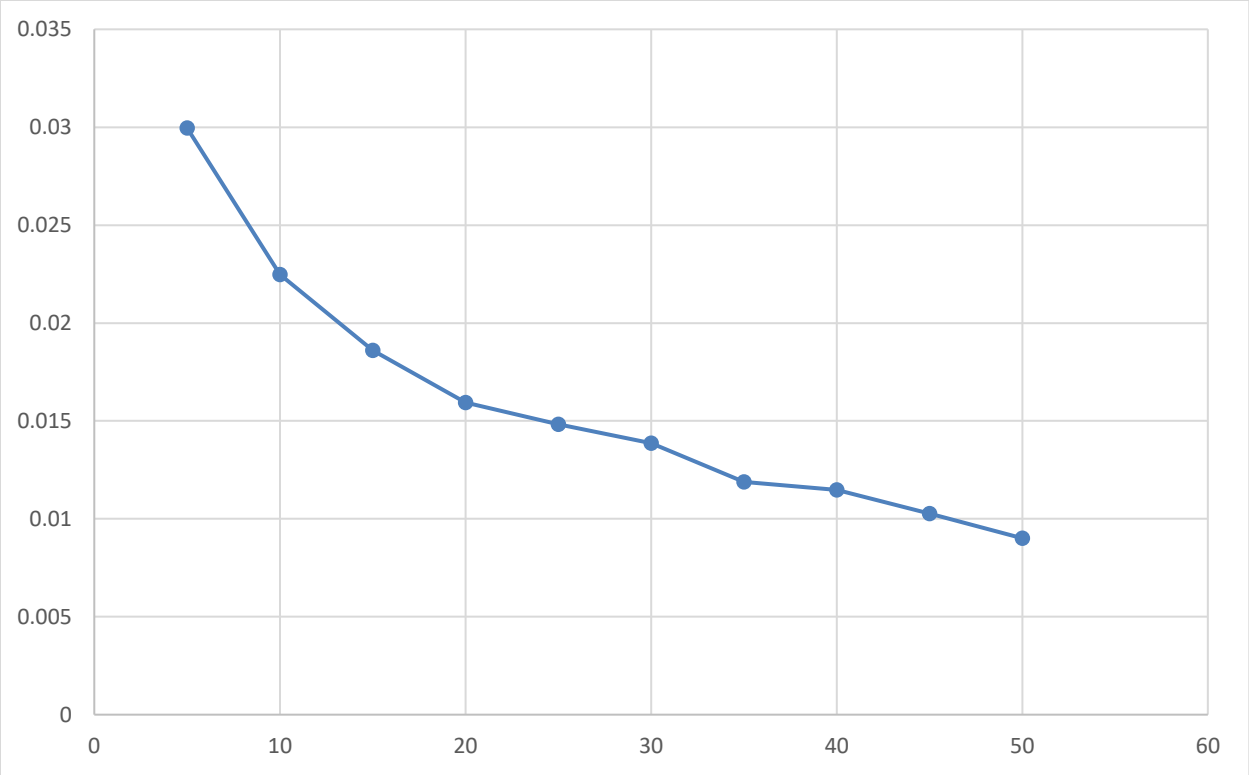

Ilustración 19-Pérdida promedio de la red de autoencoder

Fuente: Elaboración Propia

Finalmente, se obtiene el diccionario para cada malware. Este es el que será utilizado para el entrenamiento de la RNA. De esta manera, será capaz de clasificar cada malware según su comportamiento. En la ilustración 19, se muestran estos datos, ya listos para el entrenamiento de la RNA.

