45

Fecha de presentación: octubre, 2021
Fecha de aceptación: diciembre, 2021
Fecha de publicación: febrero, 2022

ESTUDIO DE TÉCNICAS

DE MINERÍA DE DATOS PARA LA DETECCIÓN DE ATAQUES EN EL CONJUNTO DE DATOS NSL-KDD

STUDY OF DATA MINING TECHNIQUES FOR THE DETECTION OF ATTACKS IN THE NSL-KDD DATA SET

Amilkar Yudier Puris Cáceres[1]
E-mail: apuris@uteq.edu.ec
ORCID: https://orcid.org/0000-0002-7288-7451
Andrés Florencia Toala[1]
E-mail: andres.florencia2013@uteq.edu.ec
ORCID: https://orcid.org/0000-0002-7920-8095
Raúl Hernández Palacios[2]
E-mail: rapalacios81@hotmail.es
ORCID: https://orcid.org/0000-0002-1131-4545
Emilio Zhuma Mera[1]
E-mail: ezhuma@uteq.edu.ec
ORCID: https://orcid.org/0000-0002-3086-1413
Ángel Torres Quijije[1]
E-mail: ezhuma@uteq.edu.ec
ORCID: https://orcid.org/0000-0002-7037-7191
Byron Oviedo Bayas[1]
E-mail: boviedo41@uteq.edu.ec
ORCID: https://orcid.org/0000-0002-5366-5917
[1] Universidad Técnica Estatal de Quevedo. Ecuador.
[2] Universidad Autónoma del Estado de Hidalgo. México.

Cita sugerida (APA, séptima edición)

Puris Cáceres, A. Y., Florencia Toala, A., Hernández Palacios, R., Zhuma Mera, E., Torres Quijije, Á., & Oviedo Bayas, B. (2022). Estudio de técnicas de minería de datos para la detección de ataques en el conjunto de datos NSL-KDD. *Revista Universidad y Sociedad*, 14(S1), 428-437.

RESUMEN

La presente investigación inicia con un estudio referencial de trabajos similares basada en escenarios para la detección de intrusos en la red aplicando técnicas de minería de datos en un conjunto de datos con atributos referentes a conexiones de una red. En particular se ha tomado el conjunto de datos NSL-KDD. Luego se pretende replicar resultados de investigaciones previas donde aplicando algoritmos clasificadores determina si una conexión es de tipo normal o un ataque a la red. Posterior a esto se complementa con la aplicación de nuevos algoritmos de clasificación para obtener mejores resultados, al igual de nuevos algoritmos selectores de atributos con el fin de reducir o cambiar ciertos atributos para obtener resultados similares. Finalmente se propone una selección de atributos en base a la frecuencia de aparición en subconjuntos previos. Para la comparación de los resultados se han tomado los porcentajes de aciertos y tiempo de construcción de los modelos de cada algoritmo aplicado.

Palabras clave: Minería de datos, detección de intrusos, NSL-KDD dataset.

ABSTRACT

The present investigation begins with a referential study of similar works based on scenarios for the detection of intruders in the network applying data mining techniques in a data set with attributes referring to network connections. In particular, the NSL-KDD data set has been taken. Then it is intended to replicate results of previous investigations where applying classifying algorithms determines if a connection is normal or a network attack. After this, it is complemented with the application of new classification algorithms to obtain better results, as well as new attribute selector algorithms in order to reduce or change certain attributes to obtain similar results. Finally, a selection of attributes based on the frequency of appearance in previous subsets is proposed. For the comparison of the results, the percentages of successes and construction time of the models of each applied algorithm have been taken.

Keywords: Data mining, intrusion detection, NSL-KDD dataset.

INTRODUCCIÓN

Desde su invención hasta nuestros días, el número de computadoras ha crecido consolidándose como un instrumento casi imprescindible en la vida cotidiana del hombre. Con la posibilidad de conectar múltiples computadores formando redes, surgieron nuevos retos y aplicaciones.

