An Empirical Comparison of Supervised Algorithms for Ransomware Identification on Network Traffic

Carlos Manzano
Escuela de Ingeniería
Universidad Católica del Norte
Coquimbo, Chile
cmanzano@ucn.cl

Claudio Meneses
Dpto. Ing. de Sistemas y Computación
Universidad Católica del Norte
Antofagasta, Chile
cmeneses@ucn.cl

Paul Leger
Escuela de Ingeniería
Universidad Católica del Norte
Coquimbo, Chile
pleger@ucn.cl

***Abstract*—Android mobile systems are currently the main target of malware attacks. In this sense, machine learning is a suitable approach to analyze network traffic, and it generally achieves good results in the identification and detection of malware. However, an underlying problem is creating a dataset with network characteristics that accurately reflect the malware´s behavior. Characterizing adequately the dataset is a relevant process to identify malware with high precision when using traditional machine learning algorithms. This paper compares empirically three supervised machine learning algorithms, in order to identify ransomware traffic based on Android mobile network traffic features. We consider 9 features related to time properties of flows and bidirectional packets in 10 families of ransomware and different benign application Android network traffic. Empirical results show that Random Forest (RF) achieved a 96% accuracy in classifying ransomware, higher than Decision Tree (DT) and K-Nearest Neighbor (KNN) approaches. We conclude that the selected features allow us to identify ransomware traffic and differentiate it from the traffic of benign applications.**

Keywords—Android, Traffic Identification, Features Selection, Ransomware, Machine Learning.

I. INTRODUCCIÓN

Actualmente Android de Google se ha convertido en el sistema operativo más popular en el mercado de teléfonos móviles [1]. Al mismo tiempo, ha existido un avance en el desarrollo de aplicaciones de usuarios para esta plataforma [2]. Sin embargo y debido a que el mercado de desarrollo de aplicaciones para Android es bastante grande y la comunidad de usuario es muy diversa, también se ha incorporado el desarrollo de aplicaciones maliciosas o malware con la principal intención de robar información y hacer daño a las personas [3]. Uno de los principales malware desarrollado para sistemas Android es el ransomware [4], el cual secuestra los datos de sus víctimas a través de la red mediante el cifrado de archivos y sólo entrega esta información a cambio de un pago, generalmente a través de bitcoin, por una clave de descifrado [5]. Según [6] y [7], ESET Threat reportó que la actividad del ransomware al final del segundo trimestre del año 2020, evidenció un aumento significativo de nuevos ataques de la campaña del ransomware WannaCryptor con un 47,9% en comparación con el 40,5% obtenido del primer trimestre del mismo año. Asimismo, para el segundo trimestre del año 2020, apareció CryCryptor, un nuevo tipo de ransomware Android que se hizo pasar por una aplicación de rastreo de COVID-19 proporcionada por el gobierno de Canadá [6]. Generar nuevas soluciones que permitan identificar los nuevos ataques de este tipo de malware, es un tema urgente que las comunidades de investigación en ciberseguridad deben abordar para prevenir la explotación y mal uso de estos sistemas.

Dado el crecimiento significativo del malware en teléfonos móviles [8], se proponen tres técnicas de análisis para la identificación y posterior detección de malware Android: análisis estático, análisis dinámico y análisis de red. El análisis estático se basa principalmente en el estudio de códigos fuentes de malware y es fácilmente evadido a través de la ofuscación de código [9]. El análisis dinámico se centra en utilizar las llamadas del sistema operativo para extraer información fiable de rastros de ejecución de malware [10]. Su principal desventaja es encontrar la trazabilidad exacta del comportamiento del malware por estar en un entorno controlado llamado sandbox [11]. Distinto a las técnicas de análisis estático y dinámico, que están basadas en el reconocimiento de código y comportamiento de malware dentro de un host [12]. El análisis de red ha planteado desafíos adicionales como el cifrado de datos y la ofuscación de puertos en el comportamiento de malware en red [13]. Casos prácticos de estos desafíos, han sido las aplicaciones Peer to Peer (P2P) [14] y las herramientas de detección ataques de ransomware como R-PackDroid [9]. Según [8], el análisis de red permite identificar el comportamiento de malware de acuerdo con las características directas o pasivas de observaciones de un flujo de red. Una de las modernas técnicas de análisis de red para identificar malware es la clasificación de tráfico de red con aprendizaje automático [15]. En [16], [2] , [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [3], [28], [29], [30], [31], [32], [12] y [33]–[36], la técnica de análisis de tráfico de red con aprendizaje automático ha demostrado buenos resultados en la identificación y posterior detección de malware. Sin embargo, un problema común con esta técnica es la necesidad de crear un conjunto de datos con características de red que reflejen con exactitud el comportamiento del malware [2].

