Universitat Oberta de Catalunya

uoc.edu

Machine Learning aplicado a la seguridad

Edith Galmés González
Master Universitario en Seguridad de las TIC
Análisis de Datos
Enric Hernández Jiménez
Helena Rifà Pous

31/12/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Machine Learning aplicado a la seguridad*
Nombre del autor:	*Edith Galmés González*
Nombre del consultor/a:	*Enric Hernández Jiménez*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	12/2019
Titulación:	*MISTIC*
Área del Trabajo Final:	*Análisis de Datos*
Idioma del trabajo:	*Lengua española*
Palabras clave	*Machine Learning, Seguridad*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Este proyecto abarca el estado del arte del Machine Learning aplicado a la seguridad informática. El objetivo de este proyecto es explicar el Machine Learning, qué es, y cómo se aplica dentro de la seguridad.

Primero se realiza una introducción a la inteligencia artificial para contextualizar el Machine Learning. Más adelante se explica el Machine Learning, sus diferentes modelos y algoritmos más comunes.

Después se enumeran las diferentes aplicaciones del Machine Learning y se realiza un estudio de sus aplicaciones en la seguridad. Se estudian más en profundidad estudios y documentación realizados en la detección anomalías, detección de malware y ataques DDoS.

Para finalizar se realizan unas conclusiones sobre el estudio realizado.

Abstract (in English, 250 words or less):

This project covers the state of the art of Machine Learning applied to computer security. The objective of this project is to explain Machine Learning, what it is, and how it can be applied within security.

To contextualize Machine Learning, an introduction to artificial intelligence is made. To continue, it is explained Machine Learning, its different models and the most common algorithms that are using it.

Then the different applications of Machine Learning are listed and a study of their security applications is carried out. Studies and documentation on anomaly detection, malware detection and DDoS attacks are studied more in depth.

Finally, conclusions were made about the study.

Índice

Lista de figuras

1 Introducción

1.1 Contexto y justificación del Trabajo

Este trabajo pretende explicar y desarrollar el Machine Learning y sus diferentes aplicaciones en la seguridad. Con el Machine Learning se pueden abarcar muchos de los problemas que se presentan dentro de la seguridad.

Los ciberataques son reconocidos como principales amenazas en el mundo digital para empresas y organizaciones gubernamentales. Cada vez son más sofisticados y difíciles de detectar, por ese motivo, el software de siempre contra muchos ataques puede no ser suficiente para proteger los datos. Por ese motivo la Inteligencia Artificial en ciberseguridad puede ayudar a detectar las nuevas amenazas a través del Machine Learning.

1.2 Objetivos del Trabajo

El objetivo del trabajo es obtener el estado del arte del Machine Learning aplicado en la seguridad informática.

1.3 Enfoque y método seguido

Este trabajo consta de realizar una introducción al Machine Learning, sus aplicaciones en la seguridad y cómo lo han desarrollado diferentes estudios. Se ha realizado un estudio de qué es el Machine Learning, sus modelos, sus aplicaciones, a través de Estudios realizados, libros publicados y documentación encontrada.

1.4 Breve sumario de productos obtenidos

Se consigue un estudio del Machine Learning abarcado sus diferentes características y aplicaciones en el mundo real y aplicado a la seguridad. De esta manera se puede tener una visión global del Machine Learning y los diferentes estudios que se han ido realizando.

1.5 Breve descripción de los otros capítulos de la memoria

Primero nos encontramos con el capítulo de Inteligencia Artificial. Este capítulo se escribe para poder dar una introducción de dónde proviene el Machine Learning y contextualizarlo.

En el siguiente capítulo se encuentra Machine Learning, en este capítulo se explica el concepto, se explican los diferentes modelos y los algoritmos más comunes. También se realiza una recopilación de servicios aplicados con Machine Learning.

En el tercer capítulo, ciberataques, es donde podemos encontrar una introducción a los problemas y estado actual de la seguridad informática y como puede aplicar al Machine Learning.

En el cuarto capítulo se encuentran la explicación de diferentes estudios relacionados con el Machine Learning en la seguridad informática.

En el último se realizan las conclusiones extraídas de esta investigación.

2 Inteligencia artificial

La Inteligencia Artificial (IA) se podría definir como la disciplina que se encarga de que una máquina pueda imitar, percibir y aprender del entorno como si fuera un humano. La Inteligencia artificial está presente en diferentes ámbitos como la salud, la educación la fabricación y la ciberseguridad.

A medida que avanza la investigación en la IA hay más empresas que apuestan y hacen uso de ella, dedicándole recursos, inversiones o realizando contrataciones de expertos en IA. Entre muchas de las aplicaciones que tiene la IA se encuentra el uso contra los ataques cibernéticos e intrusiones que puede haber en una compañía. La IA permite la protección sobre algunos de los ataques que se pueden encontrar en la red o mitigar su impacto.

La Inteligencia Artificial se trata de una técnica que permite a las máquinas imitar el comportamiento humano donde se engloban el Machine Learning y el Deep Learning.

Hay dos tipos de IA, la robusta o la aplicada:

- La IA robusta o Strong IA trata sobre una inteligencia real en el que las máquinas tienen la capacidad cognitiva parecida a los humanos.
- La IA aplicada o Applied IA trata sobre la utilización de los algoritmos y el aprendizaje automático, donde se encuentran el Machine Learning y el Deep Learning.

Por otro lado, el Machine Learning o aprendizaje automático es un subconjunto de técnicas de la IA que utilizan métodos estadísticos para permitir que las máquinas mejoren según la experiencia que obtengan. Por último, el Deep Learning o aprendizaje profundo es un subconjunto del aprendizaje automático que hace posible el cálculo de la red neuronal multicapa. Dentro del software de análisis de datos el Deep Learning simula un sistema de redes artificiales de neuronas.

En la siguiente figura vemos como se engloban estos tres conceptos:

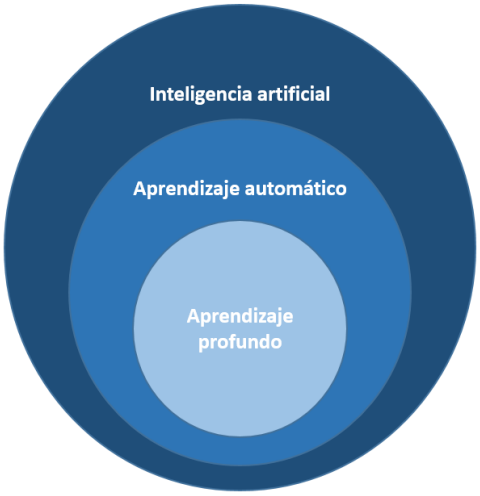

Fig. 2.1 Niveles de la IA

2.1 Cronograma

La Inteligencia Artificial aparece entre las décadas de los años 50 y 70. El primer artículo donde se plantea la combinación entre una máquina y la tecnología para realizar la organización de conocimiento complejo data del 1945. Este artículo, nombrado *As we may think*, se publica en la revista Atlantic Monthly escrito por Vannevar Bush.

Cinco años más tarde, Alan Turing hizo el estudio *Computering Machinering and Intelligence* sobre las bases de la Inteligencia artificial, donde las máquinas simulan a los seres humanos y la capacidad de hacer cosas inteligentes. Alan Turing proponía realizar una prueba, para descubrir si una máquina podía ser inteligente o no, esta prueba se conoce como el test de Turing. En ese mismo año Isaac Asimov formuló las tres leyes de la robótica.

En 1951 fue escrito el primer programa basado en IA, Christopher Strachey desarrolló un programa de damas y Dietrich Prinz escribió uno para el ajedrez. Ambos fueron escritos con la máquina de Ferranti Mark de la Universidad de Manchester. Al mismo tiempo, entre 1952 y 1962 Arthur Samuel escribió el primer programa de ajedrez capaz competir realmente con un ser humano.

No es hasta 1956 cuando aparece el término "Inteligencia Artificial" en Dartmouth durante una conferencia convocada por John McCarthy. MacCarthy y Marvin Minsky fundaron el MIT AI LAB en 1959.

En 1964 Danny Bobrow crea en el MIT la primera demo de un programa de IA que entiende el lenguaje natural para resolver un problema de algebra.

En 1965 Joseph Weizenbaum inventó el primer chat bot Eliza en el MIT que puede dialogar cualquier tema en inglés.

En 1974 Ted Shortliffe en Stanford AI Lab demostró un enfoque muy práctico basado en reglas para los diagnósticos médicos.

En 1989 Dean Pomerleau creó el primer coche autónomo usando una red neuronal. El proyecto llamado ALVINN (an Autonomous Land Vehicle in a Neuronal Network) implementó la red neuronal que funcionaba correctamente, pero el hardware limitó mucho el avance.

En 1997 la máquina de ajedrez de IBM llamada Deep Blue ganó al entonces campeón Garry Kasparov. Ese mismo año, el programa Computer Logistello ganó Takeshi Murakami en el juego de mesa japonés llamado Othello.

En 1999 Sony presenta AIBO, el perro robot. AIBO fue comercializado para uso doméstico como robot de entretenimiento, pero también sirvió para fines educativos para realizar investigaciones en las universidades.

En 2004 DARPA, en un concurso para vehículos autónomos, realiza el primer reto de larga distancia para coches sin conductor en el mundo. Ninguno de los vehículos que participaba en el reto pudo terminar la ruta, por lo que se programó el siguiente reto para el año siguiente. Ese mismo año los robots de exploración robótica de la NASA Spirit y Opportunity navegaron de forma autónoma por la superficie de Marte.

En 2009 Google empieza con el proyecto Google *self-driving car project*, un coche autónomo para conducir de manera autónoma e ininterrumpida 10 rutas de 100 millas. El líder del proyecto era Sebastian Thrun, director del Stanford Artificial Intelligence Laboratory y coinventor de Google Street View.

En 2010 aparecen las ciencias narrativas con la capacidad y estabilidad directa para escribir informes. En 2011 el ordenador de IBM Watson gana a los mejores jugadores del concurso de preguntas y respuestas *Jeopardy!*.

Seguidamente aparecen las aplicaciones para teléfonos inteligentes que utilizan lenguaje natural para responder preguntas, hacer recomendaciones y realizar acciones. Aparecen Siri de Apple en 2011, Google Now de Google en 2012 y Cortana de Microsoft en 2014.

En 2015 AlphaGo, un programa informático de inteligencia artificial de Google DeeMind, ganó a Fan Hui, el campeón del juego de mesa Go.

En 2017 se celebró la Conferencia de Asilomar sobre la IA, para discutir la ética de la IA y cómo lograr una IA beneficiosa. Ese mismo año se publica el primer algoritmo desarrollado para vencer a jugadores humanos en juegos de información imperfecta, como sería el póker. Poco después, se crea el programa AI Libratus de póker y se pone a prueba contra seres humanos, donde el programa gana a cada uno de sus oponentes.

En 2018 el lenguaje de procesamiento de la IA de Alibaba supera a los mejores humanos en una prueba de lectura y comprensión de la Universidad de Stanford. Ese mismo año, Google anuncia un servicio que permite a un asistente de IA reservar citas por teléfono.

En 2019 Google ha presentado su Doodle, que, con la IA, realiza un homenaje a Sebastian Bach. A partir de dos compases, Doodle con la IA crea el resto de la melodía.

