Proyecto 1: SIMARGL. Detección de malware y stego-malware en la red.

Marchena G., Sergio A.

Febrero 2022

Universidad del Valle de Guatemala, mar16387@uvg.edu.gt

Abstract

The present work consisted of analyzing the [1] [1] SIMARGL Project database and implementing machine learning models in order to predict and classify the type of intrusions (malware and stego-malware) within a network. For this, the first thing done ws an exploratory analysis. Then a preprocessing process was carried out for the encoding of the categorical variables, then the most important characteristics were selected based on the existing correlation (Principal Component Analysis), the database was divided into training, validation and test sets. Finally, two models were implemented: Decision Tree Classifier and K Nearest Neighbor Classifier. With said models, predictions of more than 85% accuracy were obtained.

I. Introducción

Este trabajo de investigación tiene como objetivo principal la elaboración de dos modelos de machine learning sobre una base de datos de más de doce millones de registros relacionados con intrusiones de malware y stego-malware en una red.

Sobre los datos se hizo un proceso de limpieza, análisis y preprocesamiento (correlación entre variables, cambio de escalas, codificación de variables, etc) de datos para poder dejar la base de datos lista para poder ser alimentada en un modelo de Árboles de Decisión y en un modelo de KNN. Se separaron los datos en un set de entrenamiento (55%) , uno de validación (30%) y otro de pruebas (15%). Dichos modelos tuvieron resultados muy buenos (97% y 98% de accuracy), y es posible argumentar que se haya dado un caso de overfitting, donde los modelos se apegan exactamente a los datos y predicen con mucha eficacia.

Después de haber probado los modelos se hizo un análisis de Receiver Operating Characteristic (ROC) para cada modelo. y se obtuvieron los resultados de 0.66 y 0.67, lo cual nos indica que la calidad de la clasificación de los modelos no es muy buena.

Al final, se obtuvieron dos modelos suficientemente buenos y aceptables para la clasificación de intrusiones de malware en una red.

[1] https://simargl.eu/download/SIMARGL-general_presentation.pdf

II. Marco Teórico

A. Contexto y Dataset

El proyecto SIMARGL (Secure Intelligent Methods for Advanced RecoGnition of malware and stego-malware) – es auspiciado por un consorcio de 14 miembros, de 7 países Europeos. Los objetivos estratégicos consisten en proveer métodos efectivos para contrarrestar ataques de cibercrimen, y proponer, implementar y validar métodos innovativos de machine learning (ML) y Deep Learning (DL) para la detección de malware.

Fig. 1. Logo de SIMARGL. Extraido de: https://simargl.eu

De este proyecto surge la creación de un dataset moderno en base a la captura de ataques en una red diseñada para dicho propósito, ya que anteriores datasets utilizados para la detección de intrusos, como KDD-99 registran atributos que ya no se usan en la actualidad, o tienen poca calidad en los datos, de modo que su usabilidad para la implementación de modelos es bajo.

El dataset tiene las siguientes características:

- 6,570,058 observaciones de tráfico puro (Normal Flow).
- 5, 637, 815 observaciones consideradas anomalías
  - 2,496,814 consideradas como un ataque 'SYN' Scan.
  - 2, 276, 947 consideradas como un ataque DdoS (R-U-Dead-Yet o 'RDUY').
  - 864, 054 consideradas como un ataque DdoS (Slowloris).

B. Modelos de Machine Learning

El objetivo de este proyecto es usar modelos distintos de machine learning (ML) para poder predecir las intrusiones de

malware y stego-malware en una red. Se trabajó con dos modelos de aprendizaje supervisado de clasificación.

Machine learning es una técnica muy utilizada en la actualidad cómo un método para encontrar patrones. Se podría definir cómo [2]"una disciplina informática con el diseño de algoritmos que permiten a las computadoras desarrollar comportamientos basados en datos empíricos. Estos algoritmos se pueden organizar en la siguiente jerarquía: aprendizaje supervisado, no supervisado y semi-supervisado. Por lo tanto, Machine Learning es más que nada un sub-campo multidisciplinario que se ocupa del descubrimiento de patrones en grandes conjuntos de datos que involucran métodos de inteligencia artificial, aprendizaje automático, estadísticas y sistemas de bases de datos.[2]"

El aprendizaje supervisado es la actividad de aprender una función que asigna una entrada a una salida en función muestras de entradas y salidas. Infiere una función a partir de los datos de entrenamiento y se comparan con datos de prueba. En otras palabras, los algoritmos de aprendizaje supervisado se refieren al tipo de algoritmos donde la variable objetivo o la variable a predecir en un conjunto de datos es conocida.

