Proyecto 1: Detección de Ataques

Luis Urbina[1], Isabel Ortiz[2]

Universidad Del Valle de Guatemala
CC3094 – Security Data Science
Sección 10
[1] *18473* [2] *18176*

Febrero 25, 2022

The objective of this project is to correctly classify four different types of network traffic (benign traffic, DdoS (RU-Dead-Yet) attack, DdoS (Slowloris), and SYN Scan attack) using two Machine Learning classification algorithms, these being the Random Forest and Naive Bayes algorithm. For this project, a reduced version () of the SIMARGL (Secure Intelligent Methods for Advanced Recognition of Malware and Stegomalware) project dataset was used to train and test the algorithms for performance purposes.
Keywords: network traffic, Random Forest, Naive Bayes, attacks, malware, stegomalware

1 Introducción

En la actualidad, la comunicación a través de internet es más frecuente que nunca, cada día se intercambian cientos de miles de Gigabytes de datos entre diferentes dispositivos. Debido a la cantidad masiva de información que se encuentra disponible, hay personas con fines maliciosos que han desarrollado programas para interceptar, robar y utilizar información privilegiada para sus propios fines.

Entre estos ataques se encuentran: (1) DdoS (RU-Dead-Yet) attack, que abre solicitudes HTTP concurrentes al servidor HTTP y retraza el envío del cuerpo de la solicitud POST al punto de que los recursos del servidor se saturan (Najafabadi, Khoshgoftaar, Napolitano y Wheelus, 2016). (2) Ddos Slowloris attack, que ocasiona que una máquina elimine el servidor web de otra máquina al mantener abiertas múltiples conexiones con el servidor web de destino el mayor tiempo posible al enviar una solicitud parcial en cada conexión (Tanishka, Subbaiah, Goyal, Sakxena, Y Mishra, 2018). (3) SYN Scan attack, en el el atacante configura una conexión de TCP/IP con un servidor en todos los puertos posibles mediante el envío de un paquete SYN para iniciar un protocolo de enlace de tres vías, a cada puerto del servidor (Balram y Wiscy, 2008).

En este trabajo se presentan dos modelos de Machine Learning para la clasificación de datos, Random Forest y Naive Bayes, que serán entrenados con una versión reducida del dataset provisto por el proyecto SIMARGL (Métodos inteligentes seguros para el reconocimiento avanzado de malware y estegomalware, por sus siglas en inglés). Posteriormente, se describe el proceso al que los datos fueron sometidos previo al entrenamiento del modelo, luego se incluyen los resultados obtenidos para cada uno de los modelos en conjunto con las métricas de precisión, recall, y exactitud; Así como las curvas ROC respectivas. Finalmente, se presentan las conclusiones basadas en los resultados obtenidos.

2 Marco Teórico

Para comprender completamente la metodología, los resultados y las conclusiones de este proyecto, es necesario tener conocimiento sobre algunos conceptos, términos, y modelos relevantes. Estos se presentan a continuación:

2.1 Oversampling

El problema de clasificación desequilibrada es un problema que ocurre cuando hay un sesgo severo en la distribución de clase en los datos de entrenamiento de un modelo, esto sucede cuando hay muy pocos datos de una clase en el dataset en comparación con otras clases causando que el algoritmo de Machine Learning ignore a la clase minoritaria causando un resultado erróneo en la clasificación de los datos.

El oversampling consiste en introducir un sesgo forzado para seleccionar más muestras de una clase (la minoritaria) que de otra (la mayoritaria) para compensar por el desbalance en la data (Kaur y Gosain, 2018).

En el dataset reducido que se utilizó para el desarrollo de este proyecto se encontró un desbalance entre los distintos típicos de tráfico. Por una parte, el tráfico de tipo "no benigno" era el que presentaba el mayor número de datos, mientras que el tráfico de tipo "Ddos slowloris" era el que presentaba el menor número de datos respecto a las otras dos clases. Por lo que se utilizó un algoritmo de oversampling aleatorio para balancear los datos de cada clase.