A continuación, se muestra el Cronograma de la IA con los hechos comentados:

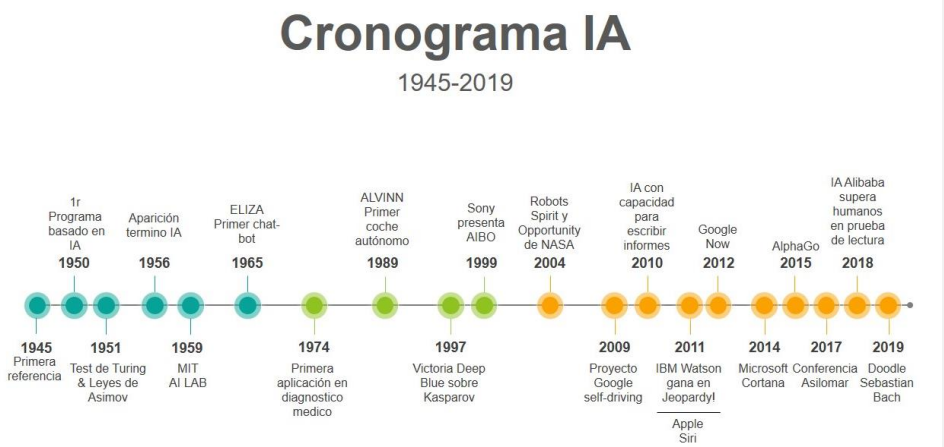

Fig. 2.2 Cronograma IA

3 Machine Learning

Hoy en día la Inteligencia Artificial (IA) cada vez es más común en aplicaciones del día a día, y entre sus aplicaciones con más fuerza se encuentra el aprendizaje automático en las máquinas, el Machine Learning.

El Machine Learning (ML) es una disciplina de la IA que tiene como objetivo desarrollar técnicas, a través de algoritmos, para que las máquinas puedan aprender de manera automática. Este aprendizaje se genera según una entrada de datos, experiencias, decisiones, entre otros factores, con el fin de poder aprender y predecir o sugerir resultados. El conjunto de datos (Dataset), es el recurso principal para realizar el aprendizaje y por tanto la predicción. Este conjunto de datos dispone de instancias, características, propiedades y entre otros distintivos que los definen.

El ML se centra principalmente en el diseño de sistemas, que permite aprender y hacer predicciones basadas en cierta experiencia, que en este caso es la entrada de datos que recibe la máquina.

En la teoría del aprendizaje hay diferentes puntos a tener en cuenta. Cuando una máquina está aprendiendo hay que saber concretar qué es lo que queremos que aprenda, cuál es su problema a resolver, la organización y su contexto. Una vez definido el objetivo, dependerá de cuantos datos y de la calidad de la información que entremos para que el modelo aprenda a reconocer un patrón. El proceso de cálculo del patrón automático, de reconocimiento y toma de decisiones inteligente se basa en datos de entrenamiento donde hay que definir cuándo la recolección de datos es suficiente.

Por último, se debe garantizar el cómo asegurar que el modelo funciona. Los métodos de aprendizaje automático se evalúan comparando los resultados de aprendizaje de los métodos aplicados en el mismo conjunto de datos o cuantificando los resultados de aprendizaje del mismo. Debe evaluarse empíricamente debido a su alto rendimiento. Dependiendo del tipo de experiencia de entrenamiento que haya experimentado la máquina, hay que tener en cuenta las métricas de evaluación de los resultados y cómo se ha definido el problema.

Hoy en día, dado que el aprendizaje automático es un sistema basado en procesamiento y análisis de datos, uno de los factores que ha permitido avanzar más en este campo es la potencia computacional de alta gama y la gran capacidad que de almacenar datos.

A continuación, se muestra un esquema genérico de cómo funciona el Machine Learning.

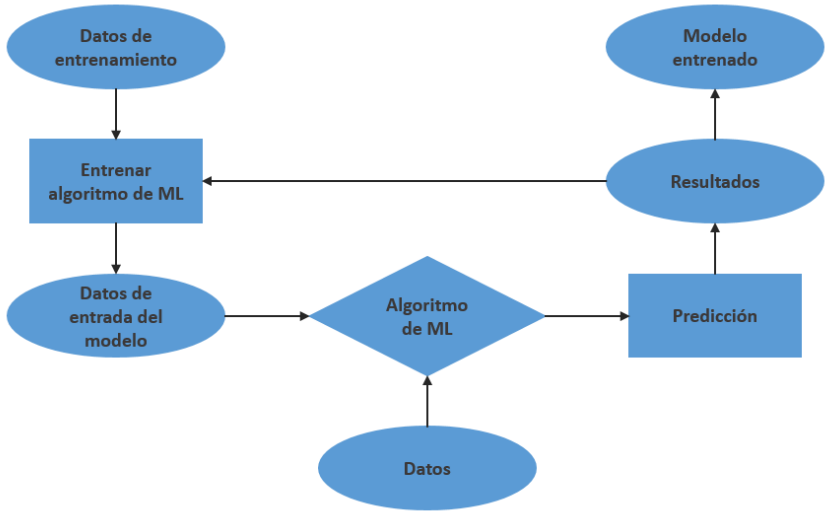

Fig. 3.1 Esquema Genérico Machine Learning

Se encuentran muchas aplicaciones del Machine Learning en nuestro día a día. Esto es gracias a que resulta fácil y práctico aplicar las técnicas de Machine Learning a muchos de los problemas cuotidianos.
Seguidamente se enumeran alguno de sus ejemplos más utilizados:

- Las redes sociales, en este caso, uno de los ejemplos son las recomendaciones que te indican, los amigos que puedes agregar, o personas que te pueden interesar.

- La detección de correos spam o correos no deseados, se realizan a través de la clasificación de los correos. Con el ML se clasifican los mensajes y se realiza un entrenamiento en el algoritmo, que después servirá para clasificar estos correos no deseados.

- En el reconocimiento de imágenes para detectar personas o patrones en imágenes. En más profundidad podemos encontrar el reconocimiento facial en el Deep Learning.

- El reconocimiento de caracteres, donde su aplicación tiene un gran impacto en el envío de paquetería y correo, para clasificar la información.

- La recomendación de productos, donde se recomiendan a través de los patrones del usuario, los productos que le interesen. Por ejemplo, utilizado por Amazon, para que sus clientes reciban recomendaciones para realizar sus compras en la app o vía web.

- El reconocimiento de voz realizado a partir de las ondas de sonido sintetizadas por el micrófono de tu smartphone, pc, dispositivo inteligente o de tu coche, que, a través de los algoritmos de ML son capaces de interpretar órdenes. Esto se debe a que pueden limpiar el ruido, intuir los silencios entre palabras y comprender el idioma en el que estés hablando para procesarlo y realizar las acciones. Ejemplos de esta aplicación serían el Google Home o Siri.

3.1 Modelos del Machine Learning

Dentro del Machine Learning podemos encontrar diferentes modelos, de entre ellos, destacan los siguientes:

Modelo supervisado

Se trata de un aprendizaje donde el modelo aprende la asociación entre diferentes atributos, sacados de un conjunto de datos de entrada al algoritmo, para conseguir las respuestas esperadas. Este aprendizaje es muy útil en los casos que hay atributos y etiquetas para llevar a cabo el aprendizaje.

Se va especificando a la máquina si es o no correcto el resultado para que el algoritmo aprenda mediante la disminución del error. Como resultado el modelo aprende la asociación entre los ejemplos. El modelo se considera que puede dejar de entrenarse cuando llega a un nivel aceptable en su predicción.

Dentro del modelo supervisado encontramos el de clasificación y el de regresión:

- Modelo clasificación: es un modelo que predice valores discretos; predice una categoría. Por ejemplo, si es blanco o negro. Con N ejemplos, un conjunto de atributos y para cada atributo una clasificación observada.

- Modelo regresión: es un modelo que predice valores continuos, por ejemplo, el valor de algún inmueble.

Una de las ventajas del modelo supervisado es que el modelo se memorice los resultados dando una precisión alta en el conjunto de datos, por el contrario, cuando el modelo utilizado es demasiado complejo para los datos, tiene una precisión baja.

Por otro lado, como este modelo se basa en reducir el error, puede darse a que el modelo asigne o sugiera todas las respuestas a una misma categoría, dando así un desequilibrio de clases.

Modelo no supervisado

En el modelo no supervisado el aprendizaje consta de que, a partir de una entrada de datos sin etiquetar, aprenda alguna manera de agruparlos, organizarlos y encontrar una estructura de ellos. Por lo que, con la entrada de datos, sin tener un conjunto de datos entrenados, se interpretan los datos de entrada, pasan por el algoritmo, se procesan y da como resultado al modelo entrenado. Para obtener un refuerzo en el aprendizaje, hay una secuencia entremedia de decisión, donde se decide si es o no correcto. Una vez entrenado, los algoritmos descubren por su cuenta el resultado.

Como se ha comentado, existen diferentes modelos para entrenar a las máquinas. Estos modelos pueden utilizar diferentes variables con el fin de aprender de manera automática. Los diferentes valores pueden ser:

- Etiquetas: son el valor que está prediciendo el modelo, y pueden usarse ciertos filtros para conseguirlo. Un ejemplo de etiqueta sería: blanco o negro, donde la etiqueta comprendería uno de estos dos valores.
- Atributos: son variables de entrada que representan los datos
- Ejemplos: son una instancia de datos que pueden dividirse en ejemplos etiquetados o sin etiquetar.

El entrenamiento que implica el aprendizaje es el ciclo del modelo, y la inferencia donde se entrena a ejemplos sin etiqueta. Después, una vez que el modelo se entrena con ejemplos etiquetados, ese modelo se usa para predecir la etiqueta en ejemplos sin etiqueta.

En consecuencia, cuantos más valores tenga de entrada un modelo, más preciso podrá ser. Esto se debe a que, en este proceso de aprendizaje, con una entrada de datos, es donde se detectan los patrones y el modelo intenta encontrar nuevos patrones que no hayan sido conocidos antes de entrenar al modelo.

En el conjunto de datos no hay una salida esperada asociada a ellos. Cuando el modelo ya tiene datos etiquetados, calcular la precisión es tan sencillo como detectar si el modelo ha predicho correctamente los datos etiquetados. Como ejemplo dentro del modelo no supervisado tendríamos el procesamiento de imagen, donde se analizan sólo algunos datos, por ejemplo, un grupo de pixeles.

3.2 Algoritmos en Machine Learning

Los algoritmos han evolucionado con el objetivo de analizar y obtener mejores resultados: árboles de decisión, *clustering* para almacenar y leer grandes volúmenes de datos, redes Bayesianas y muchas más técnicas que los programadores de *data science* pueden utilizar para desarrollar nuevos algoritmos. A continuación, se explican algunos de los algoritmos de ML más utilizados.

3.2.1 Árbol de decisión

Un árbol de decisión es la representación gráfica de las posibles respuestas que se pueden obtener según las decisiones y las condiciones que estas puedan tener. Este algoritmo aprende según el conjunto de datos añadidos analizando qué es lo que se puede predecir. Este algoritmo se basa en el modelo supervisado.

Con los datos de entrada del árbol podemos clasificar la información según sus características, a qué dominio pertenece y la salida que se quiere predecir. En el caso del dominio puede ser según categoría, que sería el modelo de clasificación o numérico que sería el modelo de regresión.

Para construir el árbol hay que tener en cuenta qué tipo de preguntas se van a realizar y cuando. Es muy importante debido a que repercute directamente en el algoritmo, para que éste sea lo más preciso posible. Hay que analizar las preguntas para saber por cuál se debe empezar primero y obtener un resultado óptimo.

Para analizar los datos de entrada se deben realizar unos cálculos. Debemos cuantificar la incerteza e intentar reducirla según la pregunta que se vaya a empezar para desarrollar el árbol. Para ello podemos hacer uso del concepto de *Information Gain*. Se trata de buscar una respuesta de manera recursiva para ir disminuyendo las posibilidades. En cada nodo iteraremos los cálculos realizados para construir el árbol con sus diferentes nodos hasta no tener más posibilidades. Consecuentemente, identificar las preguntas que se vayan a realizar al algoritmo tiene un punto muy importante a la hora de empezar.

En resumen, dada una entrada de datos, la tarea que tendrá que realizar el algoritmo será predecir la salida. Con él se examina el valor de un atributo y se pregunta si el resultado predicho es correcto o no.

Terminología del árbol de decisión

Este algoritmo se compone de diferentes partes: los nodos de decisión, los nodos de respuesta, combinables con etiquetas y los nodos raíz. El árbol de decisión empieza en la raíz, crece con los nodos de decisión y finaliza en nodos respuesta.