Los modelos utilizados para este trabajo fueron modelos de clasificación. La clasificación es una técnica en la que categorizamos los datos en un número determinado de clases. El objetivo principal de un algoritmo de clasificación es identificar la categoría/clase a la que pertenecera un nuevo dato.

Para este trabajo se trabajó con un modelo de Árboles de Decisión y un modelo de KNN (K Nearest Neighbors).

El modelo de árboles de decisión produce una secuencia de reglas que se pueden usar para clasificar los datos.

Ventajas: es fácil de entender y visualizar, requiere poca preparación de datos y puede manejar datos numéricos y categóricos.

Desventajas: el árbol de decisión puede crear árboles complejos que no se generalizan bien, y los árboles de decisión pueden ser inestables porque las pequeñas variaciones en los datos pueden generar un árbol completamente diferente.

El modelo de KNN (vecinos K más cercanos) es un tipo de aprendizaje perezoso ya que no intenta construir un modelo interno general, sino que simplemente almacena instancias de los datos de entrenamiento. La clasificación se calcula a partir de un voto de mayoría de los vecinos 'k' más cercanos de cada punto.

Ventajas: este algoritmo es simple de implementar, resistente a datos de entrenamiento ruidosos y efectivo si los datos de entrenamiento son grandes.

Desventajas: Necesita determinar el valor de K y el costo de cálculo es alto ya que necesita calcular la distancia de cada instancia a todas las muestras de entrenamiento.

III. Metodología

Este proyecto de investigación comprende 6 fases distintas de trabajo. Dichas fases tienen un objetivo claro y conciso que permite llegar a conclusiones mejor sustentadas. Las 6 fases son:

A. Análisis exploratorio.
B. Pre-procesamiento.
C. Selección de características.
D. Separación de los datos.
E. Implementación de modelos.
F. Análisis de métricas obtenidas.

A. Análisis Exploratorio

En esta primera fase se cargan los datasets a trabajar y se empieza a explorar las variables y características. Esta etapa es considerada un resumen de los datos y de las variables principales. Se hizo un análisis de las variables y se encontró qué en el set de datos hay 12,207,873 observaciones y 50 columnas o variables. De estas 50 variables, 7 tienen estructura de texto o mezcla de texto y números.

B. Pre-procesamiento.

Para esta segunda etapa, después de haber explorado los datos se puede notar que hay que hacerle algunos cambios o modificaciones al set de datos El objetivo de esta fase es preparar los datos para que estos sean introducidos a algún modelo. En esta fase se realiza la codificación de variables de texto a numéricas para que sean más fáciles de interpretar. Por ejemplo la variable objetivo 'LABEL': las observaciones con la clasificación 'Normal flow' fueron cambiadas al número 1, las observaciones con 'SYN Scan - aggressive' al 2, las de 'Denial of Service R-U-Dead-Yet' al 3 y las observaciones de 'Denial of Service Slowloris' al número 4.

C. Selección de características.

En esta fase se hizo un test de correlación multivariable de Pearson con todas las observaciones en el set de datos. El objetivo de esta fase es poder seleccionar las mejores variables (con más correlación con la variable objetivo 'LABEL') y de esta forma ayudar al modelo más adelante a tener mejores resultados.

Se obtuvo la siguiente información:

[2] Vázquez, A. M. (2018). Introducción a Machine Learning. https://unidad.gdl.cinvestav.mx/doc/investigacion/computacion/Introduccion-Machine-Learning.pdf

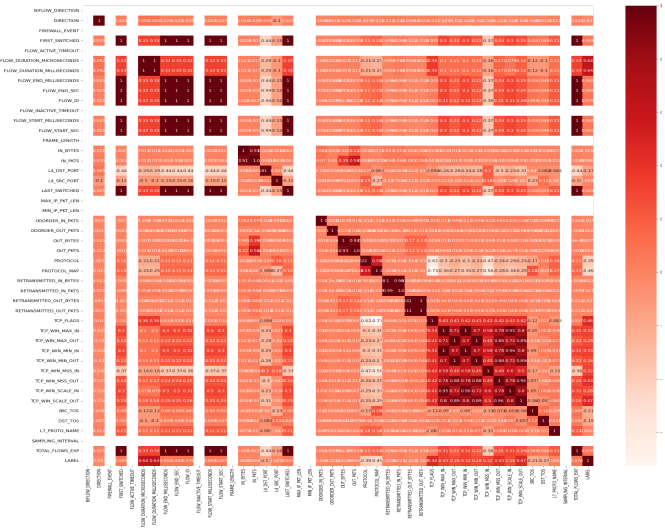

Fig. 2. Matriz de correlación en el set de datos.

