Predicción de Infección por Malware

Alexa Bravo[1]

[1]*Facultad de Ingenieria,*
Ingenieria en Ciencias de la Computacion y TI, Universidad del Valle de Guatemala,
Ciudad de Guatemala, Guatemala.

Today we are all vulnerable to a possible malware infection on the network if we do not have adequate protection. Malware is designed to interfere with the normal operation of a computer, programs that are used to infect systems and networks. It is a general term for and has different types of attacks that are used to spread to computer systems. This article describes the use of Machine Learning to achieve the detection of malware.

I. INTRODUCCIÓN

Los ataques de malware pueden provocar robos, encriptación, eliminación de datos personales o incluso el riesgo de ser espiado sin consentimiento. Malware es un término general para referirse a cualquier tipo de software malicioso diseñado para infiltrarse en un dispositivo sin su conocimiento. Existen diferentes tipos y cada uno busca lograr sus objetivos de manera diferente. En la mayoría de los casos, el malware es mucho más difícil de observar y trabaja con discreción en segundo plano. Una infección de malware puede darse por distintas maneras. Los principales son las dos más comunes en las que el malware obtiene acceso al sistema por medio del Internet y el correo electrónico, es decir, básicamente todo el tiempo que está conectado a Internet. Los ataques de malware no funcionarían sin el componente más importante, el usuario. Por ello, se han buscado formas de contrarrestarlos, proponiendo implementar y validar métodos innovadores de aprendizaje automático para detectarlos. A continuación se describe la implementación y los resultados de 'Random Forest', 'Árbol de Desición' y 'Regresión Logística', modelos de Machine Learning que se utilizaron para detectar el malware.

II. MARCO TEÓRICO

A. Random Forest Classifier

Es un algoritmo de aprendizaje supervisado y automático, que se utiliza para resolver problemas de clasificación y regresión en Machine Learning. Es un clasificador que contiene una serie de árboles de decisión en varios subconjuntos del conjunto de datos dado y toma el promedio para mejorar la precisión predictiva de ese conjunto de datos.

Este algoritmo funciona en dos fases: la primera es crear el bosque aleatorio combinando N árboles de decisión, y la segunda es hacer predicciones para cada árbol creado en la primera fase. Entre los sectores principales que utilizan Random Forest, están: bancos, medicina, uso de suelo y marketing.

Es un modelo capaz de manejar grandes conjuntos de datos con alta dimensionalidad además de aumenta la precisión del modelo y evita el problema del sobreajuste.

B. Árbol de Decisión

Es un modelo de aprendizaje supervisado, que se utiliza para resolver problemas de clasificación y regresión. Para los problemas de clasificación se emplea, cuando se quiere predecir el valor de una variable por medio de la clasificación de información en función de otras variables.

Se le conoce como "árbol de decisión"porque comienza con un nodo raíz, que se expande en ramas construyendo una arquitectura parecida a un árbol. Es una representación gráfica para obtener todas las posibles soluciones de un problema y/o decisión basada en las condiciones dadas. Son fáciles de construir, visualizar e interpretar.

Su estructura se basa en; nodos internos que son las características por considerar para tomar una decisión, las ramas que representan el rumbo de la decisión y los nodos finales que son el resultado de la decisión tomada.

C. Regresión Logística

Es un algoritmo de aprendizaje automático que forma parte del aprendizaje supervisado. Se utiliza para predecir la variable dependiente categórica utilizando un conjunto dado de variables independientes. La regresión logística predice la salida de una variable dependiente y brinda los valores probabilísticos que se encuentran entre 0 y 1 .

La regresión logística resulta útil para los casos en los que se desea predecir la presencia o ausencia de una característica o resultado según los valores de un conjunto de predictores.

Es muy parecida a la regresión lineal, pero se diferencia en el uso que se les da, ya que la regresión lineal se usa para resolver problemas de regresión, mientras que la regresión logística se utiliza para resolver los problemas de clasificación.