Entre los principales retos se encuentra la seguridad en redes informáticas, la cual es de vital importancia debido al gran volumen de datos que se manejan. La información se ha convertido en el activo más importante para las empresas, por ello éstas necesitan de herramientas que supervisen las actividades dentro de la red y alerten a los administradores de situaciones sospechosas que comprometan la integridad, confidencialidad y disponibilidad de los datos.

Una de las herramientas de seguridad de las que disponen las grandes empresas son los llamados Sistemas de Detección de Intrusos o IDS por sus siglas en inglés (Lazarevic, et al., 2003). Entre los tipos de IDS se encuentran los basados en uso indebido, un ejemplo de ello son los antivirus, los cuales necesitan de actualizaciones periódicas con una base de datos con patrones de ataques definidos. El otro tipo basado en anomalías, los cuales construyen un modelo del sistema basado en técnicas de Inteligencia Artificial para la detección de un ataque. Los IDS basados en anomalías no están tan desarrollados por los fabricantes debido a su baja fiabilidad frente a los sistemas de detección basados en el uso indebido, a pesar que son más potentes han sido más utilizados en el ámbito investigativo (Tribak, 2012).

Revathi & Malathi (2013), realizaron un análisis detallado del conjunto de datos NSL-KDD utilizando varias técnicas de sistemas de aprendizaje automático para la detección de intrusiones

Existen varios estudios previos realizados con este enfoque donde se utiliza el conjunto de datos NSL-KDD (Tavallaee, et al., 2009), el cual contiene patrones de firmas de conexiones a una red que determinan si ésta es una conexión normal o un ataque. En estos estudios se aplican técnicas de minería de datos logrando excelentes resultados. Tal es el caso de Dhanabal & Shantharajah (2015), que analizan la efectividad de los diversos algoritmos de clasificación en la detección de anomalías en los patrones de tráfico de la red, donde los algoritmos J48, SVM, Naïve Bayes ofrecen niveles de aciertos muy buenos. En el estudio presentado por Noureldien & Yousif (2016), cuyo objetivo es el de mejorar la precisión de detección de un conjunto de algoritmos de aprendizaje automático seleccionados para detectar diferentes tipos de clases de ataque DoS (Ugochukwu, 2019). Los algoritmos pertenecen a diferentes técnicas supervisadas: PART, BayesNet, IBK, Logistic, J48, Random Committee y InputMapped. En la investigación de Ayuso & Barcenilla (2008), se analiza la base de conocimiento NSL-KDD aplicando algoritmos clasificadores: ***ZeroR, OneR, PART, BFTree, J48, IBK, MultiLayerPerceptron, NaiveBayes***. Donde se determinó que los algoritmos que mejores resultados presentaron fueron PART y J48 debido a su grado de acierto y bajo coste computacional.

Sin embargo, de los estudios revisados el más completo fue la investigación de Tribak (2012), el cual aplica un estudio estadístico de las diferentes técnicas de clasificación basadas en Inteligencia Artificial aplicadas a la detección de intrusos, bajo distintas perspectivas tales como la discretización de datos y la selección de atributos. De la variedad de algoritmos utilizados en esta investigación, tales como: ***Naïve Bayes, PerceptronMulticapa, C4.5, SMO, KNN-1, RandomForest, TAN***; de los cuales RandomForest y KNN-1 fueron los mejores con porcentajes cercanos al 100% en su grado de acierto y bajo coste computacional.

MATERIALES Y MÉTODOS

En la Figura 1, se muestran los procesos o etapas implicadas en el desarrollo de la presente investigación, los mismos que se describen a continuación:

