Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Ciberseguridad
Recibido: 31/05/2018 | Aceptado: 13/09/2018

Mejorando el rendimiento en la detección de tráfico scan y backscatter

Improving scan and backscatter detection performance

Vitali Herrera-Semenets[1*], Christian Doerr[2], Osvaldo Andrés Pérez-García[1], Raudel Hernández-León[1]

[1]Advanced Technologies Application Center (CENATAV). 7a ♯ 21406, Playa, C.P. 12200, Havana, Cuba
[2]Intelligent Systems Department, Delft University of Technology. Mekelweg 4, 2628 CD Delft, the Netherlands

*Autor para correspondencia: vherrera@cenatav.co.cu

Resumen
El crecimiento exponencial del volumen de datos generado por la prestación de servicios en las redes de telecomunicaciones hace que sea cada vez más complejo su procesamiento. La presencia de datos innecesarios, datos redundantes o ruidosos, puede afectar el rendimiento de los Sistemas de Detección de Intrusos (IDS). El empleo de técnicas de Minería de Datos que permiten reducir la información innecesaria se ha hecho frecuente en estos escenarios. Sin embargo, llevar a cabo el proceso de reducción sin afectar la eficacia del proceso de detección, sigue siendo un reto. En este trabajo se presenta un método que permite reducir la existencia de información innecesaria, aportando mayor eficiencia al IDS, sin afectar en gran medida la eficacia durante el proceso de detección. Los resultados alcanzados utilizando datos reales, en la detección de paquetes de tipo scan y backscatter, muestran que es factible el uso del método propuesto en escenarios reales.

Palabras claves: reducción de datos, scan, backscatter

Abstract
The exponential growth in the volume of information generated by the provision of telecommunications services makes the data processing an increasingly complex task. The presence of unnecessary information, such as redundant or noisy data, can affect the performance of Intrusion Detection Systems (IDS). Several data mining techniques have been proposed to reduce unnecessary information. However, carrying out the reduction process without affecting the efficacy of the detection process, remains a challenge. In this paper, a method to reduce the existence of unnecessary information is presented, improving the IDS efficiency, without greatly affecting the efficacy during the detection process. Achieved results using real data to detect scan and backscatter packages, show that the application of proposed method in real scenarios is feasible.

Keywords: *data reduction, scan, backscatter*

Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

Introducción

Hoy en día, una de las amenazas más importantes en Internet son los ataques de denegación de servicio distribuido (DDoS, por sus siglas en inglés). Estos ataques están orientados a interrumpir el acceso legítimo a un sistema de redes mediante la generación de un gran flujo de información desde varios puntos de conexión hacia un mismo punto de destino. Los reportes de inteligencia estadística del primer trimestre del presente año (2018) proporcionados por la compañía Kaspersky Lab, indican que se registraron ataques de DDoS contra 79 países y el ataque más largo duró 297 horas (más de doce días), siendo uno de los más largos en los últimos años (Alexander et al., 2018).

Teniendo en cuenta la observación anterior, se hace necesario garantizar la disponibilidad de los sistemas de redes ante posibles ataques de DDoS. Una característica presente en este tipo de ataque es que el atacante falsifica de forma aleatoria la dirección origen de los paquetes IP enviados a la víctima. Al no poder distinguir entre paquetes falsificados y legítimos, la víctima responde a los paquetes falsos como lo haría ante uno legítimo. Estos paquetes de respuesta son conocidos como retrodispersión (*backscatter*). Al enviarse los paquetes *backscatter* como respuesta a destinatarios aleatorios, pudieran emplearse telescopios de red (Moore et al., 2006) (network telescopes) para tener una evidencia indirecta de dichos ataques. El análisis de los paquetes *backscatter* pueden ayudar a identificar características particulares del ataque y de la víctima.

Las operaciones de reconocimiento de la red, más conocidas como *scan*, son en muchas ocasiones un paso previo a los ataques. Identificar qué están escaneando los atacantes puede alertar a los analistas de seguridad sobre cuáles servicios o tipos de computadora están siendo objeto de un ataque. Conocer esta información antes del ataque, le da la posibilidad al analista de tomar acciones preventivas para proteger los recursos. Por ejemplo, instalar parches, deshabilitar servicios en computadoras que no deberían estar ejecutándose, entre otros.