Existen tres tipos de regresión logística, que son: binomial que solo tiene dos tipos de variables dependientes, multinomial que tiene tres o más posibles variables dependientes y ordinal que tiene tres o más variables dependientes

III. METODOLOGÍA

A. Análisis Exploratorio

Se utilizaron los dos datasets brindados, el dataset de 'train' cuenta con 8921483 datos de tipo 'object', 'float' y 'int', distribuidos en ochenta y tres columnas. Mientras el dataset de 'test' cuenta con 7853253 datos de tipo 'object', 'float' y 'int', al igual que el anterior, con la diferencia de que estos estan distribuidos en ochenta y dos columnas.

B. Pre-prosesamiento

Se procesaron los datasets de manera separada, con ambos lo primero que se realizo fue buscar las columnas con más datos faltantes, con más ceros y la que tenía datos unicos, para esto ademas de utilizar codigo se utilizó de apoyo la página de Kaggle en donde brindan cierta informacion de los datos.

Se eliminaron las siguientes columnas:

```
IsBeta, IsSxsPassiveMode, AutoSampleOptIn, SMode, Census_HasOpticalDiskDrive,
Census_IsPortableOperatingSystem, Census_IsFlightsDisabled,Census_IsVirtualDevice,
Census_IsTouchEnabled,Census_IsPenCapable,Census_IsAlwaysOnAlwaysConnectedCapable,
MachineIdentifier,DefaultBrowsersIdentifier,PuaMode,Census_ProcessorClass,
Census_InternalBatteryType,Census_IsFlightingInternal,EngineVersion,AppVersion,
AvSigVersion,OsVer,OsBuildLab,SmartScreen,Census_ChassisTypeName,Census_OSVersion,
Census_OSBranch,Census_OSEdition,Census_OSSkuName
```

Ya que no eran necesarios por lo anteriormente mencionado.

Luego de esto se quitaron los datos tipo 'Object' y se llenaron los espacios vacios N/A con valor cero. Se implemento una función para dejar los valores de los datos 'float' solo con un decimal. Y el resultado fueron dos nuevos datasets 'train' y 'test' con 40 y 39 columnas respectivamente.

C. Selección de Características

Para esto se seleccionaron los datos que cumplen con las características necesarias para implementar los modelos. Para esto se utilizó el dataset 'train' de donde se crearon el grupo 'y' que contiene los valores de la columna 'HasDetections' y el grupo 'x' que contiene las demas columnas del dataset.

Luego se separaron los datos en train con el 55 % y test con el 45 %. Esto se debe a que hubo una nueva separación de datos que incluyen los de validation para obtener mejores resultados en la implementación de los modelos seleccionando, teniendo como resultado el 55 % train, 30 % test y 15 % validation.

Se escalaron y normalizaron los datos separados, como buena práctica.

D. Implementación de Modelos

Para la implementación de los modelos de Machine Learning primero se implementó una función que se utilizó para realizar las gráficas de Receiver Operating Characteristic Curve más conocidas por sus siglas ROC.

El primer modelo en implementarse fue el Random Forest Classificator. la realización de este lo primero fue definir el modelo, con los datos 'xTrain' y 'yTrain'. Luego se realizó una predicción del modelo y la matriz de confusión. Se realizó el calculó del accuracy con los datos de prueba. Por último, se imprimió el reporte de los resultados en donde se incluyen precision, recall, f1-score y support. Se repitió el mismo proceso con los datos de validación.

El siguiente modelo fue el Árbol de Decisión, la implementación de este se definió el modelo, con los datos de entrenamiento. Luego se realizó una predicción del modelo y la matriz de confusión con los datos de prueba y la predicción calculada. Se calculo el accuracy de los mismos datos y por último, observamos el reporte de resultados. Se repitió el mismo proceso con los datos de validación en lugar de los de prueba.

El tercer y ultimo modelo en implementarse fue la regresión logística, para este, se definió el modelo binomial. Se realizaron los cálculos de predicción del modelo junto con su matriz de prueba. Se calculo el accuracy de este modelo y por ultimo se observan los resultados del modelo. Este proceso se realizo con los datos de prueba y validación.