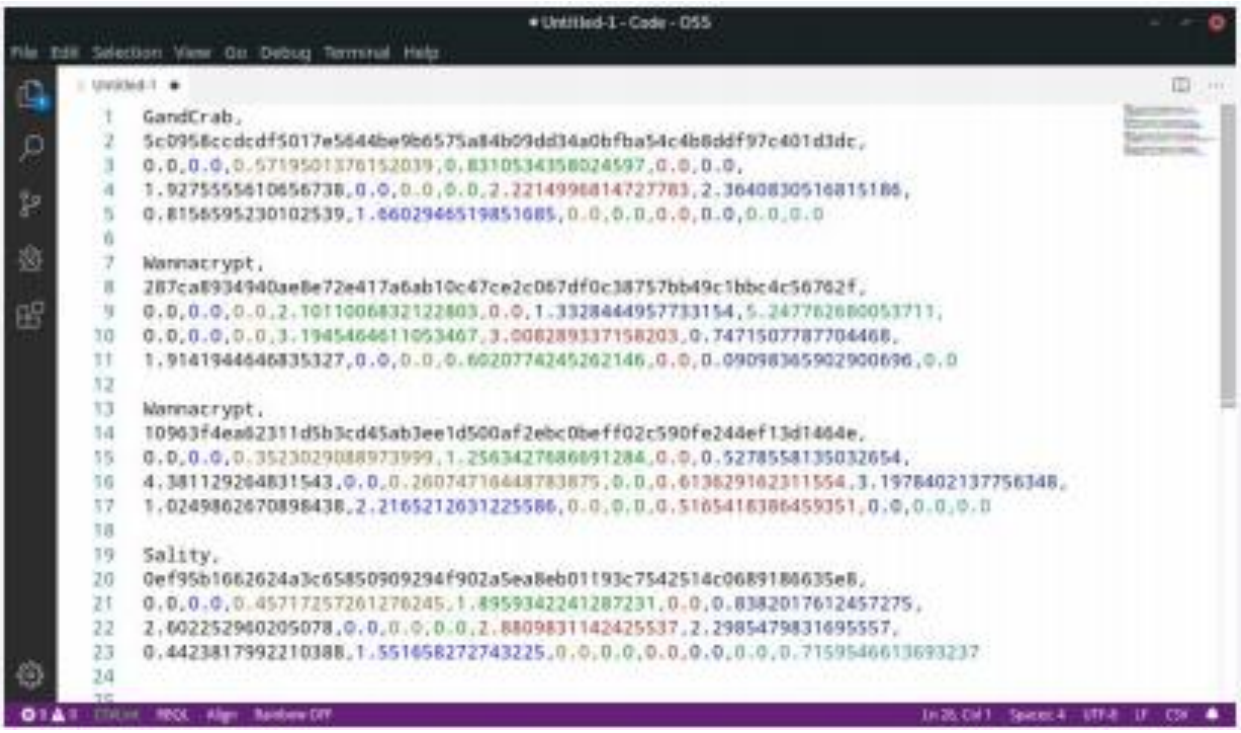

Ilustración 20-Entradas a la RNA

Fuente: Elaboración Propia

Para el entrenamiento de la RNA fue necesario dividir adecuadamente la base de datos. Por ello, se dividió la base de datos de la siguiente manera: 70% es el conjunto de entrenamiento representando 4200 muestras y el restante 30% es un conjunto para validación representando 1800 muestras, como se muestra en la ilustración 20.

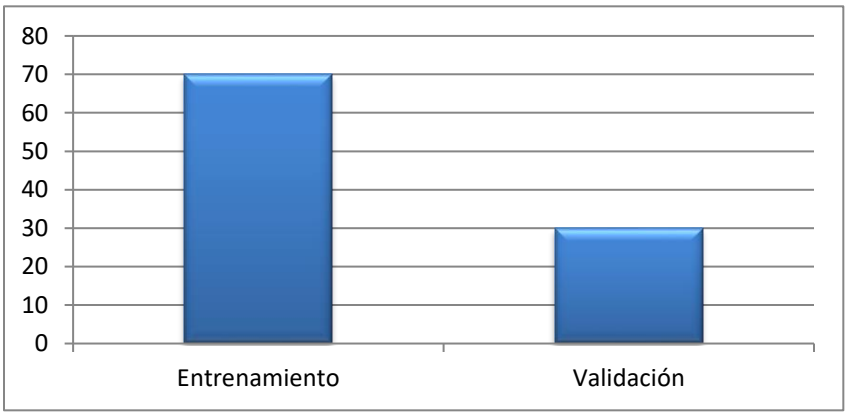

Ilustración 21-Distribución de base de datos

Fuente: Elaboración Propia

Posteriormente, se inició con el diseño de la RNA. Específicamente se realizó una arquitectura de una red neuronal profunda o *deep neural network* (DNN). Siendo una red neuronal densa con un gran número de neuronas y capas. La red diseñada contó con un total de 6 capas. Donde una capa era de entrada, una de salida y las restantes 4, eran capas ocultas. La capa de entrada consta de 20 neuronas que representa cada número real obtenido del proceso anterior, de reducción de dimensiones. La capa de salida consta de un total de 6 neuronas, donde cada una representa la probabilidad de que sea uno de los malware entrenados. Por otro lado, las capas ocultas tienes 60, 200, 40 y 15 neuronas. En la ilustración 21 se muestra esto a detalle.