Los Sistemas de Detección de Intrusos (IDS) son utilizados para inspeccionar el tráfico de red en busca de algún evento asociado a la ejecución de una actividad maliciosa (Liao et al., 2013). El empleo de IDS basados en reglas es muy frecuente en las compañías proveedoras de servicios de telecomunicaciones. Esto se debe a que permiten procesar la información generada en tiempo real o muy cercano a este, lo cual hace posible prevenir o reducir los daños que se puedan ocasionar por la ejecución de actividades maliciosas.

Desde el punto de vista de la clasificación, el objetivo principal de construir un IDS basado en reglas es entrenar un clasificador basado en reglas que pueda categorizar los datos como ataques o normales (Herrera-Semenets et al., 2017). En los escenarios de detección de intrusos, la reducción del conjunto de entrenamiento T puede resultar muy útil para minimizar el consumo de recursos computacionales, como la memoria RAM, lo cual hace posible aplicar algoritmos con un elevado costo computacional. Además, la eliminación de información redundante y ruidosa hace más eficiente la etapa de entrenamiento, permitiendo obtener reglas representativas

de ataques en menor tiempo, lo cual puede ser de mucha utilidad en escenarios que procesen información en tiempo real o muy cercano a este.

No obstante, el proceso de reducción de datos puede conducir a que se pierda información durante la etapa de entrenamiento (Aggarwal, 2015). Esto conlleva a que la eficacia del clasificador pueda ser afectada durante la etapa de clasificación. El tiempo de ejecución de las estrategias de reducción reportadas en estos escenarios suele ser elevado, lo cual se debe a los grandes volúmenes de datos que se suelen procesar. Estos aspectos inciden directamente en el rendimiento de los clasificadores utilizados en tareas de detección de intrusos, haciendo que en algunos casos no sea factible su aplicación.

En este trabajo se presenta una estrategia de reducción de datos para mejorar el rendimiento de los clasificadores basados en reglas en la detección de paquetes de tipo *scan* y *backscatter*. La estrategia propuesta combina técnicas de selección de atributos con técnicas de selección de instancias para obtener un conjunto de entrenamiento reducido $S \subset T$, proporcionando mayor eficiencia a la etapa de entrenamiento, sin afectar en gran medida la eficacia durante la etapa de clasificación.

Trabajos relacionados

Si representamos el conjunto de datos de entrenamiento T como una matriz, podemos decir que la reducción de datos puede ser en términos de reducción de filas (instancias) o reducción de columnas (atributos) (Aggarwal, 2015). Existen tres enfoques ampliamente utilizados en tareas de reducción de datos en estos escenarios, ellos son: (1) selección de atributos (Ganapathi and Duraivelu, 2015; Vinutha and Poornima, 2018), (2) selección de instancias (Guo et al., 2013; Ashfaq et al., 2017) e híbridos, donde la selección de atributos y la selección de instancias se combinan (Chen et al., 2014).

Los métodos basados en selección de atributos buscan los atributos más representativos del conjunto de datos. De esta manera solo se utiliza un subconjunto de atributos de los datos subyacentes para generar el modelo de clasificación. Esto facilita la comprensión de los patrones extraídos, representados como reglas, e incrementa el rendimiento de la etapa de entrenamiento.

La mayoría de los métodos de selección de atributos propuestos para estos escenarios solo utilizan una medida para estimar el nivel de representatividad que puede tener un atributo sobre los datos. Esto conlleva a que los resultados obtenidos estén sesgados por la medida utilizada. Cada medida analiza distinta información en los atributos, por lo cual se puede obtener un subconjunto de atributos diferente para cada una, con igual nivel de representatividad. En nuestra opinión, la combinación de varias medidas puede conducir a una mejor selección del conjunto final de atributos.

El objetivo de los métodos basados en selección de instancias es obtener un subconjunto de entrenamiento reducido S a partir de T, de forma tal que S no contenga instancias innecesarias. Estos métodos pueden categorizarse como incrementales si inician con $S = \emptyset$, o como decrementales si se inicializa $S = T$ (Olvera-López et al., 2010). Según la estrategia utilizada para seleccionar las instancias, estos métodos se pueden agrupar en *Filter* o *Wrapper* (Olvera-López et al., 2010). En los métodos *Wrapper* se utiliza el clasificador en el proceso de selección, donde aquellas instancias que no afecten la eficacia del clasificador se eliminan. Por otra parte, los métodos *Filter* son independientes del clasificador utilizado y el criterio de selección se basa en distintas heurísticas.