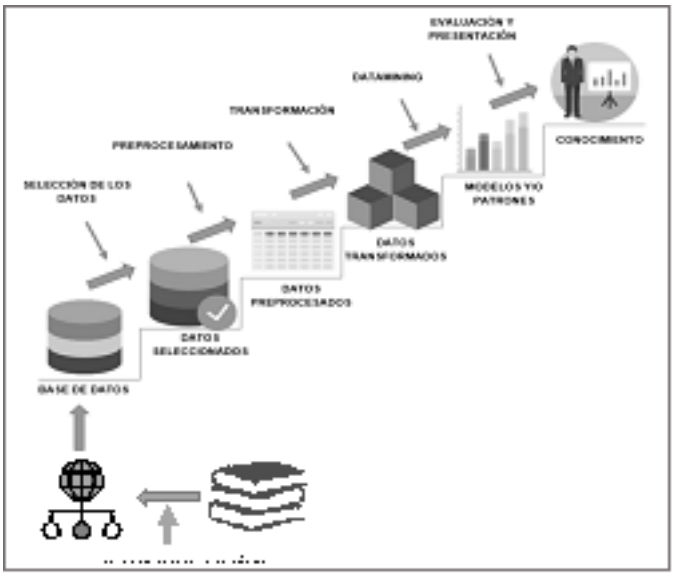

Figura 1. Etapas de desarrollo de la investigación.

a) Revisión bibliográfica

En esta fase inicial se realizó una revisión bibliográfica acerca de estudios similares, donde se encontró una gran

variedad de ellos en los cuales aplicaban distintos selectores de atributos, algoritmos clasificadores con buenos resultados.

b) Selección del conjunto de datos

Se optó por seleccionar el conjunto de datos NSL-KDD, el cual es muy completo y consta de 42 atributos que caracterizan una conexión a una red, con una clase que determina si ésta es de tipo ataque o una conexión normal. Este conjunto de datos se ha utilizado en estudios de las investigaciones revisadas en el paso anterior.

c) Replicación de resultados

Para una mayor comprensión de los resultados obtenidos en estudios anteriores, se derivó a replicar los mismos, este proceso se detalla en el siguiente apartado.

d) Preprocesamiento, Transformación y Minería de Datos

Esta serie de etapas están enlazadas debido a que el proceso es retroalimentado con la aplicación de nuevos selectores de atributos y aplicación de distintos algoritmos clasificadores:

- Para replicar los resultados se aplicó dos métodos de discretización: Fayyad & Irani, e Intervalos de Igual Frecuencia, obteniendo dos subconjuntos, luego se aplicó los cuatro métodos selectores de atributos a ambos subconjuntos anteriores: CFS (Filtro Basado en Correlación), CNS (Filtro Basado en Consistencia), C4.5, Naïve Bayes, obteniendo ahora ocho subconjuntos, para después aplicar sólo los algoritmos clasificadores que mejores resultados devolvieron en la investigación: Furia, C4.5, KNN-1, Random Forest, SMO Polinúcleo, TAN.
- Como forma de intentar encontrar diferencia en los resultados se aplicó un algoritmo para balaceo de clases: SMOTE, y luego nuevamente los algoritmos discretizadores, selectores de atributos y los algoritmos clasificadores, pero al solo tener una diferencia de 12% entre cantidades de registros para cada clase, la clase minoritaria fue elevada casi al doble convirtiéndose en clase mayoritaria y desbalanceando aún más, donde los resultados no fueron los esperados, por lo tanto se descartó el balanceo de clases, solamente se conservaron los ocho subconjuntos de datos obtenidos al aplicar los selectores de atributos para una posterior utilización.
- Después de esto se seleccionaron tres nuevos selectores de atributos: EvolutionarySearch, GeneticSearch, MultiObjetiveSearch, obteniendo así tres nuevos subconjuntos de datos, tanto para el subconjunto discretizado por Fayyad & Irani e Intervalos de Igual Frecuencia. Después se aplicaron los seis algoritmos clasificadores utilizados anteriormente: Furia, C4.5, KNN-1, Random Forest, SMO Polinúcleo, TAN. De esta manera se intentó mejorar los resultados, pero no fue así, de hecho comparando con los resultados anteriores, éstos disminuyeron su grado de acierto y aumentaron el tiempo de construcción del modelo.
- En la literatura revisada se aplican otros algoritmos clasificadores pero que no devolvían buenos resultados con los subconjuntos de datos obtenidos con los selectores de atributos: CFS, CNS, C4.5, NB. Por lo tanto, se optó por aplicar seis algoritmos también propuestos en la literatura, de tal manera que con los nuevos subconjuntos de datos se obtengan nuevos resultados, que quizá mejoren, los nuevos algoritmos aplicados fueron: PART, KNN-50, RBF-Network (Redes de Función de Base Radial), MOE-Fuzzy (clasificador basado en reglas difusas usando el algoritmo evolutivo multiobjetivo ENORA), NNge (Vecino más Cercano con Generalización).