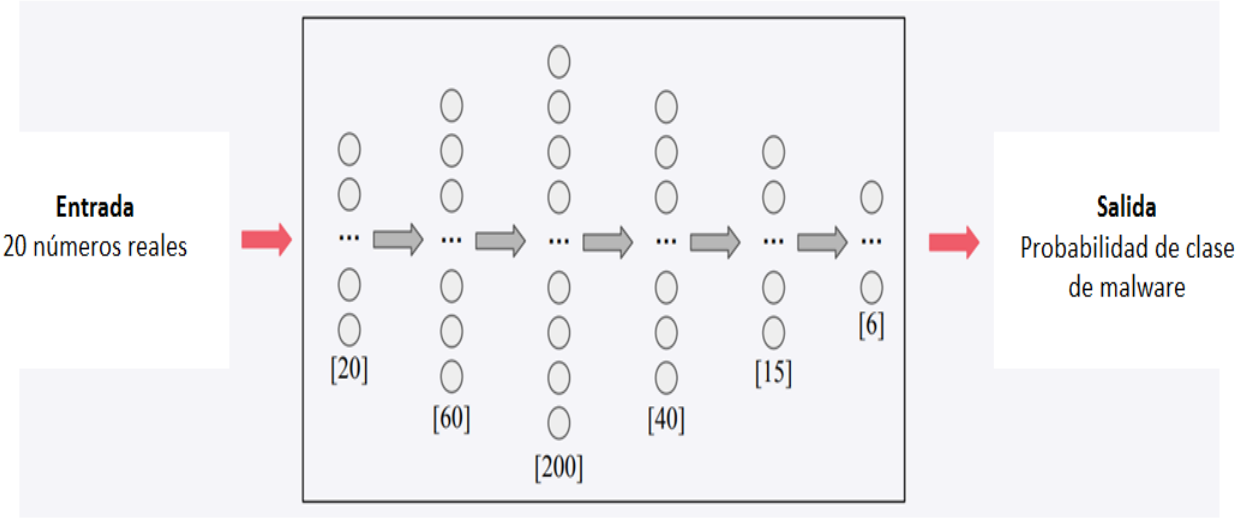

Ilustración 22-Diseño de la RNA

Fuente: Elaboración Propia

Finalmente, la capa de salida de la RNA, determinará la probabilidad de que el archivo analizado sea uno de los 6 malware con los cuales se entrenó la RNA. Por lo tanto, la suma de todos los valores de salida sería de 1. Tal como se muestra en la ilustración 22.

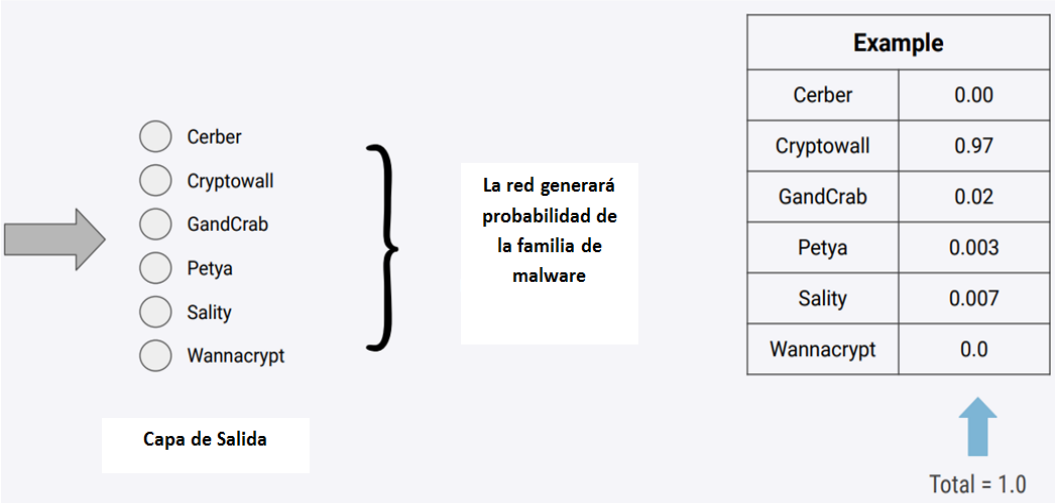

Ilustración 23-Capa de salida de la RNA

Fuente: Elaboración Propia

En el proceso de entrenamiento fue realizado múltiples iteraciones o épocas para obtener el mejor rendimiento de la RNA. En la ilustración 23, se muestra el comportamiento tanto de la perdida como la precisión de la RNA en el entrenamiento durante cada época. La mejor precisión obtenida fue de 91.41%.