El hecho de que un método *Wrapper* esté orientado a un clasificador específico, hace que la eficacia del clasificador para el cual fue concebido pueda ser superior a la alcanzada aplicando un método *Filter*. No obstante, cuando se desea aplicar en varios clasificadores, la eficacia que se alcanzada utilizando un método *Filter* es superior, ya que su criterio de selección es independiente al clasificador utilizado. Los métodos *Wrapper* suelen ser menos eficientes que los *Filter* (Olvera-López et al., 2010), lo cual se debe a la ejecución de análisis complejos sobre los datos utilizando un clasificador como parte de su estrategia de selección. La complejidad suele ser directamente proporcional al volumen de datos que se desean procesar, lo cual conlleva a que en ocasiones su aplicación en los escenarios que se abordan en este trabajo no sea factible, haciendo que los métodos *Filter* sean una opción más viable.

Teniendo en cuenta las características propias de estos escenarios, resulta complejo definir si es más ventajoso aplicar selección de atributos o selección de instancias como métodos de reducción. Si solo se aplica selección de atributos, puede permanecer información innecesaria en las instancias y si solo se aplica selección de instancias pues pudieran quedar atributos no representativos del conjunto de datos. Una solución a esta disyuntiva es la combinación de ambos enfoques dando lugar a uno híbrido.

En la propuesta de Chen *et al.* (Chen et al., 2014) se presenta un método híbrido. La selección de atributos se realiza aplicando el método OneR (Holte, 1993). Luego, se realiza la selección de instancias utilizando el método de Affinity Propagation (Frey and Dueck, 2007) (AP) basado en agrupamiento. El problema aquí radica en que el método AP tiene un elevado costo computacional y agota la memoria RAM disponible (4 Gb) cuando $\mid T \mid > 6000$. Para hacer factible su aplicación en este escenario, los autores proponen una solución distribuida para AP utilizando MapReduce (Dean and Ghemawat, 2008). Para un rendimiento adecuado, los autores recomiendan el empleo de 8 nodos cada uno equipado con un procesador quad-core 2.5 GHz y 4 Gb de memoria RAM para reducir un conjunto de datos de entrenamiento $\mid T \mid = 12872$ instancias, lo cual es considerablemente pequeño comparado con un escenario real. Esto hace que su propuesta sea costosa en términos computacionales. Los resultados son evaluados sobre un conjunto de prueba conformado por 115848, lo cual también representa un conjunto de datos pequeño para estos escenarios. Además, no es posible observar

su escalabilidad sobre un conjunto de datos más grande, característico de escenarios de detección de intrusos.

El principal problema está en el elevado tiempo de ejecución que requieren los métodos de selección de instancias para procesar un volumen grande de datos, lo cual afecta directamente el enfoque híbrido. Teniendo en cuenta esto, se propone un método híbrido que incluye un paso de selección de atributos donde se combinan tres medidas diferentes para obtener un conjunto de atributos final. El segundo paso consiste en un método *Filter* de selección de instancias que incorpora un proceso rápido de reducción de instancias, a partir de un re-etiquetado previo. De esta forma se reduce el conjunto de entrenamiento con un bajo tiempo de ejecución, sin afectar en gran medida la eficacia durante la clasificación.

Propuesta

La mayoría de los métodos que realizan selección de atributos y han reportados en estos escenarios utilizan solo una medida para seleccionar un subconjunto de atributos final con el cual representar los datos. De esta forma no se aprovechan las ventajas que puede ofrecer la combinación de varias medidas, ya que cada medida puede estimar información diferente en los atributos. Luego de un estudio, se determinó que las medidas más utilizadas pueden ser agrupadas en tres categorías: basadas en entropía (Information Gain, Gain Ratio y Symmetric Uncertainty), basadas en estadística (Chi-square) y basadas en instancias (Relief y ReliefF) (Liu et al., 2016).

Como se muestra en la Figura 1, en el método híbrido de reducción de datos (HRD) propuesto en este trabajo se utilizaron tres medidas diferentes, una representativa de cada categoría, dichas medidas son: Information Gain, Chi-square y ReliefF. La medida ReliefF (RF) estima que tan bien un atributo puede diferenciar instancias de clases diferentes, basándose en los vecinos más cercanos de su misma clase y clases diferentes. Chi-square (CHI) es una medida estadística no paramétrica que estima la correlación entre la distribución de un atributo y la distribución de la clase. La medida Information Gain (IG) calcula la cantidad de información que un atributo puede proporcionar sobre si una instancia pertenece a una clase u otra.