e) Selección de atributos en base a frecuencia

Después de analizar los resultados devueltos con cada aplicación de algoritmos discretizadores, selectores de atributos y algoritmos clasificadores se propone la selección de los atributos en base (Tabla 1) al porcentaje de frecuencia de cada atributo en los 11 dataset encontrados previamente para cada conjunto discretizado, tomando como mínima frecuencia de 40%, donde se obtuvo dos conjuntos de datos respectivamente con 8 atributos para el conjunto discretizado por Fayyad & Irani, y 7 atributos para el conjunto de Intervalos de Igual Frecuencia. De esta manera se aplicaron los siguientes algoritmos clasificadores: KNN-1, KNN-3, KNN-5 y KNN-7, Random Forest, C4.5, PART. Obteniendo resultados muy cercanos a los propuestos en la literatura con un dataset más reducido y con ciertos atributos distintos.

Tabla 1. Representación de selección de atributos en base a frecuencia.

SELECTOR / ATRIBUTO	Selector1	Selector2	SelectorN	Frecuencia (%)
Atributo1	X	x	x	100%
Atributo2		x	x	66%
Atributo3	X		x	66%
AtributoN		x	x	n%

f) Evaluación y Comparación de resultados

Finalmente se compararon todos los resultados con el fin de encontrar selectores de atributos y algoritmos clasificadores que mejoran el resultado de estudio.

Conjunto de datos NSL-KDD:

Los datos utilizados en este estudio son una pequeña selección del conjunto de datos del concurso KDD 1999, en donde se usó una versión reducida de la amplia variedad de intrusiones militares simuladas en un entorno de red, proporcionadas por DARPA Intrusion Detection Program Evaluation en 1998, que tenían como objetivo evaluar el estudio y la investigación en la detección de intrusiones (Tavallaee, et al., 2009). NSL-KDD es una base de datos con miles de patrones de firmas de ataques así como de conexiones normales, y a la vez es una mejora de los datos del concurso KDD cup"99". En este conjunto de datos cada registro de conexión está compuesto de 42 atributos, lo que supone unos 100 bytes por registro.

La Tabla 2 muestra los dataset que se encuentran al descargar el paquete NSL-KDD:

Tabla 2. Datasets del paquete NSL-KDD.

Dataset	**N° registros**
KDDTrain+.arff	125973
KDDTrain+_20Percent	**25192**
KDDTest+.arff	22544
KDDTest-21.arff	11850

De los dataset mostrados en la Tabla 2 se optó por el de 20% correspondiente al dataset completo. En las Tablas 3, 4, 5 se describen los atributos del conjunto de datos.

Tabla 3. Atributos básicos de las conexiones TCP.

ATRIBUTO	DESCRIPCION	TIPO
duration	Tiempo en segundos de la conexión	Continuo
protocol_type	Tipo de protocolo (TCP, UDP)	Discreto
service	Tipo de servicio destino (HTTP, Telnet)	Discreto
src_byte	Número de bytes del origen al destino	Discreto
dst_byte	Número de bytes del destino al origen	Discreto
flag	Estado de la conexión	Categórico
land	1 si la conexión corresponde miss/host; 0 de otro modo	Categórico
wrong_fragment	Número de fragmentos "erróneos"	Discreto
urgent	Número de paquetes	Discreto

Tabla 4. Atributos derivados de una conexión TCP.