En particular, este artículo evalúa de forma empírica el rendimiento de tres conocidos algoritmos supervisados para realizar la tarea de identificar tráfico de ransomware a partir de una selección y combinación de 15 características de propiedades de tiempos de flujos y características de paquetes bidireccionales de tráfico de red. Apoyado por un enfoque de proyectos de aprendizaje automático [37], se seleccionaron observaciones de diez familias de ransomware y de aplicaciones benignas, desde un conjunto de datos llamado CICAndMal2017 con tráfico no encriptado de red móvil Android [38]. Se utilizó el método de correlación de Kendall para seleccionar y reducir la dimensionalidad del conjunto de datos inicial a 9 características más significativas del tráfico de red [39]. Posteriormente, se ejecutaron tres algoritmos supervisados para obtener los modelos de clasificación de

978-1-7281-8328-2/20/$31.00 ©2020 IEEE

tráfico de red de ransomware. El resto del artículo se estructura de la siguiente manera. En la sección II, se exponen los trabajos relacionados con la identificación de malware basado en análisis de tráfico de red y aprendizaje automático. En la sección III, se explica la metodología de trabajo relacionada con la recopilación y de tráfico de ransomware junto a la selección de características. En la sección IV, se realiza la evaluación empírica de los algoritmos de clasificación para la identificación de ransomware y se discute los resultados obtenidos. Finalmente, se presentan las conclusiones y los trabajos futuros.

II. Trabajos Relacionados

Desde hace algunos años, la investigación acerca de clasificación de tráfico de red ha comenzado a buscar técnicas que no se basen en análisis de puertos o en inspección profunda de paquetes [40]. Sino que utilicen las características estadísticas del tráfico de red para ayudar al proceso de clasificación e identificación de malware [41]. Según [15], la técnica de clasificación de tráfico de red basada en aprendizaje automático supervisado ha logrado buenos resultados en la identificación y eventual detección de malware. Sin embargo, un desafío común de la clasificación de tráfico de red con aprendizaje automático supervisado es la necesidad de crear un conjunto de datos con características de red que reflejen con exactitud el comportamiento del malware [42]. Los estudios realizados por [43] y [44], manifiestan que existen dos tipos de categorías de características de tráfico de red para la técnica de clasificación con aprendizaje automático. Las características de tráfico de red basadas en propiedades de tiempos flujos y las características de cada paquete de red. Caracterizar de forma adecuada el conjunto de datos por parte del experto, es una tarea relevante para obtener alta precisión en el proceso en la identificación de malware [16]. A continuación, se presentan trabajos relacionados con la clasificación de tráfico de red con aprendizaje automático supervisado, junto a la selección de características de red para la identificación y detección de tráfico de malware.

En el ámbito de la identificación y detección malware Android conocidos [45], se han realizado trabajos como el de [2], donde los autores utilizaron dos conjuntos de datos de malware llamados Drebin y Contagiodumpset, junto a un conjunto de datos de software benigno de Google Playstore para realizar el análisis y detección de malware Android. Se seleccionaron siete características de propiedades de tiempo de flujo, junto a características de paquetes de red para ser analizados y clasificados a través de un algoritmo Decision Tree con WEKA. El conjunto de datos Drebin ofreció una precisión de detección de malware de un 98% para el clasificador J48. En [20], se aplicaron tres algoritmos de aprendizaje automático supervisado y un conjunto de datos relacionado con quince características de propiedades de tiempo de flujo, junto a características de paquetes de red para resolver el problema de caracterización e identificación del tráfico de malware Android. Los resultados experimentales mostraron que aplicando una proporción 8:2 de observaciones de entrenamiento y evaluación, los algoritmos Random Forest, Decision Tree y K-Nearest Neighbors lograron obtener resultados del 95% de precisión global cuando se realizó la clasificación binaria entre tráfico de malware y tráfico de software benigno. Sin embargo, cuando se realizó la clasificación con multiclases de malware, se obtuvo un rendimiento de precisión global del 90\% para los tres clasificadores. Además, los autores aportaron evidencias que, en la literatura, existen sólo algunos trabajos acerca de clasificación de tráfico de malware que han utilizado la combinación de propiedades de tiempo de flujo, con características de paquetes de red para identificar tráfico de malware. En [22], se aportan conclusiones sobre el análisis de tráfico de red para la detección malware Android en función de sus características propiedades de tiempo de flujo. Los resultados experimentales muestran que un clasificador basado en reglas es notablemente preciso y detecta más del 90% de las muestras de tráfico de malware Android. También en [21], se utilizó un conjunto de datos con 217 muestras de malware y software benigno, además de seleccionar características basadas en propiedades de tiempo de flujo de red para la detección de malware Android. El clasificador Decision Tree logró predecir correctamente más del 90% de las muestras de tráfico de red de malware. El clasificador sólo falló en las observaciones que estaban cifradas por alguna técnica de ofuscación de malware. En [27], se evaluaron los clasificadores Bayes Network, Multi-layer Perceptron, J48, K Nearest Neighbours y Random Forest para la detección de 49 familias de malware Android. Se seleccionaron cuatro categorías de tráfico basadas en contenido, tiempo y propiedades de flujo de red. La evaluación fue realizada con un conjunto de datos públicos llamado MalGenome y otro privado creado por los autores. Tanto el clasificador Bayes como el clasificador Random Forest lograron tasas de verdaderos positivos del 99,97% en comparación con Perceptron Multi-layer con solo 93,03%, en el conjunto de datos MalGenome. En el ámbito de la identificación y detección de ransomware, Alhawi et al. [17], analizaron la selección de características de propiedades de tiempo de flujo de tráfico de red para ransomware de plataforma Windows. Los autores lograron una precisión del 97,1% de detección de ransomware utilizando el clasificador Decision Tree J48. En el campo de la identificación de malware Android desconocidos, Bekerman et al. [24], presentaron un sistema basado en supervisión extremo a extremo para detectar el tráfico de malware. El método incluyó un algoritmo de selección de características basados en correlación (CFS) para elegir las características más significativas y reducir la dimensionalidad del conjunto de datos. Los resultados experimentales, muestran que inicialmente se logró extraer 972 características de propiedades de flujo de red y luego se redujo a sólo doce de ellas con el método de correlación. Utilizando validación cruzada con diez carpetas, los clasificadores Random Forest, Decision Tree y Naive Bayes, lograron una precisión global sobre el 90%. Desde otro punto de vista, en [46], [47] y [48] se seleccionaron características de paquetes de red para la clasificación de tráfico Peer to Peer (P2P), Web Real Time Communication (WebRTC) y mensajeria iMessage mediante los clasificadores Decision Tree C4.5, Naive Bayes y Random Forest.