Cada medida M luego de procesar el conjunto T, obtiene un conjunto de puntuaciones P_M, donde un elemento $p_a \in P_M$ representa la puntuación asignada a un atributo a, mientras mayor sea la puntuación, más representativo será el atributo de T. Luego de obtenerse P_M, se calcula la media de sus puntuaciones $\bar{p}_M$. Si un atributo a satisface la condición $p_a > \bar{p}_m$, se adiciona a un conjunto de atributos A_M. Para ganar en eficiencia, el método se programó para que cada medida se ejecute de forma paralela. Después de obtenerse los conjuntos de atributos para cada medida, se selecciona $A = A_{RF} \bigcup A_{CHI} \bigcup A_{IG}$ como conjunto más representativo.

Con el conjunto de datos representado solamente por los atributos seleccionados en A se procede a reducir el

Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

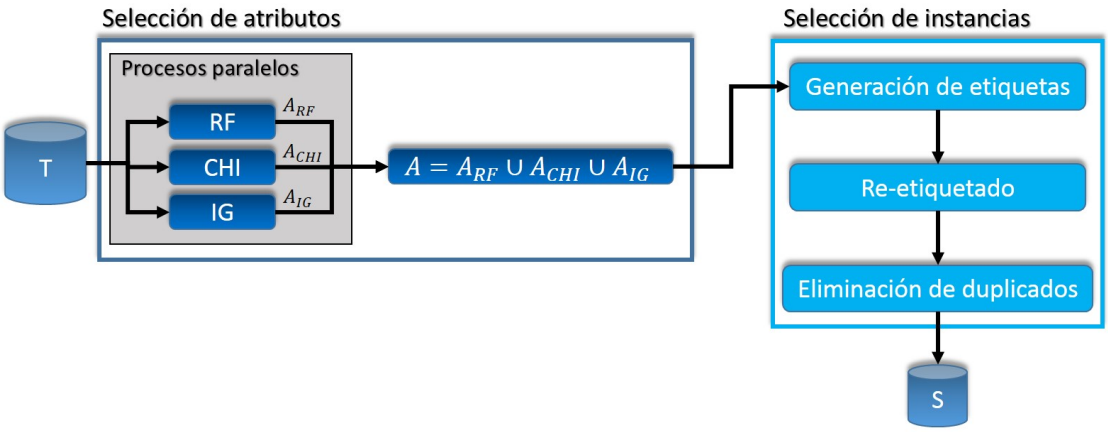

Figura 1: Esquema del método HRD.

número de instancias (ver Figura 1). Durante la etapa de generación de etiquetas se discretizan los atributos continuos. Este proceso posibilita el mapeo de un amplio rango de valores numéricos a un pequeño subconjunto de valores discretos. En varios estudios comparativos k-means ha sido utilizado como método de discretización (Maslove et al., 2012; Dash et al., 2011). En estos estudios se concluye que el proceso de discretización utilizando k-means, además de ser muy eficiente, produce resultados más consistentes y favorables que con otros métodos. El objetivo de utilizar un algoritmo de agrupamiento aquí es buscar los valores similares y agruparlos de forma tal que la distancia entre los valores sea la menor posible, mientras que la distancia entre los grupos sea la mayor posible. Teniendo en cuenta este análisis se utilizó k-means para generar las etiquetas.

El algoritmo de k-means se ejecuta sobre el conjunto de valores que toma cada atributo numérico a_i seleccionado. Cada grupo obtenido está conformado por un rango de atributos numéricos, los cuales son representados por una única etiqueta numérica. El uso de estos grupos permite cubrir valores de atributos que no estuvieron en el conjunto T, pero que pudieran estar en el conjunto de prueba.

El proceso de re-etiquetado consiste en reemplazar los valores de los atributos numéricos por sus etiquetas correspondientes. Finalmente se ejecuta la eliminación de duplicados, que como su nombre indica consiste en eliminar las instancias duplicadas. Es valido resaltar que una instancia es duplicada si al menos existe otra con los mismos valores de atributos y clase. El resultado es un conjunto de entrenamiento reducido S, que es utilizado para generar reglas que sean evaluadas sobre nuevos datos.

Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

Resultados y discusión

En esta sección se muestran los resultados alcanzados utilizando HRD para reducir el conjunto de entrenamiento original y mejorar el rendimiento de tres clasificadores basados en regla en la detección de paquetes de tipo *scan* y *backscatter*. Se utilizó una PC equipada con un procesador quad-core a 3.5 GHz, 8 Gb de memoria RAM y sistema operativo Ubuntu 16.04. Los datos utilizados en este experimento fueron recolectados por un telescopio de red de la Universidad Tecnológica de Delft (TUDelft). Específicamente el conjunto de datos utilizado representa 24 horas de tráfico *scan* y *backscatter*.

Cada paquete de red se procesó y se representó como una instancia, donde cada elemento representa un atributo del paquete. Los atributos utilizados para representar los datos fueron: *source address, source port, destination address, destination port, source MAC, destination MAC, protocol, packet length, IP length, TCP flag, ICMP message, TTL* y *ToS*.

Los paquetes que llegan a un telescopio de red son enviados a direcciones IP no utilizadas, por tanto pudieran provenir de una mala configuración, un intento de *scan* o *backscatter*. Para poder etiquetar correctamente el conjunto de entrenamiento, se siguió la propuesta presentada por Blenn *et al.* (Blenn et al., 2017). En esta se propone una estrategia para identificar en los datos recolectados por un telescopio de red, cuáles paquetes pertenecen a tráfico de tipo *scan* o *backscatter*. Básicamente, la idea se centra en el atributo *TCP flag*, que en caso de contener el valor SYN+ACK (para un puerto abierto) o RST (para un puerto cerrado) el paquete será de tipo *backscatter*, mientras que si el valor es SYN el paquete será de tipo *scan*.

Luego de etiquetar los datos retiramos el atributo TCP flag para identificar otros patrones que también puedan discriminar entre tráfico de tipo *scan* o *backscatter*. El conjunto de datos etiquetado se quedó conformado por 21470669 instancias y 12 atributos. Este conjunto de datos se fragmentó por horas, con el objetivo de entrenar con la 1ra hora y evaluar las reglas obtenidas sobre las restantes. En este sentido, el conjunto de entrenamiento T estaba formado por 794544 instancias y 12 atributos. Luego de aplicar HRD, se obtuvo un conjunto de entrenamiento reducido S con 62768 instancias y 8 atributos (*source address, source port, destination address, destination port, packet length, IP length, ICMP message and TTL*). Esto significa que se redujo en un 92 % el número de instancias en T, así como en un 33 % la cantidad de atributos.

Para evaluar el desempeño de HRD se utilizaron tres clasificadores: Non-Nested generalized exemplars (Sylvain, 2002) (NNge), Decision Table/Naive Bayes (Hall and Frank, 2008) (DTNB) y PART (Frank and Witten, 1998). Estos clasificadores han sido ampliamente utilizados en tares de detección de intrusos (Panda and Patra, 2009; MeeraGandhi et al., 2010; Azad and Jha, 2014). Los modelos de clasificación se construyeron para cada uno de los clasificadores utilizando el conjunto de entrenamiento S y se compararon los resultados con respecto a los alcanzados utilizando T para el entrenamiento. Para medir la eficacia durante el proceso

Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

de clasificación utilizamos tres medidas de calidad: *accuracy* (Acc)(ver Ecuación 1), porcentaje de verdaderos positivos (TPR)(ver Ecuación 2) y porcentaje de falsos positivos (FPR)(ver Ecuación 3).

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100 \tag{1}$$

$$TPR = \frac{TP}{TP + FN} \cdot 100 \tag{2}$$

$$FPR = \frac{FP}{FP + TN} \cdot 100 \tag{3}$$

Las variables TP, TN, FP, FN representan la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos respectivamente.

Los resultados que se muestran en la Tabla 1 se obtuvieron empleando la 1ra hora como conjunto de prueba. Se puede apreciar como los clasificadores NNge y PART obtienen resultados muy similares para ambos conjuntos de entrenamiento, S y T. Por otra parte, el clasificador DTNB alcanza los mismos valores de TPR y FPR, con una minima diferencia de 0,02 % en la medida Acc. Luego de esta evaluación, es posible decir que no existe una diferencia considerable entre los resultados obtenidos con el conjunto de entrenamiento T y el conjunto S, lo cual indica que no se afecta en gran medida la eficacia del proceso de detección (clasificación).