Los diferentes nodos se definen de la siguiente manera:

- El nodo de decisión o nodo raíz (*root node*) es el nodo principal vinculado a uno de los atributos. Éste tendrá mínimo dos ramas, por donde el árbol irá creciendo y dividiendo según las según las decisiones y condiciones que haya.
- El nodo interno (*splitting node*) se sitúa entre el nodo raíz y el nodo respuesta, por donde se dividen los valores. Tiene asociada una condición, y según la respuesta, decidirá el camino que el árbol va a realizar.
- El nodo respuesta (*leaf node*), también llamado nodo hoja, es el nodo que tiene la etiqueta donde indica la opción que se clasifica. Se sitúa por debajo del nodo decisión y de él no colgará ningún otro nodo.
- Prunning es el término para referirse a las ramas que se eliminan del árbol cuando en la toma de decisiones estas no son elegidas.

Toma de decisión

Como se ha comentado, uno de los primeros puntos a tener en cuenta es saber qué pregunta realizar primero para la creación del árbol. Es muy importante saber por dónde hay que empezar a realizar la creación de las ramas, es decir, saber empezar a dividir. Para ello, se hace uso de la entropía y ganancia de información.

La entropía, en lo que se refiere en los árboles de decisión, se basa en saber si los datos son homogéneos o no. Si lo son, significará que la entropía es 0. Pero si de lo contrario, los datos son heterogéneos la entropía aumentará, por ejemplo, al 50% la entropía sería 1.

La ganancia de información es la disminución de la entropía después de que un conjunto de datos se divide en función de un atributo. La construcción de un árbol de decisión trata de encontrar un atributo que devuelva la mayor ganancia de la información. De esta manera, el árbol se construirá según el resultado obtenido, y con cada atributo se realizará el mismo proceso.

A continuación, se presenta un ejemplo de árbol de decisión que podría crease para determinar si hay un ataque de robo el *password* de un usuario.

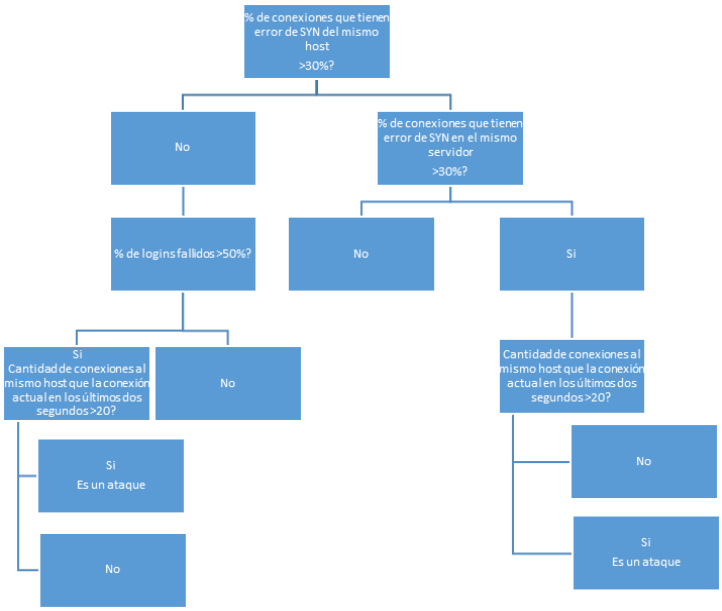

Fig. 3.2 Ejemplo Árbol de decisión

3.2.2 Naïve Bayes Clasification

Es un algoritmo de aprendizaje automático supervisado de clasificación. Se trata de un clasificador probabilístico basado en el teorema de Bayes con suposiciones de independencia entre las características relacionadas con un conjunto de datos en particular.

Se trata del algoritmo de clasificación más sencillo y rápido, y apropiado para una gran cantidad de datos. Se utiliza en diversas aplicaciones, como el filtrado de spam, la clasificación de texto, el análisis de sentimientos, los sistemas de recomendación o un detector de intrusiones en seguridad. Estos clasificadores son altamente escalables y requieren una serie de parámetros lineales en el número de variables. También requiere una pequeña base de datos para fines de capacitación. Además, este algoritmo no es sensible a características irrelevantes.

Cada vez que se realiza una clasificación, el primer paso es saber cuál es el problema e identificar características y etiquetas posibles. Podemos definir a las características como aquellas que afectan los resultados de la etiqueta.

La clasificación tiene dos fases:

- La fase de aprendizaje: es la fase donde el clasificador entrena su modelo en un conjunto de datos determinado
- La fase de evaluación: es la fase donde se prueba el rendimiento del clasificador. El rendimiento se evalúa en función de diversos parámetros, como precisión, error y recuperación.

El rendimiento se evalúa en función de diversos parámetros, como precisión, error y recuperación.

El clasificador Naïve Bayes supone que el efecto de una característica particular en una clase es independiente de otras características. Dicho esto, la ecuación es la siguiente:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- P(A|B) sea la probabilidad posterior. Este término se refiere a la probabilidad de A dados los datos de B
- P(B|A) sea la probabilidad de los datos B dado que A sea cierto.
- P(A) es probabilidad de que el resultado A sea verdadero
- P(B) predictor probabilidad previa, es decir, la probabilidad de los datos

El clasificador Naive Bayes calcula la probabilidad de un evento según los siguientes pasos si sólo hubiese una característica:

1. Calcular la probabilidad previa para las etiquetas de clase dadas.
2. Encontrar la probabilidad de la probabilidad con cada atributo para cada clase.
3. Poner los valores en la fórmula de Bayes y calcular la probabilidad posterior.
4. Ver qué clase tiene una probabilidad más alta, siendo que la entrada pertenece a la clase de probabilidad más alta.

Si hay más características se asocian un vector de características junto a las clases. Por ejemplo:

[X1, X2, X3,.., Xn] [C1]
[X1, X2, X3,.., Xn] [C4]

Donde los valores de X serían las características obtenidas y las C serian a que clase se asocia. Cada característica tiene voz para determinar qué etiqueta debe asignar el clasificador. El clasificador es el argumento máximo de probabilidad más alta por lo que A = argmax P(A|B). Se realiza para cada clase i para cada secuencia de documento.

El algoritmo utiliza la frecuencia de cada etiqueta en el conjunto de entrenamiento. Las frecuencias son sacadas del conjunto de datos.

Con este algoritmo se consigue que cada distribución se pueda estimar de forma independiente como una distribución dimensional. El algoritmo es mejor en cuanto a la velocidad y memoria que necesita. En contra, el algoritmo puede tener peores resultados en cuanto a la exactitud. Es un algoritmo que puede resultar muy útil cuando hay variables independientes y para comparar probabilidades. Este algoritmo puede ser utilizado para detectar anomalías en el sistema.

3.2.3 Isolation forest

Es un algoritmo de aprendizaje automático no supervisado que selecciona características al azar y selecciona un valor entre el máximo y el mínimo para esa característica seleccionada.

El algoritmo comienza específicamente creando árboles de decisión aleatorios, y luego la puntuación se calcula al ser igual a la longitud de la ruta para aislar la observación. Este algoritmo es usado para detectar anomalías en seguridad. Además, es un algoritmo con una baja complejidad de tiempo lineal y un requisito de memoria poco elevado.

El algoritmo aísla las observaciones seleccionando aleatoriamente una característica, y posteriormente seleccionando aleatoriamente un valor dividido entre los valores máximo y mínimo de la característica seleccionada. Dado que aislar puntos normales de anomalías requiere más cómputo, se puede utilizar un puntaje de anomalía que mide el número de condiciones necesarias para separar una observación dada.

La detección de valores atípicos, basada en el histograma de este algoritmo, es un método que puntúa los registros en tiempo lineal. Esto es debido a que asume la independencia de las características, lo que lo hace mucho más rápido que los enfoques multivariados, pero menos preciso.

3.2.4 Algoritmos de clustering

Este tipo de algoritmo es de aprendizaje automático no supervisado. Clustering es la tarea de agrupar un conjunto de objetos tales que los objetos en el mismo grupo. Este tipo de algoritmos permiten clasificar un conjunto de elementos en un determinado número de grupos. Se basan en si son más similares o diferentes entre sí que a los de otros grupos.

El factor de valor atípico local basado en clúster (CBLOF) utiliza clústeres para encontrar puntos de datos anómalos midiendo la desviación local de un punto dado con respecto a sus vecinos. Específicamente, el CBLOF utiliza el concepto de densidad local de k vecinos más cercanos al comparar las densidades de un objeto con las de sus vecinos para identificar regiones de densidad similar.

3.3 Machine Learning en la seguridad

Con el Machine Learning se pueden abarcar muchos de los problemas que se presentan dentro de la seguridad.

Los ciberataques son reconocidos como principales amenazas en el mundo digital para empresas y organizaciones gubernamentales. Como, por ejemplo, el ransomware, el malware, el phising o el acceso no autorizado, son ataques que causan una pérdida económica a las víctimas.

Los datos en las empresas pueden ser de los recursos con más valor del que dispone la empresa. Es por ese motivo que protegerlos es de gran importancia. El no hacerlo y ser víctima de un ataque puede suponer una pérdida para la empresa muy grande en valores económicos.

Los ataques de hoy en día cada vez son más sofisticados y difíciles de detectar, por ese motivo, el software actual contra muchos ataques puede no ser suficiente para proteger los datos. Por ese motivo la Inteligencia Artificial en ciberseguridad puede ayudar a detectar amenazas como la denegación de servicio (DDoS), *smurf,* …

La IA permite analizar los comportamientos en la red acelerando así los tiempos de respuesta en un ataque y disminuir así el impacto que pueda tener. De la misma manera también se puede llegar a prevenir sin que el ataque se haga efectivo debido al análisis de datos para encontrar patrones y predecir y detectar así una amenaza. Con los patrones que se puedan encontrar se puede detectar malware en el tráfico encriptado o por ejemplo proteger los datos debido a un comportamiento sospechoso en un usuario.

Hay empresas como DarkTrace que a través de la IA y el ML combaten los ciberataques. En este caso, tal y como indica DarkTrace [1] “emplea el aprendizaje de máquina sin supervisión para analizar datos de la red a escala y ejecuta miles de millones de cálculos basados en probabilidades en función de evidencias tangibles. En lugar de basarse en los conocimientos adquiridos de amenazas anteriores, clasifica de forma independiente los datos y detecta patrones convincentes”.

A continuación, podemos ver ejemplos de los diferentes tipos de problemáticas en seguridad abarcadas dentro del Machine Learning según sus modelos:

Supervisados

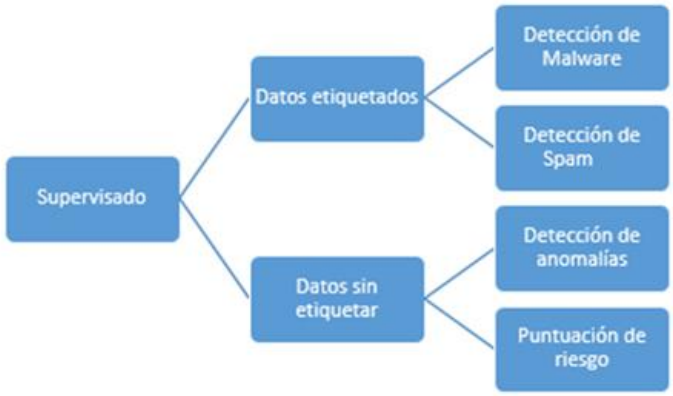

Fig. 3.3 Esquema modelo supervisado con algoritmos y aplicaciones en seguridad [2]

No supervisados

Fig. 3.4 Esquema modelo supervisado con algoritmos y aplicaciones en seguridad [2]

De los diferentes problemas de seguridad que nos podemos encontrar, se comentaran algunos de ellos, como se enfocan y tratan con el Machine Learning.

3.3.1 Amenazas en la red

En los últimos años, la importancia de la ciberseguridad y el aprendizaje automático se ha disparado. Hay nuevos sistemas que dependen de herramientas inteligentes que proporcionan el cómputo del siguiente nivel y sistemas abiertos a violaciones de seguridad. Estos sistemas han hecho que la importancia del aprendizaje automático se quiera utilizar para el análisis de datos de ciberseguridad.