III. Metodología

A. Conjunto de Datos

Los datos de tráfico de red de ransomware y de aplicaciones benignas de Android, corresponden al conjunto de datos llamado CICAndMal2017 obtenido y preparado

desde el trabajo realizado por [38]. El tráfico de ransomware corresponde a un archivo CSV (Comma Separated Values) con tráfico de red de diez familias del malware (Ver Tabla I). El tráfico de las aplicaciones benignas corresponde a un archivo CSV generados por 1500 aplicaciones de Android de Google Play. Ambos archivos CSV están estructurados con 15 características de tráfico de red y 1 característica con la clase del tipo ransomware o benigno que fueron elegidas desde [20] (Ver Tabla II).

TABLA I. DETALLES DE FAMILIAS DE RANSOMWARE

N°	Familias de Ransomware
1	Harger
2	Jisut
3	Ko|ler
4	LockerPin
5	Simplocker
6	Pletor
7	PornDroid
8	RansomBO
9	Svpeng
10	WannaLocker

TABLA II. CARACTERÍSTICAS DE TRÁFICO DE RED DEL CONJUNTO DE DATOS INICIAL

N°	Característica	Descripción
1	Flow Duration	Duración de flujos en microsegundos
2	Flow Byts/s	Número de flujos de bytes por segundo
3	Tot Fwd Pkts	Total de paquetes en al dirección de salida
4	Tot Bwd Pkts	Total de paquetes en dirección de entrada
5	Fwd Pkt Len Min	Tamaño mínimo de paquetes de salida
6	Fwd Pkt Len Max	Tamaño máximo del paquete de salida
7	Fwd Pkt Len Mean	Tamaño Medio del paquete de salida
8	Fwd Pkt Len Std	Desviación estándar de paquete de salida
9	Bwd Pkt Len Min	Mínimo tamaño de paquete de entrada
10	Bwd Pkt Len Max	Tamaño máximo de paquete de entrada
11	Bwd Pkt Len Mean	Tamaño Medio del paquete de entrada
12	Bwd Pkt Len Std	Desviación estándar de paquetes de entrada
13	Tot Len Fwd Pkts	Tamaño total de paquete de salida
14	Tot Len Bwd Pkts	Tamaño total de paquetes de entrada
15	Flow Pkts/s	Número de flujos de paquetes por segundos
16	Label	Etiqueta clase tipo Ransomware o Benigno

B. Análisis y preprocesamiento

Se propone una arquitectura de aprendizaje automático que está basada en [37] para guiar el proceso de clasificación de tráfico de red para la identificación de ransomware. Esta arquitectura consiste en una etapa de análisis y preprocesamiento de los datos, selección de características y entrenamiento de los clasificadores de aprendizaje automático. La Fig. 1 muestra la arquitectura propuesta para el proceso de clasificación de tráfico de red para la identificación de ransomware Android.