2.2 Random Forest

Un random forest, es un conjunto de árboles de decisión combinados con bagging (también conocido como bootstrap aggregation), esto permite que distintos árboles vean distintas proporciones de los datos, a pesar de que ningún arbol ve todos los datos de entrenamiento. Así se garantiza que un árbol se entrene con distintas muestras de datos para un mismo problema, al combinar los resultados de cada árbol se compensan los errores de unos con otros y se obtiene una predicción que generaliza mejor (Breiman, 2001).

Este algoritmo es mayormente utilizado para problemas de regresión y de clasificación sobre un dataset de gran tamaño donde la interpretabilidad no es una preocupación, por lo que se consideró que este sería un algoritmo ideal para alcanzar el objetivo de este proyecto.

2.3 Naive Bayes

Naive Bayes es un grupo de algoritmos de clasificación de aprendizaje automático supervisado basados en el teorema de Bayes. Esta familia de algoritmos es útil cuando la dimensionalidad de las entradas es alta y cuando se puede asumir que hay una independencia fuerte entre las características del dataset. La ventaja de los este algoritmo sobre otros es que los algoritmos Naive Bayes son entrenados de manera eficiente, por lo que necesitan una cantidad relativamente pequeña de datos para estimar los parámetros necesarios para la clasificación, por lo que se consideró que este también sería un algoritmo ideal para el objetivo planteado (Rish, 2001).

3 Metodología

El principal objetivo al explorar el impacto de utilizar tecnicas de oversampling es con caracter previo al proceso de clasificacion de datasets respectivos que se encuentran desbalanceados. Estos algoritmos permiten acondicionar el dataset finalmente clasificado, generando muestras adicionales cuando sea necesario para mejorar el proceso de clasificacion de las clases minoritarias sin perjudicar a las clases mayoritarias del dataset, lo cual resulta de gran interes en ciertas aplicaciones.

3.1 Reducción del dataset

La Reducción del dataset es muy importante puesto que, se hace una serie de pasos para poder tener una muestra significativa de datos y asi hacer todos los pasos correspondientes a lo que se necesita. Hay tres métodos para realizar reducción de dimensionalidad:

- Selección de características, es decir, escoger un subconjunto de las características originales que, según cierto criterio, representen bien el conjunto de datos.
- Derivación de características, método consistente en la creación de nuevas características a partir de las originales. Lógicamente, ésto supone una reducción de dimensionalidad solo si creamos características que combinen dos o más de las originales y podemos sustituir las originales por las nuevas sin que se produzca una pérdida de información excesiva.
- Agrupación de muestras, o aplicación de algoritmos de clustering para la identificación de clusters de muestras que sean identificables usando un menor número de características.

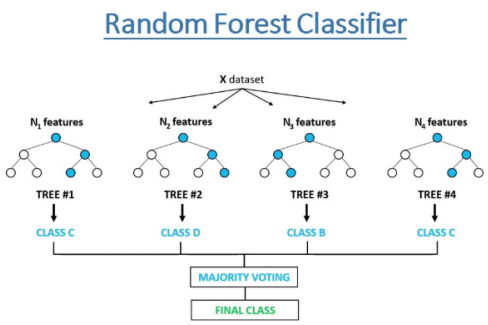
3.2 Preprocesamiento

EL preprocesamiento es una técnica que transforma los datos sin procesar en un formato comprensible.

Los datos del mundo real (datos sin procesar) siempre están incompletos y esos datos no se pueden enviar a través de modelos, ya que causarían ciertos errores. El dataset proporcionado, tenia data sin procesar es por ello que se hizo la seleccion adecuada de caracteristicas para poder trabajar con los datos deseados.

