ANALES DE LA
Academia de Ciencias de Cuba

CIENCIAS TÉCNICAS

Artículo original de investigación

La reducción de datos y el procesamiento en tiempo real aplicados a la detección de intrusos

Vitali Herrera-Semenets [1*] https://orcid.org/0000-0001-7094-2835
Raudel Hernández-León [1] https://orcid.org/0000-0001-9107-7887
Osvaldo Andrés Pérez-García [2] https://orcid.org/0000-0002-4516-8275
Andrés Gago-Alonso [3] https://orcid.org/0000-0001-5725-3410

[1] *Centro de Aplicaciones de Tecnologías de Avanzada. La Habana, Cuba*
[2] *Empresa de Tecnologías de la Información y Servicios Telemáticos. La Habana, Cuba*
[3] *Idea-Soluciones de Inteligencia Artificial, Brasil*

[*] *Autor para la correspondencia:* vherrera@cenatav.co.cu

Editor

Lisset González Navarro
Academia de Ciencias de Cuba.
La Habana, Cuba

Traductor

Darwin A. Arduengo García
Academia de Ciencias de Cuba.
La Habana, Cuba

RESUMEN

Introducción: La detección de intrusiones es una tarea crucial para identificar actividades maliciosas en sistemas informáticos. Sin embargo, los conjuntos de datos utilizados para entrenar clasificadores suelen ser voluminosos, lo que puede afectar la eficiencia del proceso. Por lo tanto, es necesario reducir el tamaño de estos conjuntos sin comprometer la eficacia de los clasificadores. **Objetivo**: Presentar un algoritmo híbrido que permita reducir eficientemente el conjunto de datos utilizado en la detección de intrusiones, sin afectar de manera significativa la eficacia de los clasificadores. **Métodos:** El algoritmo propuesto combina 2 enfoques: selección de atributos y selección de instancias. Se aplica de forma secuencial para lograr una reducción óptima del conjunto de datos sin afectar significativamente la eficacia durante la clasificación. **Resultados:** Los resultados obtenidos demuestran que el algoritmo propuesto supera a los algoritmos del estado del arte en términos de eficiencia y eficacia. Además, su aplicación en escenarios de detección de intrusos tiene un impacto significativo, ya que acelera el proceso de detección sin comprometer la calidad de los resultados. **Conclusiones:** Se ofrece una solución práctica y efectiva para la detección de intrusiones, especialmente en entornos de procesamiento de datos en tiempo real.

Palabras clave: reducción de datos; selección de atributos; selección de instancias; detección de intrusos

Data reduction and real time processing applied to intrusion detection

ABSTRACT

Introduction: Intrusion detection is a crucial task for identifying malicious activities in computer systems. However, the datasets used to train classifiers are often large, which can

Premio Nacional Anual de la Academia de Ciencias de Cuba, 2022

impact the efficiency of the process. Therefore, it is necessary to reduce the size of these datasets without compromising the effectiveness of the classifiers. **Objective:** To present a hybrid algorithm that efficiently reduces the dataset used in intrusion detection without significantly affecting classifier performance. **Methods:** The proposed algorithm combines two approaches: attribute selection and instance selection. It is applied sequentially to achieve optimal data reduction without significantly impacting effectiveness during classification. **Results:** The proposed algorithm outperforms state-of-the-art algorithms in terms of efficiency and effectiveness. Furthermore, its application in intrusion detection scenarios has a significant impact, accelerating the detection process without compromising result quality. **Conclusions:** It is provided a practical and effective solution for intrusion detection, especially in real-time data processing environments.

Keywords: data reduction; feature selection; instance selection; intrusion detection

INTRODUCCIÓN

Los clasificadores tienen cierta complejidad temporal que depende de varios parámetros. [1] En los escenarios de detección de intrusos el tamaño del conjunto de datos es un parámetro que tiene un gran peso en el rendimiento de los clasificadores. Cuando el tamaño del conjunto de datos supera el volumen de datos que el clasificador puede manejar, puede degradarse su rendimiento considerablemente, haciendo que su uso no sea factible en escenarios reales. La reducción de datos, como estrategia de preprocesamiento, se considera la etapa crucial en el proceso de minería de datos. [2] El empleo de estas estrategias permite reducir la complejidad de la etapa de entrenamiento, así como mejorar la calidad y el rendimiento del clasificador.

Existen 3 enfoques fundamentales en los que se pueden agrupar los algoritmos de reducción de datos en escenarios de detección de intrusos: selección de atributos, selección de instancias e híbridos. En los escenarios abordados en este trabajo no es posible decir que la selección de instancias sea más ventajosa que la selección de atributos o viceversa. Si solo se aplica la selección de atributos, puede quedar información innecesaria en las instancias, y si solo se aplica la selección de instancias, pueden permanecer características no representativas del conjunto de datos. En ambos casos pudieran preservarse datos innecesarios que afecten el rendimiento del clasificador.