La ciberseguridad es la práctica de defensa de la red de una organización y los datos de posibles atacantes, donde estos no tienen acceso autorizado a dicha red. A veces, la carga de datos desiguales de ciberseguridad de una variedad de fuentes diferentes, dificulta el desarrollo de una herramienta que haga que el aprendizaje automático sea más eficaz y preciso a la hora de mejorar los datos de ciberseguridad.

El ML permite detectar amenazas en la red dado que puede ir analizando los datos que hay y por consecuente buscar y detectar anomalías, comportamientos extraños en la red que pueden derivar a un ataque.

En el ML se procesan grandes cantidades de datos permitiendo predecir incidentes críticos. A partir de los patrones de tráfico, las conexiones, la actividad del usuario y aspectos de la red se pueden aprender y tomar decisiones con el ML. Las principales son la detección de malware, violaciones en políticas, amenazas internas o detección de intrusos.

Ya en 2016 los investigadores del Laboratorio de Ciencias de la Computación e Inteligencia Artificial (CSAIL) del MIT y la *startup* de aprendizaje automático PatternEx mostraban una plataforma de inteligencia artificial llamada AI^2 [3] que predecía los ataques cibernéticos significativamente mejor que los sistemas que habían al incorporar continuamente datos de expertos humanos.

Ahora en 2019 están desarrollando conjuntamente los investigadores del MIT y la Universidad de California en San Diego (UCSD) un nuevo sistema de aprendizaje automático para poder identificar aproximadamente 800 redes sospechosas de secuestradores en serie de IPs [4]. El equipo entrenó a su sistema, y descubrió que algunas de ellas habían estado secuestrando direcciones IP durante años.

Muchas tareas que realizaban los equipos de seguridad ahora se pueden automatizar.

Por otro lado, también podemos observar que, en Argentina, el Ministerio de Defensa de cara al Foro Internacional para la Cooperación Económica G20 que “se realizará en el país y a la que asistirán los principales jefes de Estado de mundo, entre ellos Donald Trump, Vladimir Putin, Xi Jinping y Angela Merkel.” Tienen como objetivo incorporar IA y ML para “reconocer anormalidades en las redes y detectar su origen con ayuda de un sistema que colaborará con el funcionamiento conjunto de las Fuerzas.” [5]

A continuación, se listan algunos de los problemas en seguridad que puede abarcar y abarca el ML:

3.3.2 Mantener a las personas seguras cuando naveguen

El ML puede predecir vecindarios malos en línea que permiten proteger a un usuario de conectarse a sitios web maliciosos dado a que analiza la actividad en internet.

3.3.3 Proporcionar protección contra el malware de punto final

El ML permite que haya algoritmos que puedan detectar comportamientos extraños en la web que puedan derivar en malware. Según la actividad, archivos y comportamientos permiten detectar malware en la red.

3.3.4 Proteger datos en la nube

Una de las prácticas para proteger los datos en la nube es restringir el acceso sólo a usuarios autorizados.

Otra de las prácticas, combinable con la anterior, sería limitar al acceso a ciertos elementos de la red y no el acceso total. Se puede limitar por usuarios, por servicios, por servidores, etc.

Con el ML se puede analizar la actividad sospechosa de los inicios de sesión de la aplicación en la nube. Entre las actividades más destacadas serían: detectar anomalías basadas en la ubicación y realizar análisis de reputación de IP para identificar amenazas y riesgos en las aplicaciones y plataformas en la nube.

Azure dispone de una plataforma para realizar ML y dispone de diferentes módulos a utilizar. Por ejemplo, el de detección de anomalías. Dicha plataforma ofrece los módulos de máquina de vectores de soporte de una clase y el módulo de detección de anomalías basada en PCA [6]. Con estos módulos se puede empezar a trabajar con el modelo, definir parámetros y posteriormente entrenarlo utilizando datos etiquetados.

3.3.5 Detectar malware en el tráfico encriptado

El ML puede detectar malware en el tráfico encriptado mediante el análisis de elementos de datos de tráfico encriptados en la telemetría de red común. Los algoritmos de ML identifican patrones maliciosos en lugar de descifrarlos. De esta manera pueden encontrar amenazas ocultas en el cifrado.

3.3.6 Spam filters

A través de ML se pueden filtrar los mensajes que pueden contener SPAM o phising. Google dice que su tecnología de aprendizaje automático ahora bloquea el 99.9% de los mensajes de *spam* y *phishing* de Gmail

Esto es debido a TensorFlow, un marco de aprendizaje automático de código abierto (ML) desarrollado por Google. Llega a bloquear alrededor de 100 millones de mensajes de spam adicionales cada día. El procedimiento se basa en identificar patrones en grandes conjuntos de datos que las personas que crean las propias reglas podrían no detectar. Las protecciones basadas en ML ayudan a tomar decisiones granulares basadas en muchos factores diferentes.

TensorFlow es una plataforma de código abierto de extremo a extremo para el Machine Learning [7]. Esta plataforma facilita la creación e implementación de modelos ML sin importar el idioma o plataforma que se utilice.

Google también está utilizando el ML para analizar las amenazas contra los puntos finales móviles que se ejecutan en Android, así como también identifica y elimina el malware de los teléfonos infectados.

3.3.7 Detección estática de archivos PE maliciosos

Los archivos ejecutables portátiles (PE) pueden contener malware. Para reducir el riesgo de que se ejecute y haya malware, se debe detectar el archivo malicioso antes de que se llegue a ejecutar y evitar así que el malware no contamine el sistema.

Para este caso encontramos que hay un estudio realizado por Yasmin Bokobza y Yosef Arbiv que consta de realizar un algoritmo que permita la detección de malware en archivos PE. Lo realizan a través de un algoritmo con modelo de árbol de decisión donde tienen en cuenta el estado de la firma y las categorías de importación de DLL según la máxima entropía que tengan. Primeramente, aprovechan las características encontradas en archivos PE que sean maliciosos para entrenar modelos que distingan entre software benigno y malicioso.

Una vez realizan la recopilación de datos, almacenan todos los archivos de muestra de malware en un entorno seguro y extraen las mismas características de archivos PE maliciosos y benignos. Después utilizan una metodología conformada por tres ciclos: aprendizaje, extracción y detección de características. Al final de cada ciclo, los resultados, verificados por analistas humanos, verificados como maliciosos se utilizan como para mejorar el clasificador para el ciclo posterior.

En este estudio, el proceso de aprendizaje consta de la entrada de características de archivos PE maliciosos y benignos, utilizado conjuntamente con el algoritmo Random Forest y con los comentarios realizados por los analistas de los resultados verificados. En el proceso de detección se utilizarían

el algoritmo de clasificación de ML con los parámetros aprendidos en el proceso de aprendizaje y las características extraídas de archivos PE aprendidos. De este proceso es de donde provienen los comentarios de los analistas de los resultados verificados. [8]

3.4 Detección de Anomalías

Las infracciones de seguridad y las amenazas están creciendo junto con el campo de la ciberseguridad. El gran avance y el rápido crecimiento en Internet y las redes ha incrementado de manera continuada y la tendencia es de seguir aumentado. De la misma manera los piratas informáticos y atacantes crecen y desarrollan técnicas para evitar ser detectados y llevar a cabo ataques informáticos con un fin.

Esto provoca daños en recursos y puede provocar grandes pérdidas económicas. Para evitar algunos de estos ataques se utiliza una herramienta llamada sistema de detección de intrusos como última línea de defensa contra intrusos que pueden tener acceso no autorizado al sistema. El sistema de detección de intrusos puede garantizar la continuidad o minimizar el impacto en los servicios afectados.

El problema de detección de anomalías ha sido ampliamente estudiado en la seguridad informática. El volumen de datos desiguales de ciberseguridad dificulta la creación de una herramienta con ML que sea más eficaz y precisa. Se han realizado muchos estudios que utilizan técnicas de ML para la detección de intrusos, pero algunos muestran una detección deficiente y algunos métodos requieren más tiempo de entrenamiento. La mayoría de los proyectos de detección actuales dependen de la selección de características de los paquetes IP capturados. En este campo faltan sistemas de extremo a extremo que puedan clasificar automáticamente las anomalías en los datos.

Una medida para determinar en qué medida un usuario en particular tiene este tipo de acceso no autorizado es detectar anomalías en los datos de tráfico de la red, por ejemplo, BRO, YAF, PCAP, SNORT. Estas son herramientas que permiten detectar anomalías, pero sin aprendizaje automático:

- YAF, o Yet Another Flowmeter, es un tipo de datos de ciberseguridad que procesa datos PCAP y exporta estos flujos al proceso de recopilación de IPFIX
- BRO es un marco de código abierto que analiza el tráfico de red y se utiliza para detectar anomalías en una red
- SNORT es un sistema de prevención de intrusiones de código abierto capaz de análisis de tráfico en tiempo real y registro de paquetes

Una plataforma potencial que pudiera detectar anomalías necesitaría procesar datos en tiempo real y luego usar un modelo que use los datos cogidos para saber si los datos actuales contienen anomalías y, por lo tanto, si la red tiene una intrusión.

El alcance de la detección de anomalías abarca no solo violaciones por parte de un extraño, sino también anomalías derivadas de violaciones por parte de un usuario autorizado. Es importante tener en cuenta que la detección de

anomalías omite la clase de violaciones de la política de seguridad que ocurren dentro de los límites del comportamiento normal de un sistema o sitio.

La detección de comportamientos anómalos se puede ver como un problema de clasificación de valores binarios en qué mediciones de la actividad del sistema, como los archivos de registro del sistema, el uso de recursos, los seguimientos de comandos y los seguimientos de auditoría se utilizan para producir una clasificación del estado del sistema como normal o anormal.

Dentro de la detección de anomalías podemos encontrar los ataques de denegación de servicio (DDoS). Los ataques de DDoS que se realizan en la capa de aplicación y utilizan poco ancho de banda. Los ataques más comunes en la dicha capa son: ataque por inundación de ampliación de DNS, el ataque de inundación SIP y de inundación HTTP.

Como ejemplo en ataque de inundación DDoS podríamos mencionar el ocurrido en febrero de 2000, afectando en el servicio prestado por la compañía YAHOO, donde hubo una desconexión de aproximadamente dos horas. Este ataque provocó grandes pérdidas en sus ingresos. Podemos observar que estos ataques no cesan, sino que cada vez son más sofisticados. Podemos ver que en este año 2019 un importante ataque de denegación de servicio distribuido (DDoS) a AWS dejó fuera de servicio su servicio S3 y otros servicios durante ocho horas.

A continuación, se explican una serie de artículos y documentación sobre la detección de anomalías donde se hace uso del Machine Learning.

3.4.1 An Application of Machine Learning to Anomaly Detection

En este artículo realizado por Terran Lane and Carla E. Brodley en la escuela *Electrical and Computer Engineering* se presenta un enfoque de aprendizaje automático para la detección de anomalías [8].

El sistema que presentan crea perfiles de usuario basados en secuencias de comandos y compara las secuencias de entrada actuales con el perfil utilizando una medida de similitud. El sistema debe aprender a clasificar el comportamiento actual como coherente o anómalo con el comportamiento pasado utilizando solo ejemplos positivos del usuario válido de la cuenta. Los resultados empíricos que obtienen demuestran que este es un enfoque prometedor para distinguir al usuario legítimo de un intruso.

En este artículo, presentan un enfoque de aprendizaje automático para la detección de anomalías diseñado para que aprenda de un perfil de usuario y posteriormente lo emplee para detectar comportamientos anómalos. Basado en secuencias de acciones (comandos UNIX) del flujo de entrada del usuario actual, el sistema clasifica el comportamiento actual como consistente o anómalo con el comportamiento pasado.

El artículo primero explica como aprender del perfil de un usuario: la recopilación de los datos para formar un usuario perfil y posteriormente el sistema de recolección de datos. Posteriormente abarca la detección del comportamiento anómalo y la secuencia de cálculo de similitud y como clasificar dicho comportamiento. Finalmente realizan el análisis empírico de los datos y el método experimental para acabar con los resultados del comportamiento que un usuario que podría ser aprendido y el comportamiento anómalo que pueda detectase mediante el uso de secuencias características.