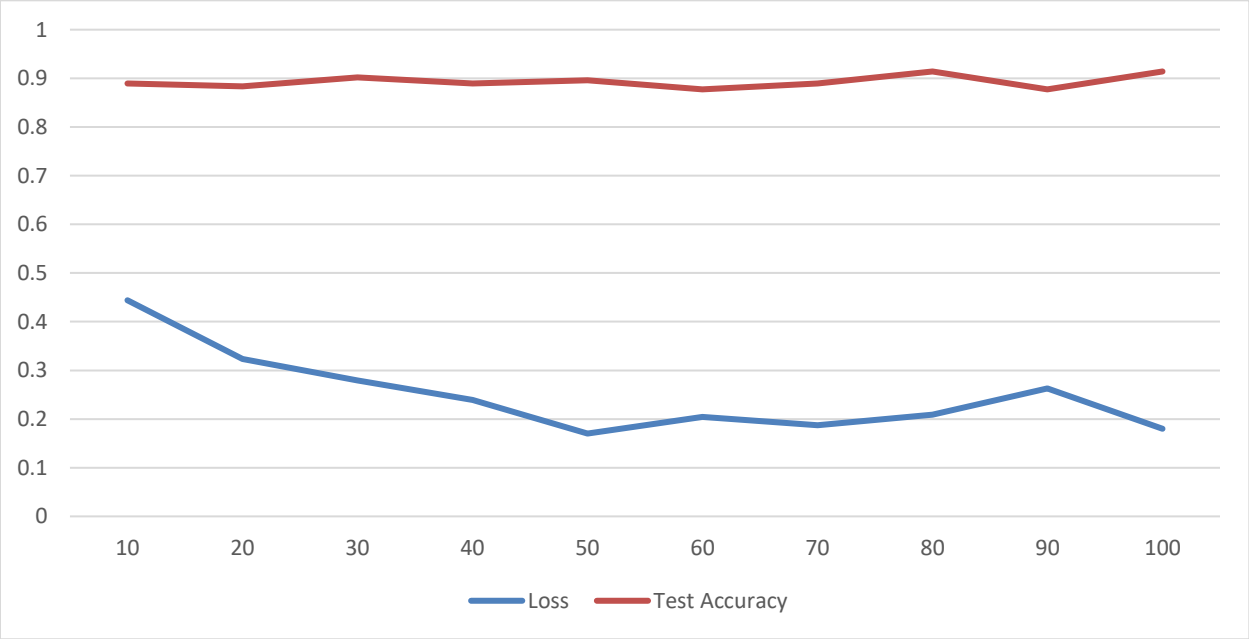

Ilustración 24-Entrenamiento de la DNN

Fuente: Elaboración Propia

CAPÍTULO VI. CONCLUSIONES

En el presente capítulo se presenta las conclusiones obtenidas las cuales estas basadas en los resultados mostrados en el capítulo anterior.

6.1 CONCLUSIÓN GENERAL

Se logró diseñar una red neuronal artificial que es capaz de determinar si el código de un archivo ejecutable contiene código malicioso (malware) a través del aprendizaje automático. Fue necesario el uso de dos redes de aprendizaje profundo clasificar un malware. Una fue la Deep autoencoders para la reducción de las dimensiones del conjunto de datos. Y una red neuronal profunda (DNN) para la clasificación del malware. La RNA finalmente posee una precisión del 91.41% para la clasificación de malware.

6.2 CONCLUSIONES ESPECÍFICAS

1) Para el entrenamiento de la RNA fueron construida una base de datos de un total de 6 familias de malware diferentes: Cerber, Cryptowall, GandCarb, Petya, Sality y Wannacrypt. Se obtuvo 1000 archivos ejecutables de cada familia de malware. Siendo un gran total de 6000 muestras para el entrenamiento de la RNA. La cual fue dividida en un 70% para entrenamiento y el restante 30% para validación.
2) Ya que el conjunto de datos de enteramiento eran archivos ejecutables, se procedió a utilizar técnicas del procesamiento del lenguaje natural. Con la técnica de extracción basada en el 1 gram, se logró transformar estos archivos ejecutables en cadenas de bits.
3) A causa de que la cadena de bits representaba un gran número de entradas para la RNA, se buscó reducir las dimensiones. Para ello, se utilizó una arquitectura de aprendizaje profundo que permitió la reducción, una red de Deep autoencoders. Donde se redujo la cadena de bits de 10,000 a 20 números reales.

6.3 RECOMENDACIONES

1) En caso de no haber predicción mayor a 0.92, se asume que la red no podría predecir un archivo ejecutable.
2) Investigar un método de red neuronal que sea para la detección en tiempo real.
3) Crear una clase o categoría "Otros" y agregar otros ejemplos de familias y entrenar todo este conjunto.