ATRIBUTO	DESCRIPCION	TIPO
hot	Número de indicadores "importantes	Continuo
num_failed_logins	Número de intentos de acceso fallido	Continuo
logged_in	1 acceso exitoso; 0 fallo	Discreto
num_compromissed	Número de condiciones sospechosas	Continuo
root_shell	1 si es superusuario; 0 en otro caso	Discreto
su_attempted	1 si se intenta comando "su root"; 0	Discreto
num_root	Número de accesos como root	Continuo
num_file_creations	Número de operaciones de creación de archivos	Continuo
num_shells	Números de Shells prompts abiertos	Continuo
num_access_files	Número de comandos externos (sesión FTP)	Continuo

is_hot_login	1 si login pertenece a la lista “hot”; 0 caso contrario	Discreto
is_guest_login	1 si login es del tipo “guest”, 0 caso contrario	Discreto

Tabla 5. Atributos con ventana de 2 segundos.

ATRIBUTO	DESCRIPCION	TIPO
count	Número de conexiones a la misma máquina que la conexión actual en los últimos dos segundos	Continuo
serror_rate	% de conexiones con error “SYN”	Continuo
rerror_rate	% de conexiones con error “REJ”	Continuo
same_srv_rate	% de conexiones al mismo servicio	Continuo
diff_srv_rate	% de conexiones a diferentes servicios	Continuo
srv_count	Número de conexiones al mismo servicio que la conexión actual en los últimos dos segundos	Contínuo
srv_serror_rate	% de conexiones con error “SYN”	Continuo
srv_rerror_rate	% de conexiones con error “REJ”	Continuo
srv_diff_host_rate	% de conexiones a diferentes hosts	Continuo
dst_host_count	Número de conexiones que tienen la misma dirección IP de host de destino	Continuo
dst_host_srv_count	Número de conexiones que tienen el mismo número de puerto	Continuo
dst_host_same_srv_count	El porcentaje de conexiones que fueron para el mismo servicio, entre las conexiones agregadas en ***dst_host_count***	Continuo
dst_host_same_srv_rate	El porcentaje de conexiones que fueron para el mismo servicio, entre las conexiones agregadas en ***dst_host_count***	Continuo
dst_host_diff_srv_rate	El porcentaje de conexiones que fueron a diferentes servicios, entre las conexiones agregadas en ***dst_host_count***	Continuo
dst_host_same_src_port_rate	El porcentaje de conexiones que estaban en el mismo puerto de origen, entre las conexiones agregadas en ***dst_host_srv_count***	Continuo
dst_host_srv_diff_host_rate	El porcentaje de conexiones que fueron a diferentes máquinas de destino, entre las conexiones agregadas en ***dst_host_srv_count***	Continuo
dst_host_serror_rate	El porcentaje de conexiones que han activado la bandera (4) s0, s1, s2 o s3, entre las conexiones agregadas en ***dst_host_count***	Continuo
dst_host_srv_serror_rate	El porcentaje de conexiones que han activado la bandera (4) s0, s1, s2 o s3, entre las conexiones agregadas en ***dst_host_srv_count***	Continuo
dst_host_rerror_rate	El porcentaje de conexiones que han activado la bandera (4) REJ, entre las conexiones agregadas en ***dst_host_count***	Continuo
dst_host_srv_rerror_rate	El porcentaje de conexiones que han activado la bandera (4) REJ, entre las conexiones agregadas en ***dst_host_srv_count***	Continuo

Herramienta de análisis: WEKA

WEKA, acrónimo de Waikato Environment for Knowledge Analysis, es un entorno para experimentación de análisis de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario. Para ello únicamente se requiere que los datos a analizar se almacenen con un cierto formato, conocido como ARFF (Attribute-Relation File Format) (De la Cruz, 2003; Fowler & Hammel, 2014).

WEKA se distribuye como software de libre distribución desarrollado en Java (García, 2015). Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación,

y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados.

Características técnicas del sistema

Las pruebas se realizaron en un equipo informático con la siguiente configuración: Dell Inspiron 3443, procesador Intel Core i5-5200 a 2.20GHz, 4GB de RAM, 1TB de disco duro y sistema operativo Windows 7 Professional de 64 bits.