El primer paso sería el aprendizaje de un perfil de usuario. Para que el sistema de detección reconozca el comportamiento anómalo, primero debe formar un perfil de usuario para caracterizar el comportamiento normal. En esta sección se describe el modelo que se utilizará para la creación de perfiles de usuario y donde analizan los detalles de implementación de cómo se forman los perfiles a partir de los datos de comandos.

Para la recopilación de patrones característicos de las acciones de los usuarios el sistema utiliza una secuencia, un conjunto ordenado de longitud fija de acciones temporalmente adyacentes, como la unidad fundamental de comparación.

Para facilitar la recopilación de datos, mantuvieron el orden temporal de los comandos dentro del contexto de un solo intérprete de comandos. El sistema de detección de anomalías lo visualizan como un asistente de software personal (un agente) que ayuda a monitorizar las penetraciones en la cuenta de un usuario. El comportamiento del usuario únicamente se caracteriza a partir de ejemplos positivos. Esto se debe a problemas de privacidad y a la dificultad para caracterizar el espacio completo de los comportamientos de los usuarios, solo se pueden obtener ejemplos positivos del comportamiento del propietario de la cuenta.

Por ese hecho explicado, el clasificador que utilizan es un clasificador con etiquetas donde todas las entradas son positivas. En consecuencia, todo lo que no se ve en los datos históricos representa un usuario diferente.

Para recopilar datos de acción del usuario, crean un analizador para la familia de lenguajes csh de UNIX (incluido tcsh) que traduce el flujo de datos sin procesar del seguimiento del comando de *shell* en un flujo de *tokens* adecuado para el almacenamiento y la comparación. Esta traducción les permite suprimir los nombres de archivo pero conserva los nombres de los comandos, los argumentos de cambio y otros símbolos sintácticamente importantes como `|,; y> & !.`

Por ejemplo, la secuencia:

```
> ls -laF
> cd /tmp
> gunzip -c foo.tar.gz | (cd \. ; tar xf -)
```

Se vería traducida como:

```
ls -laF cd <1> gunzip -c <1> | ( cd <1> ; tar - <1> )
```

Dónde el token <1> denota la aparición de un solo argumento de nombre de archivo. El analizador también presenta los *tokens* ** SOF ** y ** EOF ** que indican el inicio y el final de una sesión de intérprete de comandos, respectivamente.

Durante el entrenamiento, la secuencia de *token* procesada se almacena textualmente en el diccionario. El diccionario es una base de datos de instancia (secuencia) que, junto con una medida de similitud y un conjunto de parámetros del sistema (descritos a continuación), constituyen el perfil de un usuario.

El siguiente paso sería detectar un comportamiento anómalo. En este paso utilizan la unidad fundamental de comparación en el sistema detector de anomalías, que sería la secuencia de comandos. Para clasificar las secuencias de nuevas acciones como consistentes o inconsistentes en el historial de secuencias, todas las secuencias de tokens de entrada se segmentan en secuencias superpuestas de tokens. La longitud de cada secuencia es un parámetro para el sistema, pero se fija para una sola ejecución. Luego se pueden comparar dos secuencias de longitud fija utilizando una medida de similitud.

La acción básica del sistema de detección es comparar las secuencias de entrada entrantes con los datos históricos y formar una opinión sobre si ambos representan o no al mismo usuario según su perfil.

Posteriormente, la secuencia informática similar trata de una función de igualdad que según si coinciden o no en cada posición devuelve el valor TRUE o FALSE. Esta es la función de similitud empleada por los algoritmos de coincidencia de cadenas y tiene la ventaja de ser ampliamente estudiada y altamente optimizadle

Al depender de la variabilidad humana, para secuencias largas, la probabilidad de localizar coincidencias exactas en los datos de comandos históricos se vuelve extremadamente baja. Por lo tanto, en este caso, la función de igualdad no es una opción viable.

El sistema calcula una medida de similitud numérica que devuelve un valor alto para pares de secuencias que cree que tienen un parecido cercano, y un valor bajo para pares de secuencias que cree que difieren en gran medida. La medida de similitud se basa en la intuición de que las coincidencias de *tokens* separadas por *tokens* intercaladas tienen más probabilidades de haber ocurrido por casualidad, mientras que las coincidencias adyacentes tienen más probabilidades de haber sido causadas por un proceso causal.

Por lo tanto, si la secuencia Seq1 tiene k *tokens* en común con cada una de Seq2 y Seq3, pero los *tokens* comunes son adyacentes en Seq1 y Seq2

entonces la medida de similitud que se quiere que se tenga es la propiedad Sim (Seq1; Seq2)> Sim (Seq1; Seq3).
La medida de similitud asigna puntajes de similitud, Sim (Seq1; Seq2) de la siguiente manera:

- Se establece un contador de adyacencia, c: = 1 y el valor de la medida, Sim: = 0.
- Para cada posición, i, en la longitud de la secuencia:
  - Si Seq1(i) = Seq2(i) entonces Sim = Sim + c e incrementa c en 1.
  - De lo contrario, c: = 1.
- Después de examinar todas las posiciones, se devuelve el valor de la medida.

Esta medida produce una puntuación más alta para secuencias más similares, delimitadas entre 0 y n (n + 1) = 2 (donde n es la longitud de la secuencia) y sesgadas hacia *tokens* idénticos adyacentes en lugar de *tokens* idénticos separados por algunos *tokens* intermedios que no coinciden.

Eligen un límite superior polinómico para nuestra medida de secuencia basado en la observación de que los elementos en una secuencia de comando no son independientes. Por lo tanto, el par de secuencias que se muestran a continuación a la izquierda tendría un valor de similitud mayor que el par en el derecho.

```
ls <1>; vi   ls -l <1>;

ls <1> cat   <3> ls -a <1> cat
```

Definen la similitud de una secuencia única Seq_i con un conjunto de secuencias L como:

$$\mathrm{Sim}(Seq_i, L) = \max_{Seq_j \in L} \{\mathrm{Sim}(Seq_i, Seq_j)\}$$

Por lo tanto, la similitud de una secuencia con el diccionario del usuario es la medida de esa secuencia en comparación con la secuencia más similar en el diccionario.

Otro punto a tener en cuenta es la clasificación del comportamiento de un usuario. Dado un flujo de entrada de *tokens* de comando analizados por el módulo de recopilación de datos, el módulo de detección clasifica al usuario actual como normal o anómalo después de cada *token*. La salida del módulo de detección es una secuencia de decisiones binarias que indican, en cada punto de los datos del comando de entrada, si cree o no que la secuencia de entrada en ese punto fue generada por el usuario perfilado. Los parámetros del sistema del estudio son: la longitud de secuencia, longitud de ventana, umbral de clasificación, tamaño de diccionario.

Para tomar estas decisiones, el módulo de detección primero calcula la similitud de cada secuencia de entrada con el diccionario del usuario,

produciendo un flujo de medidas de similitud. En un sentido intuitivo, este flujo representa la familiaridad de los comandos de entrada en cada paso de tiempo, dado el conocimiento sobre el comportamiento anterior del usuario.

Si bien las secuencias individuales pueden desviarse del precedente histórico, el comportamiento agregado debe ajustarse en gran medida al comportamiento histórico para usuarios válidos, pero aún debe desviarse notablemente para los intrusos, aplican un filtro de suavizado a los datos.

Según el análisis empírico el sitio y la política de seguridad, las falsas alarmas pueden interrumpir el trabajo de los administradores o usuarios del sistema. Para determinar los falsos positivos y falsos negativos la precisión de un sistema de detección de anomalías debe ser alta. En el estudio consideran que el umbral de detección del sistema de clasificación también debe ser un parámetro configurable. Esto se debe a que, si una empresa prefiere aceptar una tasa más alta de falsas alarmas para obtener una tasa más alta de detecciones de intrusiones reales al sistema, esta puede definir su umbral.

Este experimento lo diseñaron para probar la hipótesis de que el comportamiento del usuario podría ser aprendido y el comportamiento anómalo detectado mediante el uso de secuencias características.

Los resultados demuestran que el sistema de reconocimiento tiene una detección verdadera más alta que las tasas de falso positivo. El perfil de comportamiento basado en secuencias es útil para algunos usuarios del dominio utilizado en el estudio.

Por otro lado, comentan que es posible que un único tamaño de diccionario sea aplicable a todos los usuarios. Alternativamente, es posible que se necesiten diferentes tamaños de diccionario para un rendimiento superior para diferentes usuarios. El problema viene porque parece aumentar la tasa de falsos positivos según se incrementa el tamaño del diccionario. Por estas razones parece preferible mantener el diccionario más pequeño por razones de precisión según los resultados que obtuvieron en el experimento.

Además, el tamaño óptimo del diccionario, según los resultados, parece que deberá de realizarse en función del usuario perfilado. Si el sistema de detección de anomalías debe ejecutarse en tiempo real, entonces es imperativo que el sistema sea rápido y conservador de recursos. Esto implica que gran parte de los datos disponibles deben descartarse con poco o ningún examen.

Usuarios maliciosos informados

Uno de los desafíos a los que se pueden enfrentar es que un usuario tenga conocimiento de las defensas del sistema, incluido el sistema de detección de anomalías y sus perfiles de usuario. Por lo que es un desafío distintivo el detectar a un intruso bien informado con los sistemas de detección de anomalías. El invasor puede intentar conformar su comportamiento al

registrado en un perfil para un usuario válido, evitando así que el sistema de detección de anomalías lo notifique.

Los perfiles basados en secuencias de comportamiento son difíciles de emular con éxito un usuario malintencionado ya que puede subvertir el entreno del sistema de detección, inicialmente conforme al comportamiento esperado, pero cambiando gradualmente a un comportamiento malicioso, de tal manera que no parezca sospechoso.

Para líneas futuras podría realizarse un estudio de un usuario normal, pero que según pasa el tiempo, el usuario cambia sus acciones, que pueden ser debidas a diferentes razones: el uso de diferentes aplicaciones o que puedan leer el manual de UNIX. Esto significa que algunos de los datos de la secuencia anterior ya no reflejarán con precisión el comportamiento del usuario. Se necesita un método para eliminar secuencias de datos desactualizadas del diccionario para realizarlo más preciso según pasa el tiempo. Al mismo tiempo, se debe tener en cuenta la amenaza de la formación hostil del sistema. Una dirección para el trabajo futuro será examinar la compensación entre la adaptación al cambio legítimo y la protección contra la capacitación hostil.

3.4.2 Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM Project

A partir del estudio *Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM project realizado* por Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, y Philip K [10]. Chanque utilizan el ML para la detección de fraude e indican que las técnicas que emplean pueden generalizarse y aplicarse al área importante de detección de intrusiones en los sistemas de información en red.

La UCI KDD, Información de archivo y ciencias de la computación de la Universidad de California, Irvine, utilizaron el conjunto de datos de KDD Cup 1999 Data para realizar la detección de intrusos. Tal y como indican, "*Este es el conjunto de datos utilizado para la Tercera Competencia Internacional de Descubrimiento de Conocimiento y Herramientas de Minería de Datos, que se llevó a cabo junto con KDD-99 La Quinta Conferencia Internacional sobre Descubrimiento de Conocimiento y Minería de Datos."* Irvine, CA 92697-3425, Edureka [11].

Este *dataset* contiene un conjunto de datos que constan principalmente de intrusiones, ataques, conexiones normales o conexiones malas. Estos ataques se pueden dividir en cuatro categorías principales:

- DOS: denegación de servicio, por ejemplo, inundación de sincronización;
- R2L: acceso no autorizado desde una máquina remota, por ejemplo, adivinar contraseña;

- U2R: acceso no autorizado a privilegios de superusuario local (raíz), por ejemplo, varios ataques de “desbordamiento de búfer “, sondeo, vigilancia y otros sondeos, por ejemplo, escaneo de puertos.

Comentan las características de un software para detectar intrusiones en la red, para proteger la red de usuarios no autorizados o de personas con información privilegiada. El objetivo de aprendizaje del detector de intrusiones es construir un modelo predictivo, un clasificador, capaz de distinguir entre intrusiones o ataques, y conexiones normales.