La técnica de clasificación del tráfico de red basada en aprendizaje automático debe dividir el tráfico continuo en unidades discretas según cierta granularidad [49]. En la etapa de análisis y preprocesamiento (Ver Fig. I), se obtuvo y combinó el tráfico de red de los conjuntos de datos CSV de ransomware y de aplicaciones benignas mediante una aplicación programada con librerías Sklearn del lenguaje de programación Python. El tamaño total del conjunto de datos consolidado es de 11,2MB en formato CSV. Las características del conjunto de datos ya consolidado corresponden a 15 características estadísticas de sesiones de tiempo de protocolo de capa de transporte (Ver Tabla II). Una sesión es un flujo bidireccional donde una IP de origen y una IP de destino pueden intercambiar paquetes de entrada y salida. Para cada flujo de red se consideran tres categorías de paquetes: paquetes entrantes, paquetes salientes y tráfico bidireccional (es decir, el total de paquetes entrantes y salientes). No se aplicó normalización a los datos, ya que este proceso fue realizado inicialmente en el trabajo de [38]. Además, en [38], los autores descartan los paquetes que contienen retransmisiones de TCP (Transmission Control Protocol) u otros errores.

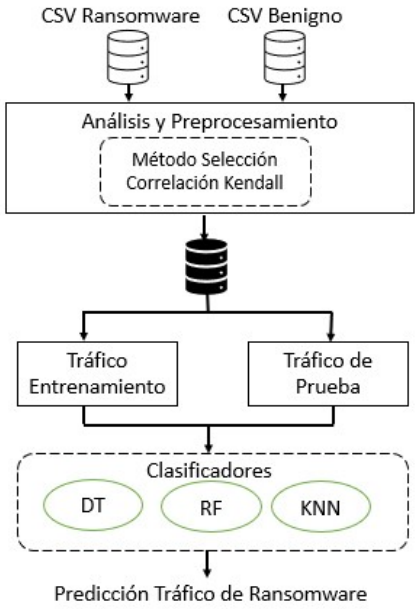

Fig. 1. Arquitectura de aprendizaje automático para la clasificación e identificación de ransomware.

La selección de características más significativas del conjunto de datos fue realizada a través del método de correlación Kendall. El método fue seleccionado por su buen rendimiento en [39]. Los resultados del método de correlación descartaron 6 de 15 características relacionadas con el fujo en bytes y paquetes por segundos (Flow Byts/s, Flow Pkts/s), el tamaño medio de paquetes de entrada y salida (Bwd Pkt Len Mean, Fwd Pkt Len Mean), y el tamaño estándar de paquetes de entrada y salida (Bwd Pkt Len Std, Fwd Pkt Len Std), Por lo tanto, se mantiene la característica del tipo clase y se reduce a 9 las características asociadas al tráfico de red del conjunto de datos original (Ver Tabla III).

TABLA III. CARATERÍSTICAS DE TRÁFICO DE RED SELECCIONADAS POR EL MÉTODO DE KENDALL

N°	Característica	Descripción
1	Flow Duration	Duración de flujos en microsegundos
2	Tot Fwd Pkts	Total de paquetes en la dirección de salida
3	Tot Bwd Pkts	Total de paquetes en dirección de entrada
4	Fwd Pkt Len Min	Tamaño mínimo de paquetes de salida
5	Fwd Pkt Len Max	Tamaño máximo de paquetes de salida
6	Bwd Pkt Len Min	Tamaño mínimo de paquetes de entrada
7	Bwd Pkt Len Max	Tamaño máximo de paquetes de entrada
8	Tot Len Fwd Pkts	Tamaño total de paquetes de salida
9	Tot Len Bwd Pkts	Tamaño total de paquetes de entrada
10	Label	Etiqueta clase tipo Ransomware o Benigno

C. Entrenamiento de clasificadores

Se eligieron tres algoritmos de aprendizaje automático supervisado para la clasificación de tráfico de red con el objetivo de identificar tráfico de ransomware. En la

literatura los algoritmos Random Forest, Decision Tree y K-Nearest Neighbors han mostrado buen rendimiento en la clasificación de tráfico de red con características de propiedades de tiempo y paquetes de flujo de red [37] y [50]. Random Forest es un conjunto (ensemble) de árboles de decisión combinados con bagging. Esto quiere decir que distintos árboles ven distintas porciones de los datos. Ningún árbol ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema. De esta forma, al combinar sus resultados, unos errores se compensan con otros y tenemos una predicción que generaliza mejor [51]. K-Nearest Neighbors clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias [52]. El algoritmo Decision Tree es un modelo de predicción. En un árbol de decisión cada nodo del árbol es un atributo (campo) de los ejemplos, y cada rama representa un posible valor de ese atributo. El proceso de generación de un árbol de decisión se divide principalmente en las siguientes tres partes: selección de características, construcción del árbol de decisión, poda del árbol de decisión [21]. Para el proceso experimental del algoritmo Decision Tree, se utilizó el coeficiente de Gini para selección de características [53]. En la Ecuación (1), cuanto mayor sea el grado de mezcla de características en el conjunto de datos, mayor será el índice de Gini. Cuando el conjunto de datos D tiene solo una categoría, el índice de Gini tiene un valor mínimo de 0. Si el atributo seleccionado es A, entonces la fórmula de cálculo para el índice de Gini del conjunto de datos D se expresa como:

$$Gini_A(D) = 1 - \sum_{1}^{k} \frac{D_j}{D} Gini(D_j) \quad (1)$$

IV. Experimentos y Resultados

Esta sección presenta la evaluación de forma empírica del rendimiento de tres conocidos algoritmos de aprendizaje automático supervisado, para identificar el tráfico de ransomware a partir de una selección y combinación de características de tráfico de red móvil Android. Brevemente, se presentan el diseño de los experimentos y las métricas de evaluación para el análisis de rendimiento de los clasificadores. Luego se discuten los resultados experimentales. Dos experimentos se definieron para realizar la tarea de clasificación. Cada uno de los experimentos presentados en este artículo incorporan tres tareas de clasificación binaria e incluyen observaciones de tráfico de red de ransomware y de aplicaciones benignas. El experimento N°1 está compuesto por el conjunto de datos inicial con 15 características de tráfico de red y 1 característica del tipo clase. El experimento N°2 está compuesto por las 9 características de tráfico de red más significativas obtenidas desde el conjunto de datos inicial por el método de Kendall, Además, de considerar 1 característica del tipo clase para la rotulación de sus observaciones. Para cada tarea de clasificación realizada, la proporción del conjunto de entrenamiento y prueba es de 8:2. La Tabla IV presenta la distribución de las observaciones de tráfico de red en el conjunto de datos.

TABLA IV. DISTRIBUCIÓN DE OBSERVACIONES DE TRÁFICO DE RED EN EL CONJUNTO DE DATOS

Tipo	Ransomware	Benigno
Total de observaciones	38.159	137.105
Total de observaciones entrenamiento	30.527	109.684
Total de observaciones prueba	7.632	27.421

A. Métricas de evaluación

Se utilizó la matriz de confusión para evaluar el rendimiento de los algoritmos de aprendizaje automático supervisado. La matriz de confusión contiene tres métricas fundamentales, precision (P), recall (R), f-measure (F). Estas métricas de confusión se componen de verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). Específicamente para el proceso de identificación de tráfico de red, (TP) y (TN) son el número de instancias que predicen correctamente si es ransomware o aplicación benigna. Por otro lado, (FP) y (FN) son el número de instancias que se predicen incorrectamente como ransomware o aplicación benigna.

- Precisión (P): presenta el porcentaje de todas las muestras predichas como tráfico de ransomware que es realmente ransomware.

$$P = \frac{TP}{TP + FP} \quad (2)$$

- Recall (R): presenta el porcentaje de todas las muestras de tráfico de ransomware que se prevé que sean realmente ransomware.

$$R = \frac{TP}{TP + FN} \quad (3)$$

- F-Score $(F1)$: el valor F1 es la media armónica de precisión y recall que puede ser mejor para evaluar el rendimiento.

$$F1 = \frac{2PR}{P + R} \quad (4)$$

B. Resultados Experimentales

En el experimento N°1, basado en el conjunto de datos inicial, se encontró que los valores de todas las métricas de evaluación, lograda por los tres algoritmos, estaban por encima del 80%. Para la clasificación de ransomware, Random Forest obtuvo el mejor desempeño, con una precisión del 91%, recall del 97% y valor F1 del 94%. Decision Tree obtuvo un rendimiento ligeramente más bajo de precisión con un 90%, recall del 92% y valor F1 del 91%. K-Neighbors obtuvo el peor desempeño, con una precisión del 83%, recall del 86% y valor F1 del 84% (Ver Fig.2). Respecto a la precisión global (Accuracy) en la clasificación de ransomware y aplicaciones benignas, Random Forest logró un 96% sobre lo obtenido por Decision Tree (DT) y K-Nearest Neighbors (KNN) (Ver Fig 3.). El experimento N°2, diseñado a partir de las 9 características de red obtenidas por el método de Kendall desde el conjunto de datos inicial. Encontró que los valores de todas las métricas de evaluación, lograda por los tres algoritmos, estaban por encima del 80%. Para la clasificación de ransomware, Random Forest

mantuvo el mejor desempeño, con precisión del 91%, recall del 97% y valor F1 del 94%. Decision Tree mejoró su rendimiento de recall con un 94% sobre el 92% de recall obtenido en el experimento N°1. El valor F1 se mantuvo en un 91%. K-Neighbors mantuvo su bajo desempeño con una precisión del 83%, recall del 86%, valor F1 del 84%. Respecto a la precisión global (Accuracy) en la clasificación de ransomware y aplicaciones benignas, Decision Tree (DT) obtuvo un mejor rendimiento con un 94% respecto al experimento N°1 con un 90%. El algoritmo K-Nearest Neighbors (KNN) obtuvo ligeramente un mejor resultado con un 90% sobre el 89% del experimento N°1 (Ver Fig 5.). Los resultados de la clasificación de ransomware para el experimento N°2 se presentan en la Fig 4.