RESULTADOS Y DISCUSIÓN

A continuación, se presentan los resultados obtenidos a lo largo de la aplicación de varios algoritmos para la selección de atributos, discretización y clasificación, al igual que como trabajo base se ha tomado el conjunto de datos NSL-KDD (Castellanos & García, 2020) para la obtención de modelos para la detección de intrusos.

Replicación de resultados

En la Tabla 6 se muestran los resultados de los algoritmos KNN-1 y Random Forest, los cuales fueron los que mejores resultados presentaron tanto en los estudios anteriores como en la replicación de los mismos. Para el procesamiento se utilizaron los algoritmos:

Preprocesamiento

- ***Discretización:*** Fayyad & Irani e Intervalos de Igual Frecuencia.
- ***Selección de atributos:*** CFS, CNS, C4.5 (J48) y Naïve Bayes.

Clasificadores:

- Furia
- KNN-1
- C4.5 (J48)
- Random Forest
- SMO Polinúcleo
- TAN

Tabla 6. Resultados de los algoritmos KNN-1 y Random Forest.

Clasificador	Origen	Discretización	Selector de atributos				
			All	CFS	CNS	C4.5	NB
KNN-1	Estudios	fay	0 99.23	0 95.65	0 98.72	0 97.70	0 88.75
		i_freq	0 99.83	0 99.48	0 99.83	0 99.83	0 93.72
	Replicación	fay	0.03 99.7	0.03 97.5	0 99.6	0 99.7	0.03 99.7
		i_freq	0 99.6	0 99.7	0.02 99.6	0.02 99.5	0 99.4

Random Forest	Estudios	**fay**	0.03 99.49	0.03 95.40	0.05 98.47	0.03 97.70	0.02 88.75
		i_freq	0.05 99.83	0.06 99.48	0.05 100.0	0.05 99.83	0.05 93.72
	Replicación	**fay**	2.36 99.7	0.83 97.5	1.12 99.7	1.11 99.7	1.03 99.6
		i_freq	2.08 99.7	0.94 99.7	0.98 99.7	0.92 99.6	1.34 99.5

Nuevos selectores de atributo:

Luego de replicación de datos se optó por la aplicación de nuevos algoritmos selectores de atributos manteniendo los algoritmos discretizadores, de esta manera se trató de reducir la dimensionalidad de los subconjuntos de datos y mantener la calidad de los clasificadores. Los nuevos selectores aplicados fueron:

- EvolutionarySearch.
- GeneticSearch.
- MultiObjectiveSearch.

En la Tabla 7 se puede observar los resultados de la aplicación de los mejores resultados con los nuevos subconjuntos obtenidos con los selectores de atributos. Como se puede observar nuevamente KNN-1 y Random Forest se destacan en su porcentaje de acierto y tiempo de construcción de modelo.

Tabla 7. Resultados de los algoritmos KNN-1 y Random Forest con nuevos selectores.

Clasificador	**Discretización**	**Selector de atributos**			
		All	**ES**	**GS**	**MOS**
KNN-1	**fay**	0.03 99.7	0.02 99.4	0.02 99.3	0.02 96.8
	i_freq	0 99.6	0 99.5	0.03 99.6	0 97.3
Random Forest	**fay**	2.36 99.7	1.39 99.5	1.11 99.3	0.78 96.8
	i_freq	2.08 99.7	1.48 99.7	1.42 99.6	1.11 97.5

Nuevos algoritmos clasificadores:Después se optó por aplicar nuevos algoritmos clasificadores a los subconjuntos de selectores de atributos: CFS, CNS, C4.5 (J48), Naïve Bayes, EvolutionarySearch, GeneticSearch, MultiObjectiveSearch. Los nuevos algoritmos clasificadores aplicados fueron PART, KNN-40, RBF-Network, MOE-Fuzzy y NNge..