Para todos los modelos se realizo un Cross Validation, que es una técnica muy utilizada para validar los modelos generados, ya que nos brinda las medias de evaluación sobre diferentes particiones de los datos. También se realizó la grafica ROC correspondiente

IV. RESULTADOS

A. Random Forest

	precision	recall	f1-score	support
0.0	0.51	0.68	0.58	136
1.0	0.63	0.45	0.53	162
accuracy			0.56	298
macro avg	0.57	0.57	0.56	298
weighted avg	0.58	0.56	0.55	298

Figura 1: Resultados utilizando el grupo de datos 'test'

	precision	recall	f1-score	support
0.0	0.52	0.72	0.61	64
1.0	0.70	0.50	0.58	84
accuracy			0.59	148
macro avg	0.61	0.61	0.59	148
weighted avg	0.62	0.59	0.59	148

Figura 2: Resultados utilizando el grupo de datos 'validation'

```
array([0.56      , 0.65656566, 0.54545455, 0.61616162, 0.68686869,
       0.55555556, 0.57575758, 0.60606061, 0.5959596 , 0.6969697 ])
```

Figura 3: Resultados Cross Validation

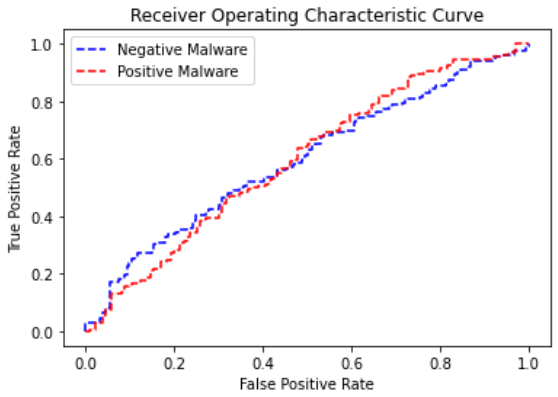

Figura 4: Gráfica de ROC

B. Arbol de Desición

	precision	recall	f1-score	support
0.0	0.50	0.54	0.52	136
1.0	0.59	0.54	0.56	162
accuracy			0.54	298
macro avg	0.54	0.54	0.54	298
weighted avg	0.55	0.54	0.54	298

Figura 5: Resultados utilizando el grupo de datos 'test'

	precision	recall	f1-score	support
0.0	0.48	0.55	0.51	64
1.0	0.61	0.55	0.58	84
accuracy			0.55	148
macro avg	0.55	0.55	0.54	148
weighted avg	0.56	0.55	0.55	148

Figura 6: Resultados utilizando el grupo de datos 'validation'

```
array([0.51      , 0.50505051, 0.4040404 , 0.47474747, 0.60606061,
       0.47474747, 0.50505051, 0.5959596 , 0.51515152, 0.48484848])
```

Figura 7: Resultados Cross Validation

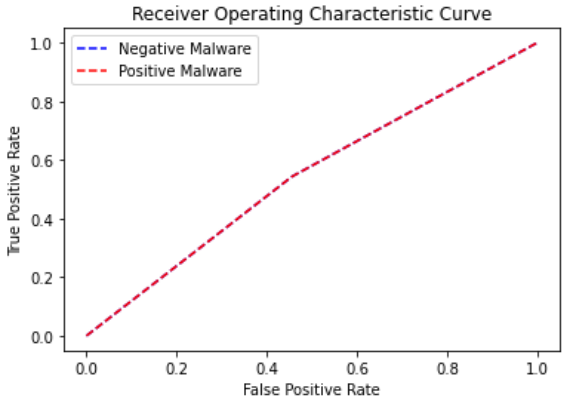

Figura 8: Gráfica de ROC

C. Regresión Logística

	precision	recall	f1-score	support
0.0	0.49	0.30	0.37	134
1.0	0.57	0.75	0.65	164
accuracy			0.55	298
macro avg	0.53	0.52	0.51	298
weighted avg	0.53	0.55	0.52	298