Tabla 1: Resultados alcanzados utilizando S y T como conjuntos de entrenamiento.

Classifier	TPR	FPR	Acc.	Clase	Conjunto
DTNB	1,000	0,020	98,41	backscatter	S
	0,980	0,000	98,41	scan	S
	1,000	0,020	98,43	backscatter	T
	0,980	0,000	98,43	scan	T
NNge	0.992	0,015	98,47	backscatter	S
	0,985	0,008	98,47	scan	S
	0,995	0,015	98,58	backscatter	T
	0,985	0,005	98,58	scan	T
PART	0.995	0,018	98,44	backscatter	S
	0,982	0,005	98,44	scan	S
	0,995	0,015	98,58	backscatter	T
	0,985	0,005	98,58	scan	T

El uso del conjunto reducido S aporta ciertas ventajas al clasificador. En la Figura 2 se muestran los tiempos empleados por cada clasificador, durante el entrenamiento, para construir el modelo de clasificación. En los tres clasificadores el tiempo se reduce significativamente, pero en el caso específico del clasificador PART el tiempo se reduce en aproximadamente un 90 %.

Revista Cubana de Ciencias Informáticas
Vol. 12, No. Especial UCIENCIA, Septiembre 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 1-12
http://rcci.uci.cu

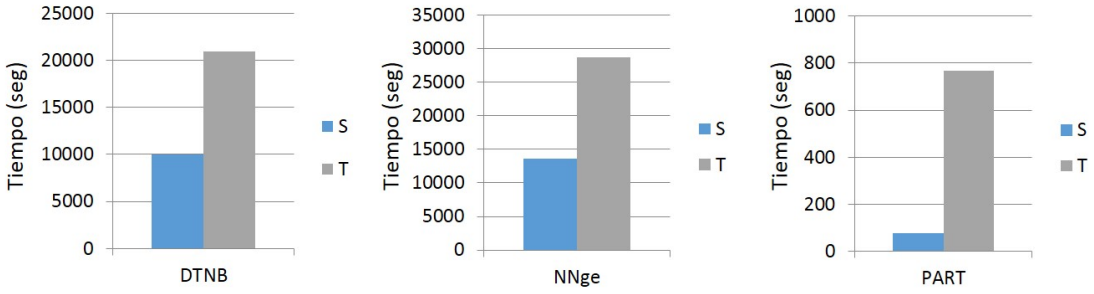

Figura 2: Tiempo empleado para construir los modelos de clasificación.

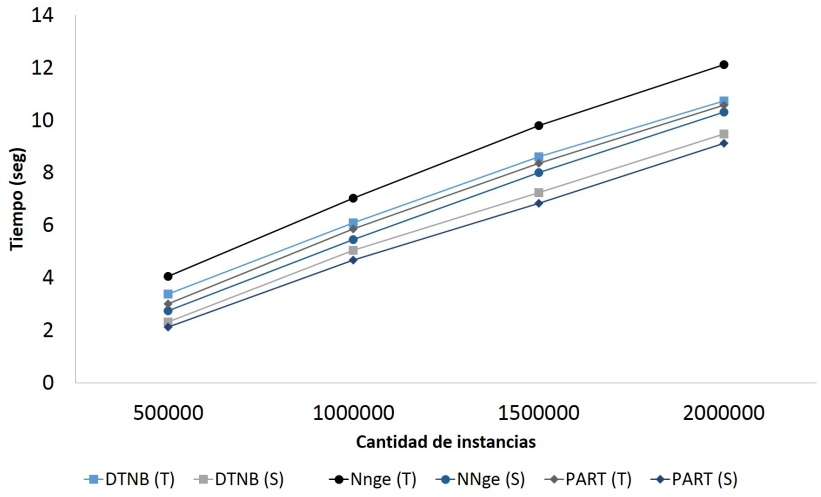

Figura 3: Tiempo empleado durante el proceso de clasificación.

Otra contribución de HRD consiste en que mejora el tiempo empleado durante el proceso de clasificación. En la Figura 3 se puede ver como mejoran los clasificadores en cuanto a tiempo empleado durante la clasificación cuando se entrena con el conjunto reducido S. Esto puede ser muy útil en escenarios que requieran procesar los datos en tiempo real.