BIBLIOGRAFÍA

Amamra, A., Talhi, C., & Robert, J. M. (2012). Smartphone malware detection: From a survey towards taxonomy. *2012 7th International Conference on Malicious and Unwanted Software, Fajardo, PR*, 79-86.

Baca Urbina, G. (2010). *Evaluación de Proyectos* (6ta ed.). McGraw-Hill.

Bissell, K., Lasalle, R. M., & Cin, P. D. (2019). The Cost of Cybercrime: Ninth Annual Cost of Cybercrime Study, Unlocking the Value of Improved Cybersecurity Protection. *Accenture Security*.

Chanana, A., Singh, S., & Paliwal, K. K. (2017). Malware detection using GA optimized K-means and HMM. *International Conference on Computing, Communication and Automation (ICCCA)*, 355-362.

Chapman, I. M., & Leblanc, S. P. (2011). Taxonomy of cyber attacks and simulation of their effects. *Proceedings of the 2011 Military Modeling & Simulation Symposium*, 73-80.

Check Point Research. (2020). Security Report. *Check Point Software Technologies Ltd*.

Conti, M., Gangwal, A., & Ruj, S. (2018). On the Economic Significance of Ransomware Campaigns: A Bitcoin Transactions Perspective. *Computers & Security*, *79*, 162-189.

De Donno, M., Dragoni, N., Giaretta, A., & Spognardi, A. (2018). DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation. *Security and Communication Networks*, *2018*, 30.

Feizollah, A., Anuar, N. B., Salleh, R., & Abdil Wahab, A. W. (2015). A review on feature selection in mobile malware detection. *Digital Investigation*, *13*, 22-37.

Imran, M., Arif, T., & Shoab, M. (2018). A Statistical and Theoretical Analysis of Cyberthreats and its Impact on Industries. *International Journal of Scientific Research in Computer Science Applications and Management Studies*, *7*(5).

Khlaif Abuzaid, A. M., Saudi, M. M., Taib, B. M., & Abdullah, Z. H. (2013). An efficient Trojan Horse Classification (ETC). *International Journal of Computer Science*, *10*(2), 1694-1704.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436-444.

Milošević, N. (2013). History of malware. *Computer security*, 1-11.

Minsky, M., & Pappert, S. (1969). *Perceptrons*. MIT Press.

Ngiam, J., Khosla, A., Kim, M., Nam, J., & Lee, H. (2019). Multimodal Deep Learning. *ICML*, *10*.

Ponce Cruz, P. (2010). *Inteligencia artificial: Con aplicaciones a la ingeniería*. Alfaomega.

Ramon, & Cajal, S. (1911). *Histologie du Système Nerveux*.

Raschka, S. (2015). *Python machine learning*. Packt Publishing Ltd.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan Books.

Saini, H., Rao, S. Y., & Panda, T. C. (2012). Cyber-Crimes and their Impacts: A Review. *International Journal of Engineering Research and Applications*, *2*(2), 202-209.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85-117.

Simmons, C. B., Shiva, S. G., Bedi, H., & Dasgupta, D. (2014). AVOIDIT: A Cyber Attack Taxonomy. *Annual Symposium on Information Assurance (ASIA)*, 2-12.

Souri, A., & Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. *Human Centric Computing and Information Sciences*, *8*(3).

Symantec. (2019). Internet Security Threat Report. *Symantec Corporation*, *24*.

Tang, M., & Qian, Q. (2019). Dynamic API call sequence visualisation for malware classification. *IET Information Security*, *13*(4), 367-377.

Wang, A., Liang, R., Liu, X., Zhang, Y., Chen, K., & Li, J. (2017). An Inside Look at IoT Malware. *Industrial IoT Technologies and Applications. Industrial IoT 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, *202*, 176-186.

Watson, M. R., Shirazi, N., Marnerides, A. K., Mauthe, A., & Hutchison, D. (2016). Malware Detection in Cloud Computing Infrastructures. *IEEE Transactions on Dependable and Secure Computing*, *13*(2), 192-205.

WEFORUM. (2020). The Global Risks Report 2020. *World Economic Forum*. https://www.weforum.org/reports/the-global-risks-report-2020

Zhang, Z. K., Yi Cho, M. C., Wang, C. W., Hsu, C. W., Chen, C. K., & Shieh, S. (2014). IoT Security: Ongoing Challenges and Research Opportunities. *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, Matsue*, 230-234.