D. Separación de los datos.

Esta fase es muy importante cuando se trabaja con modelos de machine learning. Luego de seleccionar las variables con más correlación, se hace un nuevo set de datos con esas variables. Aquí se separa el nuevo set de datos, en 3 diferentes set de datos. Cada uno tiene la misma cantidad de variables (características o columnas) pero difieren en cantidad de observaciones. Para el primer set de datos, set de entrenamiento, solamente tendrá el 55% del total del número de observaciones, para el segundo set de datos, set de validación, tendrá el 15% del total del número de observaciones y para el tercer set de datos, set de pruebas, solamente tendrá el 30% del total del número de observaciones, para un total de 100% de las observaciones.

E. Implementación de modelos.

En esta etapa del proyecto de detección de intrusiones malignas en la red, se implementan los dos modelos anteriormente mencionados, árboles de decisión y KNN.

Algo que hay que resaltar es que cada modelo se aplica de la siguiente manera: primero se entrena el modelo con el 55% de los datos de alta correlación. Luego, el modelo intenta predecir (clasificar) el un 15% de los datos de alta correlación (set de validación). Si los resultados obtenidos son aceptables (se miran las métricas) se procede a predecir (clasificar) el otro 30% de los datos de alta correlación (set de pruebas).

F. Análisis de métricas obtenidas.

Por último, ya teniendo las predicciones de cada modelo se tiene que observar varias métricas. Para comenzar se mira la exactitud del modelo. Esta medida significa qué tan bien predice el modelo en el sentido de que el valor real y la predicción es exactamente la misma.

Luego se calcula la matriz de confusión. Esta matriz evalúa la exactitud de las predicciones para cada caso. Nos dice qué exactamente esta prediciendo el modelo. Es una representación para ver los valores reales vs los valores predecidos por el modelo.

Después, se obtiene el reporte de clasificación, donde se muestran los valores de: 'precisión', 'recall' y 'f1-score' para cada clasificación de la variable objetivo.

Además, se calcula el valor de ROC. Esta medida ayuda a saber la calidad de la clasificación de los modelos.

IV. Resultados

Para el primer modelo, de árbol de decisión obtuvieron los siguientes resultados:

1. Para el set de datos de validación:

```
validation data:

accuracy: 91.73 %
------------------------------------------------------------
confusion matrix
 [[849305 109274   9414   8228]
 [  3437 367099     13      3]
 [  8395     23 326344   2867]
 [  5401      8   2784 120274]]
------------------------------------------------------------
              precision    recall  f1-score   support

  Normal Flow       0.98      0.87      0.92    976221
          SYN       0.77      0.99      0.87    370552
        RUDY        0.96      0.97      0.97    337629
   Slowloris        0.92      0.94      0.93    128467

    accuracy                           0.92   1812869
   macro avg        0.91      0.94      0.92   1812869
weighted avg        0.93      0.92      0.92   1812869

------------------------------------------------------------
```

Fig. 3. Exactitud, matriz de confusión y reporte por clases de Árbol de decisión con datos de validación (15%).

2. Para el set de prueba:

```
test data:
------------------------------------------------------------
accuracy: 91.75 %
------------------------------------------------------------
confusion matrix
 [[1723689  221609   19118   16659]
 [   7064  746158      28       5]
 [  16906      34  663895    5796]
 [  10975       6    5526  243206]]
------------------------------------------------------------
              precision    recall  f1-score   support

  Normal Flow       0.98      0.87      0.92   1981075
          SYN       0.77      0.99      0.87    753255
        RUDY        0.96      0.97      0.97    686631
   Slowloris        0.92      0.94      0.93    259713

    accuracy                           0.92   3680674
   macro avg        0.91      0.94      0.92   3680674
weighted avg        0.93      0.92      0.92   3680674

------------------------------------------------------------
```

Fig. 4. Exactitud, matriz de confusión y reporte por clases de Árbol de decisión con datos de prueba (30%).

Con este modelo se obtuvo un 91.7% de exactitud al momento de predecir las intrusiones en la red.