De los algoritmos mencionados se pudo observar que con PART y NNge

se obtuvieron mejores resultados de acierto y tiempo de construcción del modelo, pero no tanto como los ya aplicados con anterioridad (Tabla 8).

Tabla 8. Resultados de los algoritmos KNN-1 y Random Forest con diferentes selectores.

Clasific.	Discret.	Selector de atributos							
		All	CFS	CNS	C4.5	NB	ES	GS	MOS
PART	fay	1.2 99.5	0.08 97.5	0.28 99.3	0.19 99.5	0.13 99.2	0.14 99.3	0.17 99.3	0.06 96.8
	i_freq	1.57 99.5	1.57 99.5	0.33 99.3	0.11 99.6	0.19 99.2	0.39 99.3	0.47 99.1	0.09 97.5
NNge	fay	4.57 99.5	0.39 97.2	1.64 99.5	1.5 99.5	1.69 99.6	4.23 99.2	3.98 98.9	0.37 96.6
	i_freq	3.93 99.5	1.95 99.6	2.59 99.6	1.17 99.6	1.37 99.5	2.00 99.6	3.87 99.4	1.72 96.6

Propuesta: Selección de atributos en base a frecuencia Con la aplicación de todos los selectores de atributos antes mencionados se determinó que muchos de los atributos aparecen con frecuencia en muchos de los subconjuntos obtenidos, por ello se optó en seleccionar aquellos que aparecen con un mínimo de frecuencia de 40% en los 11 datasets previos. La Tabla 9 muestra los atributos seleccionados para la discretización *Fayyad & Irani*, y la Tabla 10 muestra los atributos obtenidos para la discretización por *Intervalos de Igual Frecuencia*.

Posteriormente se aplicaron los siguientes algoritmos clasificadores KNN-1, KNN-3, KNN-5, KNN-7, Random Forest, C4.5(J48) y PART a los subconjuntos de la Tabla 9 y 10.

Los resultados obtenidos se muestran en la Tabla 11 y 12 respectivamente, donde se puede destacar que al reducir la dimensionalidad de los subconjuntos se logró mantener el nivel de acierto en los clasificadores aplicados y su tiempo de construcción del modelo.

Tabla 9. Atributos obtenidos para la discretización Fayyad & Irani.

DISCRETIZACIÓN FAYYAD & IRANI		
SELECCIÓN DE ATRIBUTOS		
ATRIBUTO	# de veces encontrados	%
src_bytes	9	82%
dst_byte	9	82%
logged_in	6	55%
hot	6	55%
dst_host_srv_diff_host_rate	6	55%
dst_host_serror_rate	6	55%
dst_host_same_src_port_rate	5	45%
service	5	45%

Tabla 10. Atributos obtenidos para la discretización por Intervalos de Igual Frecuencia.

DISCRETIZACIÓN INTERVALOS DE IGUAL FRECUENCIA		
SELECCIÓN DE ATRIBUTOS		
ATRIBUTO	# de veces encontrados	%
service	10	91%
src_bytes	9	82%
dst_bytes	8	73%

logged_in	6	55%
dst_host_serror_rate	5	45%
hot	5	45%
flag	5	45%

Tabla 11. Resultados de clasificadores utilizando el subconjunto de la Tabla 9.

Filtro	Discret.	Algoritmo	Tiempo-Tr	AcGlobal	AcNormal	AcAtaque
mix	fay	KNN-1	0.02	**99.5**	99.5	99.5
mix	fay	KNN-3	0	99.3	99.4	99.3
mix	fay	KNN-5	0	99.3	99.4	99.2
mix	fay	KNN-7	0.01	99.2	99.4	99.1
mix	fay	Random Forest	0.95	**99.6**	99.6	99.6
mix	fay	C4.5 (J48)	0.11	99.3	99.1	99.5
mix	fay	PART	0.23	99.4	99.3	99.4

Tabla 12. Resultados de clasificadores utilizando el subconjunto de la Tabla 10.