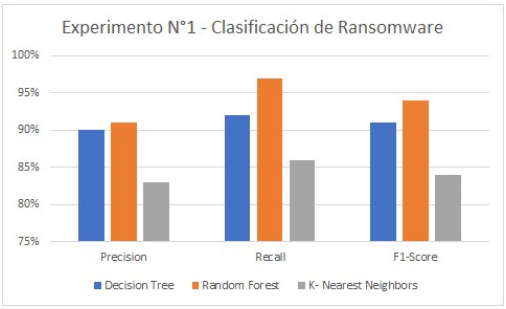

Fig. 2. Resultados de la clasificación de ransomware para el experimento N°1.

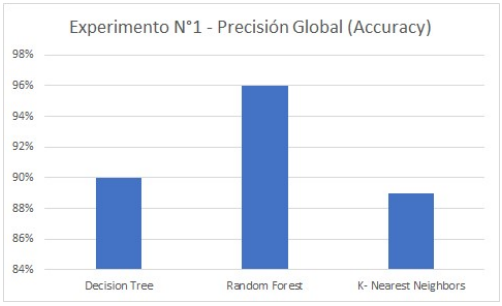

Fig. 3. Resultados de la clasificación global (accuracy) entre ransomware.y aplicaciones benignas para el experimento N°1.

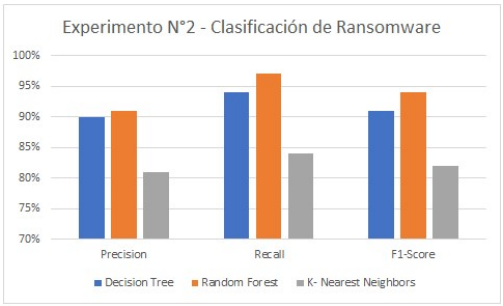

Fig. 4. Resultados de la clasificación de ransomware para el experimento N°2.

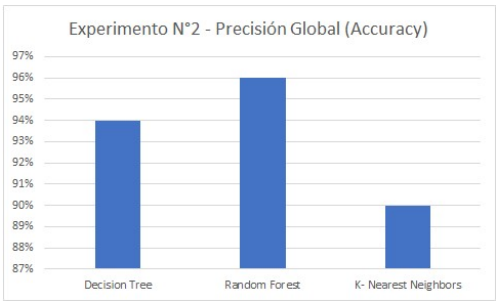

Fig. 5. Resultados de la clasificación global (accuracy) entre ransomware.y aplicaciones benignas para el experimento N°2.

V. Conclusiones y Trabajos Futuros

Este artículo evalúa de forma empírica el rendimiento de tres conocidos algoritmos de aprendizaje automático supervisado para identificar el tráfico de ransomware a partir de una selección y combinación de características de tráfico de red móvil Android. Se ha estudiado la identificación de propiedades relacionada con el tiempo de flujo y las características de paquete de red para abordar el problema de la caracterización del tráfico de ransomware. Se utilizó el método de correlación Kendall para obtener las características más significativas de tráfico de red y reducir la dimensionalidad del conjunto de datos inicial. Los resultados experimentales muestran que la técnica de clasificación de tráfico de red basada en Random Forest, obtuvo la mejor identificación de tráfico de ransomware y el tráfico benigno con un promedio de precisión global (Accuracy) del 96% sobre el resto de los algoritmos de clasificación. Asimismo, Random Forest obtuvo el mejor rendimiento de precisión en la clasificación de ransomware con un 91% en ambos experimentos. Decision Tree logró un buen rendimiento de precisión global del 94% (Accuracy) y una identificación del 94% de casos correctamente clasificados de ransomware (recall) cuando se experimentó con características relacionadas con la duración del flujo, el tamaño máximo, mínimo y total de paquetes de entrada del tráfico de red. El resultado más bajo de clasificación de ransomware en torno a la precisión, lo obtuvo K-Nearest Neighbors en ambos experimentos con un 83%. Sin embargo, K-Nearest Neighbors presentó ligeramente una mejor precisión global (Accuracy) del 90% en el segundo experimento con las características seleccionadas por el método de Kendall. Como trabajo futuro se considerará mejorar la tasa de identificación de ransomware y de aplicaciones benignas a través de experimentos basados en validación cruzada. Se abordará métodos de balanceo de clases para lograr un trabajo más eficiente en la clasificación de ransomware. Por último, como la evolución de los ataques del ransomware de Android es rápida, las características de nuestro conjunto de datos puede no ser práctica para futuros casos de este malware. Por lo tanto, estudiar los métodos de aprendizaje profundo puede ser una buena alternativa para la clasificación e identificación del tráfico de nuevos casos de ransomware, dado que este enfoque no depende de características predefinidas, sino que las construye internamente como parte del proceso de aprendizaje profundo.