Figura 9: Resultados utilizando el grupo de datos 'test'

	precision	recall	f1-score	support
0.0	0.53	0.27	0.36	85
1.0	0.41	0.68	0.51	63
accuracy			0.45	148
macro avg	0.47	0.48	0.44	148
weighted avg	0.48	0.45	0.42	148

Figura 10: Resultados utilizando el grupo de datos 'validation'

```
array([0.53      , 0.52525253, 0.49494949, 0.51515152, 0.58585859,
       0.47474747, 0.49494949, 0.54545455, 0.63636364, 0.51515152])
```

Figura 11: Resultados Cross Validation

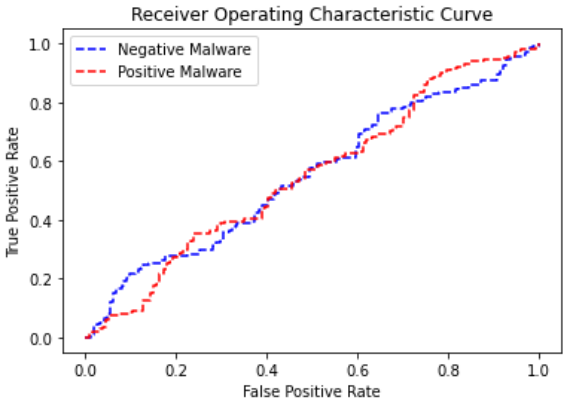

Figura 12: Gráfica de ROC

V. DISCUSIÓN

Se puede observar el desempeño de los tres algoritmos implementados, tanto con los datos de prueba, como los de validación. Los primeros resultados que se observan son los del algoritmo 'Random Forest' en donde podemos observar una precisión del 56 % para los datos de prueba y del 59 % para los datos de validación. El resultado del array de la validación cruzada nos dio una media de 0.6095 con las 10 particiones que se

realizaron y por último en la gráfica ROC observamos como es que empiezan y terminan en los mismos puntos la detección de malware.

Los siguientes resultados son los del modelo de un árbol de decisión, en donde se obtuvo una precisión del 54 % para los datos de prueba y una del 55 % para los datos de validación. La validación cruzada nos dio una media de 0.5075 de las diez particiones. Por último, en la gráfica ROC solo se observan los resultados positivos a una infección de malware.

Los ultimo los resultados de la regresión logística nos dan una precisión del 55 % para los datos de prueba y un 45 % para los datos de validación. La media del array obtenido fue de 0.5317 y en la grafica de ROC observamos como se van cruzando los resultados de una posible infección de malware.

Podemos notar que ningún modelo paso del 60 % de precisión y en el modelo de Random Forest y árbol de decisión la precisión para los datos de validación fue más alto que para los datos de prueba, cosa contraria con el modelo de regresión logística.

VI. CONCLUSIONES

- Es importante limpiar lo mejor posible el dataset para descartar todos los datos innecesarios que pueden solo retrasar la implementación de los modelos.

- Separar los datos en grupos de entrenamiento, prueba y validación, brinda mejores resultados en la implementación de los modelos.

- Para todos los modelos en necesario tomar en cuenta la posibilidad de sobreajuste.

- Los resultados de la media de validación cruzada son importantes porque nos brindan la precisión de la predicción de cada modelo.

VII. BIBLIOGRAFIA

[1] Avast Software. (1988-2022). Malware. https://www.avast.com/es-es/c-malware

[2] Brownlee, J. (2018). How to Use ROC Curves and Precision-Recall Curves for Classification in Python. Retrieved 26 February 2022, from https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

[3] JavaTpoint. (2011-2021). https://www.javatpoint.com/machine-learning-random-forest-algorithm

[4] JavaTpoint. (2011-2021). Random Forest Algorithm. https://www.javatpoint.com/machine-learning-random-forest-algorithm

[5] Malwarebytes. (2022). Malware. https://es.malwarebytes.com/malware

[6]Palo Alto Network. (2020). Malware. https://www.paloaltonetworks.com/cyberpedia/what-is-malware

[7] scikit-learn developers. (2007-2022). Modelos de Machine Learning. https://scikit-learn.org