Filtro	Discret.	Algoritmo	Tiempo-Tr	AcGlobal	AcNormal	AcAtaque
mix	i_freq	**KNN-1**	0.01	**99.3**	99.0	99.7
mix	i_freq	**KNN-3**	0.01	99.1	98.9	99.4
mix	i_freq	**KNN-5**	0.01	99.1	99.0	99.2
mix	i_freq	**KNN-7**	0.02	99.0	98.9	99.0
mix	i_freq	**Random Forest**	0.92	**99.4**	99.1	99.7
mix	i_freq	**C4.5 (J48)**	0.06	99.2	98.8	99.7
mix	i_freq	**PART**	0.13	99.2	98.9	99.4

Para el caso del subconjunto obtenido en base a su frecuencia de aparición se obtuvo ocho atributos (Tabla 9), discretizado por el algoritmo Fayyad & Irani, se puede observar en las Tabla 11 y 12 que los algoritmos KNN-1 y Random Forest siguen manteniendo la calidad de los resultados de clasificación, 99.5% y 99.6% respectivamente. Por su parte, el subconjunto de la Tabla 10 donde se obtuvo siete atributos, discretizado por el algoritmo Intervalos de Igual Frecuencia, se puede observar que los algoritmos KNN-1 y Random Forest siguen destacándose por sus resultados de clasificación, 99.3% y 99.4% respectivamente.

CONCLUSIONES

Con la comparación de todos los algoritmos clasificadores aplicados, tanto de la replicación de los resultados de trabajos investigativos como de la aplicación de nuevos selectores de atributos, nuevos clasificadores y la selección de atributos basado en la frecuencia de aparición se determinó que no se podía mejorar, pero si mantener la eficacia de los clasificadores con poca diferencia tanto en el grado de acierto como en el tiempo de construcción del modelo.

De esta forma al obtener nuevos subconjuntos a partir de la frecuencia con que éstos aparecen, previo a la aplicación de diferentes algoritmos selectores, se puede obtener un nuevo subconjunto con aquellos atributos que tienen mayor relevancia, disminuyendo así la dimensionalidad y manteniendo los resultados de clasificación.

REFERENCIAS BIBLIOGRÁFICAS

Ayuso, M. A., & Barcenilla, M. Á. (2008). ***Minería de datos: intrusiones de Red. línea***. http://www.it.uc3m.es/~jvillena/irc/practicas/07-08/IntrusionesDeRed.pdf

Castellanos Leyva, O., & García Borroto, M. (2020). Análisis y caracterización de conjuntos de datos para detección de intrusiones. ***Serie Científica de la Universidad de las Ciencias Informáticas***, ***13***(4), 39-52.

De la Cruz, E. S. (2003). Cancer Detection using the KDD Process. ***Advances in Soft Computing Algorithms***, 109.

Dhanabal, L., & Shantharajah, S. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. ***International Journal of Advanced Research in Computer and Communication Engineering***, 4(6), 446-452.

Fowler, C. A., & Hammel, R. J. (2014). Converting PCAPs into Weka mineable data. (Ponencia). ***15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing.*** Las Vegas, USA.

García Morate, D. (2015). Manual de WEKA. https://knowledgesociety.usal.es/sites/default/files/MANUAL%20WEKA.pdf

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. (Ponencia). ***Proceedings of the 2003 SIAM international conference on data mining***. San Francisco, USA.

Noureldien, A., & Yousif, I. (2016). Accuracy of machine learning algorithms in detecting dos attacks types. Science and Technology, ***6***(4), 89–92.

Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. ***International Journal of Engineering Research & Technology (IJERT), 2***(12), 1848-1853.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorban, A. (2009). A detailed analysis of the KDD CUP 99 data set. (Ponencia). ***IEEE symposium on computational intelligence for security and defense applications***. Ottawa, Canada.

Tribak, H. (2012). ***Análisis estadístico de distintas técnicas de inteligencia artificial en detección de intrusos.*** (Tesis doctoral). Universidad de Granada.

Ugochukwu, C. J. (2019). ***An intrusion detection system using machine learning algorithm.*** Lambert Academic Publishing.