3.4.3 Mecanismo de defensa para el ataque DDoS a través del Machine Learning

En el artículo DEFENSE MECHANISM FOR DDoS ATTACK THROUGH MACHINE LEARNING, realizado por Sujay Apale, Rupesh Kamble, Manoj Ghodekar, Hitesh Nemade, Rina Waghmode del departamento de ingeniería de computación de AISSMS COE, Pune, India [12]. Este estudio enfatiza los ataques de inundación DDoS de la capa de aplicación ya que estos ataques son cada vez más comunes. Se numeran diferentes problemas en cuanto a las diferentes visiones en la detección de intrusiones. Para combatirlos, proponen el uso del algoritmo clasificador Naïve Bayes en ML.

Mencionan que según una encuesta realizada el algoritmo Naïve Bayes (NB) proporciona una velocidad de entrenamiento y aprendizaje más rápido que otros algoritmos de ML. Por lo que les puede permitir mejorar el tiempo requerido para entrenar IDS.

Este algoritmo tiene más precisión en la clasificación y detección de un ataque. Por este motivo desarrollan un sistema de detección de intrusos en la red (IDS) que utiliza un enfoque de aprendizaje automático con la ayuda del algoritmo NB.

- Describe la clasificación de ID
- Clasifica los diferentes tipos de ataques de inundación DDoS de la capa de aplicación
- Analizan algunos artículos de la literatura.
- Presenta el algoritmo Naïve Bayes, propone un sistema eficiente de detección de intrusos basado en la técnica de aprendizaje automático.

El enfoque que proponen es construir un modelo analítico para la detección de intrusos con una capacidad de aprendizaje más rápida que cualquier otro enfoque existente. Usando el método NB proponen un clasificador para diferenciar entre la actividad habitual y la inusual. Los resultados del algoritmo NB se compararán con el enfoque de detección de intrusos existente.

La arquitectura propuesta es la siguiente:

1.- Entrada de datos de la red
2.- Procesar datos en forma legible por máquina,
3.- Extraer las características requeridas y ponerlas en una base de datos
4.- Aplicar el algoritmo clasificador Naïve Bayes junto con la base de datos creada
5.- Predecir un usuario o atacante legítimo
6.- Bloquear si es un atacante o dejar que el usuario acceda al servicio. En este último paso se actualiza la base de datos.

Con esta arquitectura proponen utilizar el algoritmo de Naïve Bayes para el ML con el fin de mejorar el tiempo requerido para entrenar sistemas de detección de intrusiones.

3.4.4 CAMLPAD: Cybersecurity Autonomous Machine Learning Platform for Anomaly Detection

Este estudio, realizado en 2019 llamado CAMLPAD: Cybersecurity Autonomous Machine Learning Platform for Anomaly Detection realizado por Ayush Hariharan; del departamento de ciencias de computación de la Academia de Ciencias del condado de Loudon, EEUU; Ankit Gupta y Trisha Pal; del departamento de ciencias de la computación TJHSST de Alexandria, de EEUU [13].

CAMLPAD proviene de las siglas Cybersecurity Autonomous Machine Learning Platform for Anomaly Detection o la Plataforma de Ciberseguridad y Aprendizaje Automático de Máquina para la Detección de Anomalías.

El sistema CAMLPAD proporciona un enfoque preciso y racionalizado a la detección de anomalías de ciberseguridad en tiempo real, lo que resulta en una puntuación atípica, ofreciendo una solución novedosa que tiene el potencial de revolucionar el sector de ciberseguridad. CAMLPAD logró un puntaje de rand ajustado del 95 por ciento de precisión del sistema.

Las dos ideas que abarca la investigación son:

- El diseño de un sistema de extremo a extremo para la clasificación inteligente y automática de anomalías
- Los sistemas tradicionales usan técnicas de estadísticas elementales y a menudo son inexactas, lo que lleva a plataformas de análisis de datos centralizadas débiles

Los sistemas CAMLPAD se basan en el enfoque holístico, empiezan con la recuperación de una multitud de diferentes datos de ciberseguridad en tiempo real utilizando Elasticsearch y formateados en un cuaderno local, donde luego

ejecuta varios algoritmos de aprendizaje automático, para procesar los datos. Elasticsearch es un servidor que proporciona un motor de búsqueda de texto completo, distribuido y con capacidad de multitenant con una interfaz web RESTful y con documentos JSON.

El sistema holístico CAMLPAD incorpora el flujo de datos de Elasticsearch en tiempo real con el algoritmo de aprendizaje automático. El sistema que proponen determina de inmediato, según la presencia de anomalías, si un entorno particular está en riesgo inmediato de una violación de política de seguridad.

Una vez que se han procesado los datos y se han calculado las anomalías, el sistema CAMLPAD utiliza Kibana para visualizar datos atípicos extraídos de Elasticsearch y para medir qué tan alto es el puntaje atípico. Di el puntaje de anomalía alcanza un umbral particular, se envía una alerta a la organización, normalmente al administrador del sistema, quien tiene la opción de reenviar alerta a todos los empleados de la empresa para que sepan que se ha producido una infracción de ciberseguridad.

Se procesan una gran cantidad de datos de ciberseguridad, como YAF, BRO, SNORT, PCAP y Cisco Meraki utilizando diferentes modelos de aprendizaje automático. El sistema CAMLPAD utiliza los siguientes algoritmos de aprendizaje automático para procesar los datos:

- Isolation Forest
- Histogram-Based Outlier Score (HBOS), Detección de valores atípicos basados en histogramas
- Cluster-Based Local Outlier Factor (CBLOF), Factor de valores atípicos locales basados en clústeres
- K-Means Clustering, Agrupación de medias K
- Gaussiano multivariante

El modelo que utiliza el sistema es no supervisado. Una vez se ejecutan los modelos, estos dan como resultado una visualización de los datos y una puntuación atípica. De los diferentes modelos, las anomalías calculadas se visualizan con Kibana y es cuando se les asigna un puntaje atípico, que sirve como el indicador de si se debe enviar o no la alerta al administrador del sistema de que existen posibles anomalías en la red. Con cada algoritmo consiguen diferentes puntos de la detección de anomalías.

El factor de valor atípico local basado en clúster (CBLOF) utiliza clústeres para encontrar puntos de datos anómalos midiendo la desviación local de un punto dado con respecto a sus vecinos. Específicamente, el CBLOF utiliza el concepto de densidad local de k vecinos más cercanos al comparar las densidades de un objeto con las de sus vecinos para identificar regiones de densidad similar.

El algoritmo Histogram-Based Outlier Score (HBOS), realiza la detección de valores atípicos basados en histogramas, más en específico, utiliza el concepto de densidad local de k vecinos más cercanos al comparar las densidades de un objeto con las de sus vecinos para identificar regiones de densidad similar.

Los datos utilizados en el estudio describían varias transacciones en línea que ocurrían la sede donde cogían los datos. Los datos tenían que transferirse con precisión desde los sensores, que se ejecutaban en máquinas virtuales Linux, a un servidor local. Entonces es cuando el modelo puede procesar los datos y alertar al usuario si hay alguna anomalía presente.

Los datos se almacenan temporalmente localmente antes de cargarse en un servidor hadoop que consta de un nodo maestro y tres nodos esclavos. Posteriormente, se utiliza Apache NiFi (un proyecto de Apache Software Foundation, fue diseñado específicamente para automatizar el flujo de datos entre software sistemas) para racionalizar y procesar los registros del sensor antes de enviar la información procesada a la Kafka Queue. Los datos se transfieren desde el almacenamiento virtual en el servidor hadoop a la Kafka Queue, donde se pueden almacenar de manera más eficiente. Una vez que los datos han entrado en la cola, se envía a la base de datos de Elasticsearch, donde se almacena para su procesamiento futuro.

Kafka Queue Elasticsearch es una base de datos que analiza y normaliza los datos sin procesar antes de asignar a cada consulta de información un número de identificación único. Usando este número de identificación y el índice asociado con los datos, se puede consultar la información de los registros del sensor para su posterior procesamiento utilizando los modelos de aprendizaje automático. Sin embargo, una advertencia con Elastic Search es que no permite ejecutar scripts de procesamiento personalizados dentro de la base de datos.

Los algoritmos de aprendizaje automático utilizan la capacidad de indexación de la búsqueda elástica para transmitir datos a una máquina separada. Estos datos se transmiten directamente desde la base de datos, sin tener que descargar los datos como CSV o JSON, lo que significa que los datos se transfieren rápidamente del almacenamiento a un procesador local en otra máquina. Usando un algoritmo único donde los datos se indexan e importan en un marco de datos.

Una vez que se ha creado el marco de datos, los datos de los días actuales se indexan e importan a otro marco de datos. Este marco de datos contendrá la información más reciente utilizada para la detección de anomalías basada en patrones observados en los datos almacenados anteriormente.

Una vez tienen los datos importados con éxito a los respectivos marcos de datos, los datos categóricos presentes en los registros del sensor, como el tipo de solicitud o url, deben codificarse en valores numéricos antes de un análisis posterior por parte de los algoritmos de aprendizaje automático.

Después de codificar los datos, se utilizarán dos métodos de imputación: regresión lineal, para valores puramente numéricos, e inserción de relleno, para valores categóricos codificados. Ahora que los datos perdidos han sido imputados, los datos se pueden importar al modelo de conjunto personalizado para la detección de anomalías.

El modelo de conjunto personalizado consta de un algoritmo Isolation Forest, Histogram-Based Outlier Score (HBOS) y un Cluster-Based Local Outlier Factor (CBLOF). Una vez que los datos se ajustan al modelo general, los datos de validación y prueba se asignan a un valor atípico.

Puntuación

Según el puntaje de valores atípicos y un algoritmo PCA simple (el objetivo del PCA es reducir la dimensionalidad de un conjunto de datos que consta de muchas variables correlacionadas entre sí mientras se mantiene la variación presente en el conjunto de datos, hasta el máximo), los grupos se desarrollan dependiendo del puntaje de valores atípicos asignado por el modelo respectivo. Esos grupos se procesan y se crea un mapa que describe los diversos niveles de valores atípicos presentes en los datos. Este proceso se repite para cada modelo creado, lo que resulta en tres mapas de calor que describen los puntajes atípicos asignados por cada modelo para los datos.

Después de que se hayan asignado los puntajes atípicos, el modelo se crea a través de un sistema de votación democrático, donde cada modelo tiene el mismo dictamen.

Un punto de datos es un valor atípico o un valor interno. Una vez que se ha completado el sistema de votación, las puntuaciones atípicas finales se ejecutan a través del algoritmo PCA y se crea un mapa de calor final. El proceso se repite para los diferentes tipos de datos que se almacenan en la base de datos de Elasticsearch, incluidos YAF, BRO, SNORT y Meraki. Específicamente, los datos BRO se dividen por protocolo en DNS y CONN para etiquetar los datos con precisión.

Una vez que los puntajes atípicos finales se han compilado para cada tipo de datos, se crea un modelo de conjunto final, utilizando un sistema de votación democrático para reclasificar cada punto de datos. Este modelo final toma en consideración, no solo diferentes modelos de detección de valores atípicos que han tenido éxito en investigaciones anteriores, sino también diferentes tipos de datos de sensores, que capturan diferentes capas de tráfico de Internet.

Una vez creado el modelo, la precisión se determina calculando la puntuación RAND ajustada, un método común para evaluar algoritmos de aprendizaje

automático no supervisados. Esto representa la última parte del flujo de trabajo, donde los datos originados de diferentes sensores se han procesado de manera efectiva y se ha completado el escaneo de anomalías.

Después de que la precisión ha sido probada y confirmada, los puntajes atípicos recién asignados se indexan y se vuelven a consultar en la base de datos Elasticsearch para que se puedan crear visualizaciones de estos puntajes atípicos en Kibana. Específicamente, el índice se representa como un indicador, donde se compara el puntaje atípico de los datos de los días actuales con los datos anteriores sobre los que el modelo ha entrenado.

Cuando el indicador supera el 75%, se envía una alerta personalizada al administrador de la base de datos de Apache, que le alerta de que puede haber anomalías en los datos actuales. Después, el usuario tiene la opción de responder a esta alerta bloqueando ciertos puertos de destino o direcciones IP o puede realizar una investigación adicional para determinar la causa de las anomalías.