REFERENCIAS

[1] Statista, "Statista." [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] A. Zulkifli, I. R. A. Hamid, W. M. Shah, and Z. Abdullah, "Android malware detection based on network traffic using decision tree algorithm," Adv. Intell. Syst. Comput., vol. 700, no. January, pp. 485–494, 2018.

[3] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," J. Netw. Comput. Appl., vol. 153, no. November 2019, p. 102526, 2020.

[4] S. Alsoghyer and I. Almomani, "Ransomware detection system for android applications," Electron., vol. 8, no. 8, pp. 1–36, 2019.

[5] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," Comput. Secur., vol. 74, pp. 144–166, 2018.

[6] ESET, "Threat Report Q2 2020," Eset, vol. 1, p. 40, 2020.

[7] ACSC, "ESET Threat Report Q1 2020," Aust. Cyber Secur. Cent., vol. 1, p. 40, 2019.

[8] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, "The evolution of android malware and android analysis techniques," ACM Comput. Surv., vol. 49, no. 4, 2017.

[9] M. Scalas, D. Maiorca, F. Mercaldo, C. A. Visaggio, F. Martinelli, and G. Giacinto, "On the effectiveness of system API-related information for Android ransomware detection," Comput. Secur., vol. 86, pp. 168–182, 2019.

[10] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," Futur. Gener. Comput. Syst., vol. 90, pp. 211–221, Jan. 2019.

[11] J. Stiborek, T. Pevný, and M. Rehák, "Probabilistic analysis of dynamic malware traces," Comput. Secur., vol. 74, pp. 221–239, 2018.

[12] J. Baldwin and A. Dehghantanha, "Leveraging support vector machine for opcode density based detection of crypto-ransomware," Adv. Inf. Secur., vol. 70, pp. 107–136, 2018.

[13] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," IEEE Commun. Mag., vol. 57, no. 5, pp. 76–81, 2019.

[14] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," Proc. - IEEE Comput. Soc. Annu. Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst. MASCOTS, pp. 179–188, 2006.

[15] E. Biersack, F. Measurement, and D. Hutchison, LNCS 7754 - Data Traffic Monitoring and Analysis. Springer Berlin Heidelberg, 2013.

[16] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Toward effective mobile encrypted traffic classification through deep learning," Neurocomputing, vol. 409, pp. 306–315, 2020.

[17] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection," Cyber Threat Intell. Adv. Inf. Secur., no. https://doi.org/10.1007/978-3-319-73951-9_5, 2018.

[18] Z. Chen et al., "Machine learning based mobile malware detection using highly imbalanced network traffic," Inf. Sci. (Ny)., vol. 433–434, pp. 346–364, 2018.

[19] S. Wang et al., "Deep and Broad Learning Based Detection of Android Malware via Network Traffic," 2018 IEEE/ACM 26th Int. Symp. Qual. Serv. IWQoS 2018, 2019.

[20] R. Chen, Y. Li, and W. Fang, "Android Malware Identification Based on Traffic Analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11632 LNCS, pp. 293–303, 2019.

[21] D. Sharma, "Android Malware Detection using Decision Trees and Network Traffic," Int. J. Comput. Sci. Inf. Technol., vol. 7, no. 4, pp. 1970–1974, 2016.

[22] A. Arora, S. Garg, and S. K. Peddoju, "Malware detection using network traffic analysis in android based mobile devices," Proc. - 2014 8th Int. Conf. Next Gener. Mob. Appl. Serv. Technol. NGMAST 2014, pp. 66–71, 2014.

[23] R. Moussaileb, N. Cuppens, J. L. Lanet, and H. Le Bouder, "Ransomware Network Traffic Analysis for Pre-encryption Alert," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12056 LNCS, pp. 20–38, 2020.

[24] D. Bekerman, B. Shapira, L. Rokach, and A. Bar, "Unknown malware detection using network traffic classification," 2015 IEEE Conf. Commun. NetworkSecurity, CNS 2015, pp. 134–142, 2015.

[25] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android Encrypted Network Traffic to Identify User Actions," IEEE Trans. Inf. Forensics Secur., vol. 11, no. 1, pp. 114–125, 2016.

[26] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahn, and C. Glezer, "Applying machine learning techniques for detection of malicious code in network traffic," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 4667 LNAI, pp. 44–50, 2007.

[27] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," Soft Comput., vol. 20, no. 1, pp. 343–357, 2016.

[28] M. Stevanovic and J. M. Pedersen, "On the use of machine learning for identifying botnet network traffic," J. Cyber Secur. Mobil., vol. 4, no. 2–3, pp. 1–32, 2015.

[29] J. Kohout, T. Komárek, P. Čech, J. Bodnár, and J. Lokoč, "Learning communication patterns for malware discovery in HTTPs data," Expert Syst. Appl., vol. 101, pp. 129–142, 2018.