En el caso de este proyecto, se contaba con una serie de datos bastante variada, es por ello que el preprocesamiento se encargo de seleccionar, manejar y preparar los datos para posteriormente ser analizados en su totalidad. Asi con una muestra significativa, pudo realizarse la cantidad de pruebas pertinente para tener una respuesta clara y concreta.

3.3 Random Forest



Tal y como se pude observar, este modelo sirver para entrenar el algoritmo con las características procesadas seleccionadas de nuestro conjunto de datos, realizaremos predicciones y luego calcularemos la precisión del modelo.

3.4 Naive Bayes

El clasificador Naive Bayes asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica. Este modelo considera que cada una de estas características contribuye de manera independiente a la probabilidad de que esta fruta sea una manzana, independientemente de la presencia o ausencia de las otras características.

4 Resultados

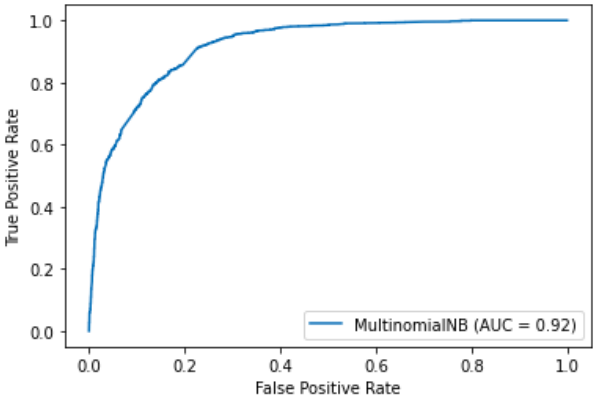
4.1 Random Forest

```
0.73764021603656
Matriz de confusion [[2064  676]
 [ 587 1487]]
              precision    recall  f1-score   support

           0       0.78      0.75      0.77      2740
           1       0.69      0.72      0.70      2074

    accuracy                           0.74      4814
   macro avg       0.73      0.74      0.73      4814
weighted avg       0.74      0.74      0.74      4814
```

En este caso, la precision dada con este modelo de 0.78 de porcentaje. Por lo cual, se hace la semejanza que pudo entrenarse mejor el dataset utilizado y poder tener un precision mas acertada.

4.2 Naive Bayes

Para este modelo, puede verse que tiene un 0.92 porcentaje de aprobacion lo que indica que este modelo es el adecuado para realizar el analisis pertinente.



5 Conclusiones

- Conclusión 1: Los resultados obtenidos evidencian las bondades del proceso de balanceo datos introducido por algoritmos de oversampling en situaciones en las que existe una gran cantidad de ruido en las muestras de los datos.
- Conclusión 2: El Modelo Bayes Naive se puede aplicar a diferentes tipos de clasificación y predecir la probabilidad de que un nuevo dato pertenezca a una de estas clases. Siempre habrá la posibilidad de clasificación errónea para cualquier tipo de modelo de aprendizaje automático, pero con seguridad mediante la aplicación de este modelo, el número de trabajos manuales se reduce drásticamente, así como el nivel de confianza de los aumentos de clasificación. En el caso de este proyecto, fue el que mas accuracy tuvo.

6 Referencias

Najafabadi, M., Khoshgoftaar, T., Napolitano, A., y Wheelus, C. (2016). "*RUDY Attack: Detection at the Network Level and Its Important Features*". Actas de la Vigésima Novena Conferencia Internacional de la Sociedad de Investigación de Inteligencia Artificial de Florida.

Tanishka, S., Subbaiah, D., Goyal, A., Sakxena, A., y Mishra, A. (2018). *Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools*. IEEE. Bangalore, India

Balram, S. y Wiscy, M. (2008). *Detection of TCP SYN Scanning Using Packet Counts and Neural Network*. IEEE. Bali, Indonesia

Kaur, P. y Gosain, A. (2018). *Comparing the Behavior of Oversampling and Undersampling Approach of Class Imbalance Learning by Combining Class Imbalance Problem with Noise*. 10.1007/978-981-10-6602-3_3.