En términos de resultados, el sistema CAMLPAD consta de cinco componentes de datos principales: BRO DNS, BRO CONN, YAF, SNORT y Meraki. Estos componentes, junto con los tres modelos: lsolation Forest, CBLOF y HBOS, se combinan en un sistema de votación democrático para determinar el puntaje de valor atípico final.

Aunque el usuario no recibe alertas en función de cada tipo de datos individual y solo del modelo combinado final, en el estudio se observan cómo los diferentes tipos de datos muestran patrones similares en la detección de anomalías. En todos los mapas, hay dos puntos de datos separados, datos previos en los que el modelo entrenó y los datos de días actuales. Muestran unos datos donde las diferencias o similitudes entre el día actual y los datos anteriores se pueden observar junto con otros patrones que representan el nivel de anomalías presentes.

Los datos DNS de BRO se procesaron mediante tres algoritmos principales: HBOS, CBLOF e Isolation Forest. Además, el algoritmo de votación individual, aislado para los datos DNS de BRO, lo implementaron junto con las visualizaciones de Kibana, donde representan los datos del día reciente y los datos de los días anteriores.

Basado el resultado con HBOS hubo más valores atípicos y para el día actual tuvo menos valores atípicos.

Para el resultado con CBLOF salió que el día actual tenía más valores atípicos que los días anteriores. Esto significa que el día actual probablemente fue un caso atípico.

Con el algoritmo de Isolation Forest representa menos de un valor atípico.

Para el resultado con el algoritmo HBOS de datos de BRO CONN revela que los datos del día actual son un poco atípicos en comparación con los datos de

días anteriores. Esto es corroborado por el algoritmo CBLOF, pero los algoritmos Isolation Forest y el algoritmo combinado ofrecen un resultado completamente diferente.

Para los datos de YAF los datos del día actual son atípicos cuando se combinan con los datos de los días anteriores. Por ejemplo, en el diagrama HBOS, hay puntos negros rodeados de puntos de luz. Esto se repite para CBLOF, Isolation Forest y el algoritmo combinado.

Para los datos SNORT, HBOS, CBLOF y el algoritmo combinado perciben que los datos del día actual son atípicos. Sin embargo, el algoritmo Isolation Forest no percibe que los datos del día actual sean atípicos.

Por último, para los datos combinados en cada uno de los diferentes algoritmos, los datos del día actual son atípicos cuando se combina con los datos de los días anteriores.

Con el fin de determinar la medida de precisión, utilizan el RAND Score para medir la similitud entre dos agrupaciones al considerar todos los pares de muestras y contar pares asignados. El puntaje RAND toma en cuenta los positivos reales, los falsos positivos y determina qué tan precisa es la medición para un día anterior a partir de los datos del día actual. Esto se usa porque los datos de días anteriores ya están validados, por lo que es más fácil usar los datos del día actual como un conjunto de capacitación.

Todos los algoritmos implementados alcanzan puntajes atípicos similares, por lo que es evidente que el algoritmo holístico es correcto, ya que ninguno de los algoritmos se contradice entre sí. Después de probar con todos los algoritmos implementados y promediar, logramos un puntaje RAND de 0.95.

El sistema CAMLPAD utiliza una variedad de tipos de datos, como BRO (DNS / CONN), SNORT, YAF y Meraki, sino que también utiliza una medida combinada basada en la votación democrática que incorpora todos los diferentes tipos de datos para dar un mecanismo de detección de anomalías más holístico. Además de los datos, el uso de cuatro algoritmos diferentes, específicamente Isolation Forest, Puntaje de valor atípico basado en histograma, Factor de valor atípico local basado en clúster y detección de valor atípico basado en ángulo, da como resultado un sistema integrado que aprovecha las fortalezas de cada uno de los aprendizajes automáticos.

Y aunque el sistema CAMLPAD ha demostrado ser preciso y útil debido a la puntuación RAND de 0.95, un 95 por ciento similares entre sí, existen problemas específicos con el uso a gran escala de sistemas de detección de intrusos.

CAMLPAD aborda varios tipos de datos con los algoritmos más eficientes para proporcionar una canalización optimizada en tiempo real para la detección autónoma de anomalías.

En el futuro utilizaran aprendizaje automático con modelos supervisados para garantizar que todos los puntos de datos estén representados. En general, CAMLPAD logró una puntuación de rand ajustada del 95 por ciento, pero con el uso de varios modelos ML y haciendo que CAMLPAD sea más eficiente.

Planean ejecutar directamente su secuencia de comandos en el esquema de Metadata Encoding and Transmission Standard (METS) para alcanzar la máxima eficiencia y resultados oportunos. El esquema METS ayuda a codificar metadatos descriptivos y estructurales para objetos en una biblioteca digital. Además de ejecutar los modelos de aprendizaje automático en un servidor METS, planeamos limpiar los datos y hacer que cada campo sea coherente. Dado que los datos, transmitidos por Elasticsearch, son diferentes para cada grupo (según el día), el preprocesamiento de los datos para hacer que todos los puntos de datos sean similares ayudará a aumentar la eficiencia.

3.4.5 Effective and efficient network anomaly detection system using machine learning algorithm

Este estudio ha sido realizado por Mukrimah Nawir, Amiza Amir, Naimah Yaakob, Ong Bi Lynn Embedded, Cluster de Investigación de Redes e Informática Avanzada (ENAC), Escuela de Ingeniería Informática y de las Comunicaciones (SCCE), Universiti Malaysia Perlis (UniMAP), Malasia [14].

En este estudio utilizan el Machine Learning supervisado para la detección de anomalías de red en su sistema y que minimiza el coste de la comunicación y el ancho de banda de la red. Esto lo consiguen utilizando el conjunto de datos UNSW-NB15 para comparar su rendimiento en términos de su precisión (efectivo) y procesamiento tiempo (eficiente) y que un clasificador construya un modelo.

Con UNSW-NB15 tienen los datos que monitorizan del sistema, clasificando los datos ya sean normales o anómalos. Para descubrir patrones complejos anómalos conocidos o desconocidos de varios ataques maliciosos en el protocolo de red hay que elegir l algoritmo adecuado.

En este estudio realizan dos experimentos:

1.- El mejor algoritmo ML (algoritmo centralizado) para la detección de anomalías en el conjunto de datos UNSW-NB15

2.- Algoritmo distribuido para el sistema de detección de anomalías de red

<u>Primer experimento</u>

En el primer experimento para la detección de anomalías en el conjunto de datos UNSW-NB15 junto con la herramienta WEKA versión 3.8 utilizan cinco algoritmos de clasificación: Naïve Bayes (NB), Estimador de dependencia promedio (AODE), Red de función de base radial (RBFN), Perceptrón multicapa (MLP) y árboles J48.

También hay cuatro etapas involucradas: preparación del conjunto de datos, capacitación, validación y pruebas. Primero cargan el conjunto de datos de red (UNSW-NB15) que necesita clasificar las instancias de datos. Una vez que el conjunto de datos está listo, proceden a la segunda etapa (capacitación). Según los algoritmos que se utilizaron en este experimento, se configuran los parámetros de los algoritmos utilizados por defecto. Tercero, realizan validación cruzada diez vece, donde ven una alta precisión. Por último, la etapa de prueba mediante la recopilación de las medidas de rendimiento, la precisión y tiempo necesario.

Segundo experimento

Con el segundo experimento el algoritmo distribuido lo diseñan como un algoritmo centralizado para la detección de anomalías. El código se escribió con el lenguaje JAVA usando eclipse.

Realizan la ampliación del primero experimento. Realizan el diseño de un algoritmo AODE distribuido solo para superar el problema de la centralización. El experimento, una vez cargado los datos y la ruta de clase del conjunto de datos etiquetado de red. Para el algoritmo distribuido, la toma de decisiones seleccionando aleatoriamente el nodo disponible en el sistema de red para agregar el resultado recolectado para medir el resultado de la predicción.

Con el ML supervisado tienen en cuenta las características importantes del etiquetado en los conjuntos de datos. Y comentan que el mejor algoritmo de aprendizaje automático para el conjunto de datos de red es AODE, con una precisión comparable de 97.26% y un tiempo de aproximadamente 7 segundos. Además, el algoritmo distribuido resuelve el problema de la centralización con la precisión y el tiempo de procesamiento aún en comparación con un algoritmo centralizado, a pesar de que se necesita una pequeña.

3.4.6 Patente de “System and method for automated machine-learning, zero-day malware detection”

La Patente creada consta de realizaciones que proporcionan sistemas y métodos mejorados para aprendizaje automático en detección de malware de día cero. Diferentes ventajas se consiguen mediante métodos para la

detección mejorada de malware de día cero donde reciben un conjunto de archivos de entrenamiento que se sabe que son maligno o benigno. A continuación, se explicarán alguno de esos métodos.

Método 1

Divide el conjunto de archivos de entrenamiento en una pluralidad de categorías y entrena clasificadores específicos de categoría que distinguen entre archivos malignos y benignos en una categoría de archivos.
El entrenamiento puede incluir:

- Seleccionar una de la pluralidad de categorías de archivos de entrenamiento
- Identificar las características presentes en los archivos de entrenamiento en la categoría seleccionada de archivos de entrenamiento
- Evaluar las características identificadas para determinar las características identificadas más efectivas para distinguir entre archivos malignos y benignos
- Construir un clasificador específico de categoría basado en las características evaluadas

Método 2

Analiza un archivo de entrenamiento del conjunto de archivos de entrenamiento para determinar las características del archivo de entrenamiento.

Etiqueta las características determinadas del archivo de entrenamiento con etiquetas meta-características calificadas (QMF), repite el análisis y el etiquetado de los archivos de entrenamiento restantes en el conjunto de archivos de entrenamiento.

Por último, construye un modelo que identifica las características indicativas de un archivo maligno que utiliza las características etiquetadas con QMF, en el que el modelo puede usarse para detectar archivos malignos.

El etiquetado incluye extraer una de las características determinadas del archivo de entrenamiento, identificar una ubicación de la característica extraída en el archivo de entrenamiento, determinar una etiqueta QMF apropiada de la característica extraída en función de la ubicación identificada, aplicar la etiqueta QMF determinada a la característica extraída y repetir la extracción, identificación, determinación y aplicación de las características determinadas restantes del archivo de capacitación.

Método 3

Analiza el conjunto de archivos de entrenamiento para determinar las características de archivos de entrenamiento.

Después recibe una descripción del conjunto de características que incluye una etiqueta semántica para cada clase de atributo presente en los archivos de entrenamiento y un conjunto de atributos correspondientes que componen la clase de atributo.

Genera una pluralidad de vectores de características (FV) específicos de clase de atributo para los archivos de entrenamiento usando las características determinadas y la descripción del conjunto de características, en donde los FV son vectores de características presentes en archivos malignos de la clase de atributo. Concatena la pluralidad de FV específicos de clase de atributo en un vector de características extendido (EFV) para los archivos de entrenamiento.

Por último, genera un clasificador de archivo de destino basado en el EFV usando una pluralidad de algoritmos de clasificador.

BluVector ha desarrollado un método de detección utilizando su motor patentado de aprendizaje automático supervisado para detectar malware basado en archivos en milisegundos en la red. Incluso si el malware nunca se ha visto antes [16].

4.Conclusiones

El ML cada vez está más utilizado en empresas y en las organizaciones gubernamentales para protegerse de posibles ataques y amenazas que puedan tener en el entorno de la seguridad. Las empresas tendrán que empezar por entender los principios básicos de esta tecnología, para poder usarla a su favor y mejorar la productividad de todas las operaciones de su negocio.

Hay que tener en cuenta que el ML es una herramienta que ayuda a proteger de ataques y amenazas, pero que de la misma manera se extiende ese poder a las personas que pueden utilizarlo para el uso contrario. Siempre habrá personas que quieran encontrar debilidades en los sistemas o algoritmos de ML para su propio beneficio y así poder llegar a evitar los mecanismos de seguridad implementados.