[30] S. Katal and A. P. H. Singh, "A Survey of Machine Learning Algorithm in Network Traffic Classification," Int. J. Comput. Trends Technol., vol. 9, no. 6, pp. 301–304, 2014.

[31] J. Modi and B. Eng, "Detecting Ransomware in Encrypted Network Traffic Using Machine Learning," 2019.

[32] R. Kumar and T. Kaur, "Machine Learning based Traffic Classification using Low Level Features and Statistical Analysis," Int. J. Comput. Appl., vol. 108, no. 12, pp. 6–13, 2014.

[33] S. Talukder and Z. Talukder, "A Survey on Malware Detection and Analysis Tools," Int. J. Netw. Secur. Its Appl., vol. 12, no. 2, pp. 37–57, 2020.

[34] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," IEEE Access, vol. 3, pp. 1132–1142, 2015.

[35] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. R. Ma'arof, and S. Shamshirband, "A study of machine learning classifiers for anomaly-based mobile botnet detection," Malaysian J. Comput. Sci., vol. 26, no. 4, pp. 251–265, 2013.

[36] S. Garg, S. K. Peddoju, and A. K. Sarje, "Network-based detection of Android malicious apps," Int. J. Inf. Secur., vol. 16, no. 4, pp. 385–400, 2017.

[37] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," IEEE Access, vol. 8, pp. 124579–124607, 2020.

[38] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification," Proc. - Int. Carnahan Conf. Secur. Technol., vol. 2018-Octob, no. Cic, pp. 1–7, 2018.

[39] Z. Zhu and T. Dumitras, "FeatureSmith: Automatically engineering features for malware detection by mining the security literature," Proc. ACM Conf. Comput. Commun. Secur., vol. 24-28-Octo, pp. 767–778, 2016.

[40] T. T. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using ML," Ieee Comst, vol. 10, no. 4, pp. 56–76, 2008.

[41] G. Xiaolin, L. Jun, L. Chenyu, L. Qiujian, and L. Zhenming, "Analysis of Malware Application Based on Massive Network Traffic," J. China Univ. Posts Telecommun., vol. 23, no. 3, pp. 70–75, 2016.

[42] Z. Berkay Celik, R. J. Walls, P. McDaniel, and A. Swami, "Malware traffic detection using tamper resistant features," Proc. - IEEE Mil. Commun. Conf. MILCOM, vol. 2015-Decem, pp. 330–335, 2015.

[43] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," Int. J. Netw. Manag., 2015.

[44] R. Alshammari and A. N. Zincir-Heywood, "An investigation on the identification of voip traffic: Case study on Gtalk and Skype," Proc. 2010 Int. Conf. Netw. Serv. Manag. CNSM 2010, pp. 310–313, 2010.

[45] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, and M. Conti, "Android Security : A Survey of Issues , Malware Penetration and Defenses," vol. 00, no. 0, pp. 1–27, 2015.

[46] D. Wang, L. Zhang, Z. Yuan, Y. Xue, and Y. Dong, "Characterizing application behaviors for classifying P2P traffic," 2014 Int. Conf. Comput. Netw. Commun. ICNC 2014, pp. 21–25, 2014.

[47] S. Coull, K. Dyer, J. Sommers, S. E. Coull, and K. P. Dyer, "Public Review for Traffic Analysis of Encrypted Messaging Services : Apple iMessage and Beyond a c m s i g c o m m Traffic Analysis of Encrypted Messaging Services : Apple iMessage and Beyond," vol. 44, no. 5, pp. 5–11, 2014.

[48] M. Di Mauro and M. Longo, "Revealing encrypted WebRTC traffic via machine learning tools," SECRYPT 2015 - 12th Int. Conf. Secur. Cryptogr. Proceedings; Part 12th Int. Jt. Conf. E-bus. Telecommun. ICETE 2015, pp. 259–266, 2015.

[49] G. Dogan and T. Brown, "A Survey of Methods for Encrypted Traffic Classification and Analysis," Int. J. Netw. Manag., pp. 17–31, 2014.

[50] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification," Proc. - Int. Carnahan Conf. Secur. Technol., vol. 2018-Octob, no. Cic, 2018.

[51] Y. L. Pavlov, "Random forests," Random For., pp. 1–122, 2019.

[52] S. Hahn, M. Protsenko, and T. Müller, “Comparative evaluation of machine learning-based malware detection on Android,” Sicherheit 2016 Sicherheit, Schutz und Zuverl{ä}ssigkeit, Beitr{ä}ge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft f{ü}r Inform. e.V. (GI), 5.-7. April 2016, Bonn, pp. 79–88, 2016.

[53] L. Čehovin and Z. Bosnić, “Empirical evaluation of feature selection methods in classification,” Intell. Data Anal., vol. 14, no. 3, pp. 265–281, 2010.