Para el modelo de KNN (K Nearest Neighbors) se obtuvo los siguientes resultados:

1. Para el set de datos de validación:

```
validation data:
------------------------------------------------------------
accuracy: 86.24 %
------------------------------------------------------------
confusion matrix
 [[884795  55598  30022   5806]
 [117906 252623     23      0]
 [ 27927     56 308643   1003]
 [  9136      4   1943 117384]]
------------------------------------------------------------
              precision    recall  f1-score   support

 Normal Flow       0.85      0.91      0.88    976221
         SYN       0.82      0.68      0.74    370552
        RUDY       0.91      0.91      0.91    337629
   Slowloris       0.95      0.91      0.93    128467

    accuracy                           0.86   1812869
   macro avg       0.88      0.85      0.87   1812869
weighted avg       0.86      0.86      0.86   1812869

------------------------------------------------------------
```

Fig. 5. Exactitud, matriz de confusión y reporte por clases de KNN con datos de validación (15%).

2. Para el set de datos de prueba:

```
test data:
------------------------------------------------------------
accuracy: 86.25 %
------------------------------------------------------------
confusion matrix
 [[1795166  112629   61313   11967]
 [ 239064  514133      57       1]
 [  56728     100  627757    2046]
 [  18271       8    3986  237448]]
------------------------------------------------------------
              precision    recall  f1-score   support

 Normal Flow       0.85      0.91      0.88   1981075
         SYN       0.82      0.68      0.75    753255
        RUDY       0.91      0.91      0.91    686631
   Slowloris       0.94      0.91      0.93    259713

    accuracy                           0.86   3680674
   macro avg       0.88      0.85      0.87   3680674
weighted avg       0.86      0.86      0.86   3680674

------------------------------------------------------------
```

Fig. 6. Exactitud, matriz de confusión y reporte por clases de KNN con datos de validación (30%)

Este segundo modelo obtuvo una exactitud de 86.25%. Lo cual es indicador que el modelo predice muy bien los datos y es aceptable para el objetivo del proyecto.

Para los datos de prueba, el modelo de árbol de decisión tiene el valor de precisión más bajo en las predicciones de SYN con 77% pero las demás predicciones con más de 92%, mientras que el modelo de KNN tiene el valor de precisión más bajo en predicciones de SYN con 82% y su valor más alto es de 94%. La precisión indica el valor de la proporción de los casos identificados como positivos fue correcta realmente.

Ahora bien, en el recall, el modelo de árbol de decisión tiene valores muy altos (más de 87% en todas las predicciones), mientras que el KNN tiene un 68% en predicciones de SYN y sus predicciones más altas son de 91%. El valor de recall dice qué proporción de los positivos fue identificada correctamente.

Es notable que el primer modelo predice de manera casi exacta todos los casos (91%). Pero esto no significa que el modelo sea bueno. Esto se debe a un pequeño problema de overfitting o sobreajuste.

El sobreajuste o overfitting significa que el modelo se ajusta tan bien a los datos que captura todas las complejidades de los datos, incluso el ruido o los valores atípicos también. El problema de tener overfitting o sobreajuste, es que el modelo es muy bueno pero solo para el set de datos con el cual fue entrenado, y debido a eso, el modelo no podría predecir correctamente datos nuevos o datos nunca antes vistos.

El overfitting o sobreajuste se debe al gran tamaño de los datos de entrenamiento, siendo estos el 55% de 12 millones, un total de 6.6 millones de observaciones aproximadamente y por la naturaleza del modelo en sí.

V. Conclusiones

El objetivo principal de este trabajo de investigación es la elaboración de dos modelos de machine learning sobre una base de datos de más de doce millones de registros relacionados con intrusiones de malware y stego-malware en una red.

Se crearon dos modelos de machine learning, uno de árbol de decisión y otro de KNN (K Nearest Neighbors). Después de realizar el trabajo necesario para limpiar y procesar los datos y después de la implementación de ambos modelos y análisis de resultados se llegan a las siguientes conclusiones:

- La matriz de correlación da una representación visual de las correlaciones entre las variables del dataset, de esta manera se pueden obtener las más significativas para el modelo.
- El modelo que es aceptable es el modelo de KNN. Debido a que tiene una buena exactitud. A pesar de que el modelo de árbol de decisión tiene mayor exactitud, este está sobre ajustado.
- La cantidad de observaciones de esta base de datos hace que los modelos al ser entrenados sufran de overfitting o los modelos se sobreajuste a los datos.

Referencias

[1] https://simargl.eu/download/SIMARGL-general_presentation.pdf

[2] Vázquez, A. M. (2018). Introducción a Machine Learning. https://unidad.gdl.cinvestav.mx/doc/investigacion/computacion/Introduccion-Machine-Learning.pdf