Hay mucho interés general para desarrollar sistemas que puedan ayudar en la problemática en la seguridad con la ayuda ML. En el ML se procesan grandes cantidades de datos permitiendo predecir incidentes críticos. A partir de los patrones de tráfico, las conexiones, la actividad del usuario y aspectos de la red se pueden aprender y tomar decisiones con el ML. Las principales aplicaciones que se utilizan en ML en el campo de la seguridad informática son la detección de malware, violaciones en políticas, amenazas internas o la detección de intrusos.

Los atacantes crecen y desarrollan técnicas para evitar ser detectados y llevar a cabo ataques informáticos con un fin. Para evitar algunos de estos ataques se utiliza la detección de intrusos.

Podemos decir que no hay un algoritmo específico para el ML sino que son los datos los que definirán que la aplicación de un algoritmo será aplicado a la ciberseguridad. Por ejemplo, de la misma manera que Google Home necesita textos para entrenar al modelo, en el caso de la ciberseguridad, los datos pueden provenir de ficheros infectados, ficheros de log, captura de paquetes y demás.

También mencionar, que, del conjunto de datos, las características que se puedan obtener van a ser de los factores más importantes a la hora de entrenar un modelo. La calidad y selección de estos datos hará que un modelo funcione con la mayor precisión, por lo que es fundamental la correcta colección de datos. Una vez se obtienen los datos se requiere de convertir las características a un formato donde pueda aplicarse el ML.

Es necesario ser específico para obtener conjuntos de datos. En la mayoría de las ocasiones, la obtención de estos datos es una tarea difícil, ya que estos datos están sujetos a confidencialidad o protegidos por la protección de datos.

Comentado [EGG1]:

Tal y como se ha visto en los estudios se han realizado detección de anomalías según el comportamiento del usuario, donde uno de los desafíos a los que se pueden enfrentar es que un usuario tenga conocimiento de las defensas del sistema. Es difícil crear este tipo de sistemas dado a que son necesarios un conjunto de datos para entrenar los modelos y este tipo de datos suele ser información sensible. Por este motivo, suelen ser empresas especializadas que puedan recopilar información des de dentro de la red y esta misma empresa desarrolle el algoritmo según las necesidades, este podría ser el ejemplo de la empresa comentada BluVector.

Otro de los puntos a comentar es que se requiere de expertos para poder analizar los ataques, problemas de seguridad que puedan presentarse. Es necesario analizar los datos por un tiempo largo y poder extraer las características que puedan aportar el mejor conjunto de datos y en consecuencia los mejores datos para entrenar el modelo y obtener resultados con predicciones con más exactitud.

Construir el modelo puede ser una labor complicada por lo que se necesitan herramientas externas para gestionar la gran cantidad de datos. Al tener este gran volumen de datos es necesaria la utilización de Big Data o Hadoop. Esto hace que sean necesarias máquinas de alta gama dedicados, gpus, i en general maquinaria más potente para poder procesar los datos.

Entre los algoritmos más comunes utilizados se encuentra el Naïve Bayes, algoritmo que no destaca por su exactitud en los resultados sino por su velocidad y la memoria que necesita. Este algoritmo también permite realizar estudios con un PC sin muchos requerimientos si se compara con otros algoritmos. Por este motivo, siempre y cuando no sea un conjunto de datos de dimensiones muy grandes, se podrá utilizar dicho algoritmo para realizar experimentos y prueba como puede ser con el software WEKA. Por otro lado, es un algoritmo que aun no siendo el algoritmo con mayor exactitud suele ser suficiente para la mayoría de las aplicaciones. En el caso práctico aplicado a la seguridad podemos ver que su uso es extendido y utilizado para la detección de anomalías. Es un algoritmo que puede resultar muy útil cuando hay variables independientes y se pueden comparar probabilidades. Por ejemplo, calcular las probabilidades de los diferentes ataques que podemos recibir.

Con el algoritmo de Isolation forest se pueden aislar las observaciones de anomalías y calcular un puntaje de anomalía. Este último calculado como el número de condiciones requeridas para separar una observación dada. Por otro lado, aislar observaciones normales requiere más condiciones. Este algoritmo se puede encontrar en diversos estudios para reducir el fraude. Este método detecta anomalías basadas únicamente en el concepto de aislamiento sin emplear ninguna medida de distancia o densidad, fundamentalmente diferente de todos los métodos existentes.

Por otro lado, en los ataques DDoS podemos ver todo tipo de algoritmos utilizados sin tener una clara tendencia. Entre estos algoritmos se encuentra el árbol de decisión e Isolation Forest entre otros. Por lo que se puede definir que no hay el algoritmo en concreto para un ataque en concreto, sino que se trata

de realizar una buena recopilación de datos, seleccionar las mejores características y luego entrenar el modelo con un algoritmo e ir adaptándolo y/o personalizando según los requerimientos que se necesiten.

Hay que ser conscientes de que tiene que llevarse a cabo una divulgación ética y responsable del uso del ML. La controversia va más allá de si las máquinas pueden llegar a pensar como los humanos. Si realmente fuera así, tendría profundas implicaciones sociales y causaría cambios irrevocables en los cimientos de nuestra sociedad. Para procesar toda esa información algunos plantean que, en el futuro, la computación cuántica podrá ayudar donde la computación clásica no pueda llegar.

5. Glosario

Data Science Ciencia basada en el estudio de los datos, siendo una de las partes más importantes la parte del Data mining.

ML (Machine Learning) El aprendizaje automático es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan.

IA (Inteligencia Artificial) Inteligencia llevada a cabo por máquinas. En ciencias de la computación, una máquina «inteligente» ideal es un agente flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea

DL (Deep learning) es un conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial.

Entropía En el ámbito de la teoría de la información la entropía, también llamada entropía de la información y entropía de Shannon (en honor a Claude E. Shannon), mide la incertidumbre de una fuente de información.

Clustering es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud

DDoS es el llamado ataque de denegación de servicio distribuido, también llamado DDoS (por sus siglas en inglés, Distributed Denial of Service) el cual se lleva a cabo generando un gran flujo de información desde varios puntos de conexión hacia un mismo punto de destino.

Smurf es un ataque de denegación de servicio que utiliza mensajes de ping al broadcast con spoofing para inundar (flood) un objetivo (sistema atacado).

Telemetría Sistema de medición de magnitudes físicas que permite transmitir los datos obtenidos a un observador lejano.

Spam El spam puede definirse como mensajes no deseados que se envían principalmente por vía electrónica.

Phishing conocido como suplantación de identidad, es un término informático que denomina un modelo de abuso informático y que se comete mediante el uso de un tipo de ingeniería social, caracterizado por intentar adquirir información confidencial de forma fraudulenta

PE Portable Executable (PE) es un formato de archivo para archivos ejecutables.

DLL (Dynamic-link library) Una biblioteca de enlace dinámico es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

PCAP El pcap es una interfaz de una aplicación de programación para captura de paquetes.

IPFIX (Internet Protocol Flow Information Export) es un protocolo IETF y define cómo se formatea y transfiere la información del flujo IP de un exportador a un recolector.

DNS (Domain Name System) Su función es "traducir" nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red

SIP (Session Initiation Protocol) Protocolo de inicio de sesión es un protocolo desarrollado por el grupo de trabajo MMUSIC (Multiparty Multimedia Session Control) del IETF con la intención de ser el estándar para la iniciación, modificación y finalización de sesiones interactivas de usuario donde intervienen elementos multimedia como el video, voz, mensajería instantánea, juegos en línea y realidad virtual.

HTTP (Hypertext Transfer Protocol) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

AWS (Amazon Web Services) es una colección de servicios de computación en la nube pública ofrecidos por Amazon.com

UNIX es un sistema operativo portable, multitarea y multiusuario; desarrollado en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Dennis Ritchie, Ken Thompson y Douglas McIlroy.

Shell En informática, el shell o intérprete de órdenes o intérprete de comandos es el programa informático que provee una interfaz de usuario para acceder a los servicios del sistema operativo.
Token es el proceso de sustitución de un elemento de datos sensible por un equivalente no sensible, denominado token, que no tiene un significado o valor extrínseco o explotable.

Dataset Un conjunto de datos es una colección de datos habitualmente tabulada.

WEKA (Waikato Environment for Knowledge Analysis) es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato.

HADOOP es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre.1 Permite a las aplicaciones trabajar con miles de nodos y petabytes de datos

BIG DATA es un término que hace referencia a conjuntos de datos tan grandes y complejos como para que hagan falta aplicaciones informáticas no tradicionales de procesamiento de datos para tratarlos adecuadamente.

6. Bibliografía

[1] https://www.darktrace.com/es/ Noviembre-Diciembre 2019

[2] “AI and Machine Learning in Cyber Security” Raffael Marty . Enero de 2018 [https://towardsdatascience.com/ai-and-machine-learning-in-cyber-security-d6fbee480af0]

[3] “System predicts 85 percent of cyber-attacks using input from human experts” Adam Conner-Simons, Abril 2016 [http://news.mit.edu/2016/ai-system-predicts-85-percent-cyber-attacks-using-input-human-experts-0418]

[4] “Using machine learning to hunt down cybercriminals” Adam Conner-Simons, Octubre de 2019 [http://news.mit.edu/2019/using-machine-learning-hunt-down-cybercriminals-1009]

[5] “Avanzan las tareas de ciberseguridad para la cumbre del G20” Noviembre del 2018 [https://www.ambito.com/politica/avanzan-las-tareas-ciberseguridad-la-cumbre-del-g20-n4038270]

[6] “Anomaly Detection” Autores: j-martens, garyericson, ktoliver , PeterCLu, jeannt, v-thepet, v-kents, sdgilley, MijeongJeon . Mayo del 2019 [https://docs.microsoft.com/es-es/azure/machine-learning/studio-module-reference/anomaly-detection?WT.mc_id=mlrcg-acomblog-martydon]

[7] “Spam does not bring us joy—ridding Gmail of 100 million more spam messages with TensorFlow” Neil Kumaran, Febrero de 2019 [https://cloud.google.com/blog/products/g-suite/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow]

[8] “Machine Learning for Cyber Security – Static Detection of Malicious PE Files” Yasmin Bokobza, Yosef Arbiv. Enero del 2019 [https://www.cyberbit.com/blog/endpoint-security/machine-learning-for-cyber-security-static-detection/]

[9] Artículo: “An application of Machine Learning to anomaly Detection” Terran Lane and Carla E. Brodley en la escuela Electrical and Computer Engineering
[10] Artículo: “Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM Project” Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, y Philip K

[11] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html Universidad de California, Irvine, Edureka

[12] Artículo: “DEFENSE MECHANISM FOR DDoS ATTACK THROUGH MACHINE LEARNING, realizado por Sujay Apale, Rupesh Kamble, Manoj Ghodekar, Hitesh Nemade, Rina Waghmode del departamento de ingeniería de computación de AISSMS COE, Pune, India.

[13] Artículo: “CAMLPAD: Cybersecurity Autonomous Machine Learning Platform for Anomaly Detection” realizado por Ayush Hariharan; del departamento de ciencias de computación de la Academia de Ciencias del condado de Loudon, EEUU; Ankit Gupta y Trisha Pal; del departamento de ciencias de la computación TJHSST de Alexandria, de EEUU.

[14] Artículo: “Effective and efficient network anomaly detection system using machine learning algorithm” realizado por Mukrimah Nawir, Amiza Amir, Naimah Yaakob, Ong Bi Lynn Embedded, Cluster de Investigación de Redes e Informática Avanzada (ENAC), Escuela de Ingeniería Informática y de las Comunicaciones (SCCE), Universiti Malaysia Perlis (UniMAP), Malasia

[15] Patente US9665713B2. “System and method for automated machine-learning, zero-day malware detection” Inventores: Bhargav R. AVASARALABrock D. BOSEJohn C. DayDonald Steiner

[16] ”Trained to find the bad” https://www.bluvector.io/technology/machine-learning-engine/

Otros recursos

- Canal de Youtube: “Edureka!”
- Canal de Youtube: “Machine Learning for Cyber Security”
- Canal de Youtube: “Ligdi Gonzalez”
- Libro: “Data Mining and Machine Learning in Cibersecurity” Autores: Sumeet Dua y Xian Du . CRC Press Taylor & Francis Group AN AUERBACH BOOK.