Conclusiones

En este trabajo se presentó un método para mejorar el rendimiento de los IDS basados en reglas. La propuesta fue evaluada sobre un conjunto de datos reales asociados a tráfico de tipo *scan* y *backscatter*. El uso de HRD permitió reducir en gran medida el tiempo empleado durante la construcción de los modelos de clasificación, sin

afectar en gran medida la eficacia durante el proceso de clasificación. Además, su aplicación logró reducir los tiempos utilizados por los clasificadores para la clasificación, lo cual es un aspecto importante para el análisis de datos en tiempo real.

Teniendo en cuenta las propiedades estándar de la PC donde se ejecutó la propuesta, se hace notable que no requiere de grandes recursos computacionales para procesar grandes volúmenes de datos. Esta característica hace factible su uso como etapa de preprocesamiento en escenarios que no cuentan con muchos recursos y necesitan ejecutar algoritmos costosos para extraer conocimiento.

HRD no está limitado a la detección de tráfico de tipo *scan* y *backscatter*, sino que también puede ser empleado en escenarios con características similares. Un ejemplo pudieran ser la detección de fraudes en servicios de telecomunicaciones, detección de fraudes en transacciones bancarias, entre otras.

Como trabajo futuro nos proponemos integrar la propuesta presentada en un IDS basado en reglas y evaluar los resultados obtenidos en un escenario real.

Referencias

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.

Alexander, K., Oleg, K., and Ekaterina, B. (2018). Ddos attacks in q1 2018. *[citado 16 de mayo de 2018]. Disponible en Internet: https://securelist.com/ddos-report-in-q1-2018/85373/*.

Ashfaq, R. A. R., He, Y.-l., and Chen, D.-g. (2017). Toward an efficient fuzziness based instance selection methodology for intrusion detection system. *International Journal of Machine Learning and Cybernetics*, 8(6):1767–1776.

Azad, C. and Jha, V. K. (2014). Data mining based hybrid intrusion detection system. *Indian Journal of Science and Technology*, 7(6):781–789.

Blenn, N., Ghiëtte, V., and Doerr, C. (2017). Quantifying the spectrum of denial-of-service attacks through internet backscatter. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, page 21. ACM.

Chen, T., Zhang, X., Jin, S., and Kim, O. (2014). Efficient classification using parallel and scalable compressed model and its application on intrusion detection. *Expert Systems with Applications*, 41(13):5972–5983.

Dash, R., Paramguru, R. L., and Dash, R. (2011). Comparative analysis of supervised and unsupervised discretization techniques. *International Journal of Advances in Science and Technology*, 2(3):29–37.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.

Ganapathi, N. P. and Duraivelu, V. (2015). A knowledgeable feature selection based on set theory for web intrusion detection system. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, pages 51–59. Springer.

Guo, C., Zhou, Y.-J., Ping, Y., Luo, S.-S., Lai, Y.-P., and Zhang, Z.-K. (2013). Efficient intrusion detection using representative instances. *computers & security*, 39:255–267.

Hall, M. A. and Frank, E. (2008). Combining naive bayes and decision tables. In *FLAIRS Conference*, volume 2118, pages 318–319.

Herrera-Semenets, V., Pérez-García, O. A., Gago-Alonso, A., and Hernández-León, R. (2017). Classification rule-based models for malicious activity detection. *Intelligent Data Analysis*, 21(5):1141–1154.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90.

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., and Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24.

Liu, W., Liu, S., Gu, Q., Chen, J., Chen, X., and Chen, D. (2016). Empirical studies of a two-stage data preprocessing approach for software fault prediction. *IEEE Transactions on Reliability*, 65(1):38–53.

Maslove, D. M., Podchiyska, T., and Lowe, H. J. (2012). Discretization of continuous features in clinical datasets. *Journal of the American Medical Informatics Association*, 20(3):544–553.

MeeraGandhi, G., Appavoo, K., and Srivasta, S. (2010). Effective network intrusion detection using classifiers decision trees and decision rules. *Int. J. Advanced network and application, Vol2.*

Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., and Savage, S. (2006). Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2):115–139.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., and Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review*, 34(2):133–143.

Panda, M. and Patra, M. R. (2009). Ensembling rule based classifiers for detecting network intrusions. In *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*, pages 19–22. IEEE.

Sylvain, R. (2002). Nearest neighbor with generalization. *University of Canterbury, Christchurch, New Zealand.*

Vinutha, H. and Poornima, B. (2018). An ensemble classifier approach on different feature selection methods for intrusion detection. In *Information Systems Design and Intelligent Applications*, pages 442–451. Springer.