Una vía directa y efectiva para tratar este problema es el empleo de un enfoque híbrido, es decir, la combinación de selección de atributos con selección de instancias. El empleo de dicho enfoque proporciona una mayor reducción del costo computacional de los clasificadores durante la etapa de entrenamiento que el uso individual de algoritmos de selección de atributos e instancias. [3]

El costo temporal de los algoritmos de selección de instancias está directamente asociado al tamaño del conjunto de entrenamiento, [4] lo cual puede afectar su eficiencia en los escenarios abordados en este trabajo. El empleo de una estrategia híbrida añade un costo adicional en cuanto a tiempo de ejecución, ya que además del tiempo empleado para la selección de instancias, se agrega el tiempo que toma seleccionar los atributos representativos.

Esta razón pudiera ser la causa por la cual resulta difícil encontrar en la literatura propuestas de estrategias híbridas orientadas a escenarios de detección de intrusos. Una estrategia híbrida de reducción de datos para ser aplicada en escenarios de detección de intrusos, utilizando el paradigma de la computación en la nube, [5] es propuesta por Chen *et al*. [6]

Para la selección de atributos proponen el uso del algoritmo OneR, [7] con el cual se seleccionan los 12 atributos con menor error total para representar el conjunto de datos reducido. Para la selección de instancias emplean el algoritmo de agrupamiento Affinity Propagation (AP). [8] AP es un algoritmo muy costoso, al punto que puede consumir 4 GB de memoria RAM procesando un conjunto de datos de solo 6000 instancias, cada una representada por 41 atributos.

Una vía para hacer posible la aplicación de AP en estos escenarios es utilizar un modelo de programación distribuida y paralela como es MapReduce. Esta alternativa fue utilizada por los autores para buscar una solución factible de AP sobre grandes volúmenes de datos. A pesar de que la versión distribuida mejora la eficiencia con respecto a la ejecutada en un solo ordenador, toma 100 seg reducir un conjunto de datos conformado por 6000 instancias y 41 atributos. Si se tiene en cuenta que por lo general los conjuntos de datos de entrenamiento en estos escenarios suelen estar conformados por cientos de miles de instancias, su uso en escenarios reales puede no ser factible.

Este trabajo tiene como objetivo proponer un algoritmo híbrido de reducción de datos que permita eficientemente reducir el conjunto de datos, sin afectar significativamente la eficacia de los clasificadores.

MÉTODOS

Como se muestra en la figura 1, en el algoritmo híbrido propuesto (MHS) se aplica de forma secuencial la selección de atributos y la selección de instancias. El primer paso consiste en seleccionar los atributos más representativos del conjunto de datos de entrenamiento D. Las medidas de selección de atributos más utilizadas en escenarios de detección de intrusos se pueden agrupar en 3 categorías: basadas en entropía (Information Gain, Gain Ratio, y Symmetric Uncertainty), basadas en estadística (Chi-square), y basadas en instancias (Relief y ReliefF). [9]

La estrategia utilizada en MHS se basa en emplear una medida representativa de cada categoría, ya que cada categoría puede medir distinta información cualitativa en los atributos. [10] Existen estudios comparativos donde Information Gain (IG) y Chi-square (CS) están reportadas como las medidas de selección de atributos más efectivas para tareas de clasificación. [11] Desde el punto de vista cualitativo, IG mide la cantidad de información que un atributo puede proveer al proceso de determinar si una instancia pertenece a una clase u otra.

Por otra parte, CS es una medida estadística no paramétrica que estima la correlación entre la distribución de un atributo y la distribución de la clase. Cualitativamente se puede decir que mide el grado de dependencia de un atributo respecto a una clase. Finalmente, del grupo de medidas basadas en instancias, ReliefF (RfF) se ajusta mejor que Relief (Rf) al escenario abordado en este trabajo.

La causa principal se debe a que la medida Rf está orientada a la clasificación binaria, mientras que RfF hace efectivo el trabajo con conjuntos de datos multiclase, característica común en los escenarios de detección de intrusos. Cualitativamente RfF mide que tan bien un atributo puede diferenciar instancias de distintas clases, para ello se basa en la búsqueda de los vecinos más cercanos de instancias con clases iguales y diferentes. A partir del análisis realizado, las medidas IG, CS y RfF son escogidas para realizar la selección de atributos.

Con las medidas seleccionadas se procede a obtener el conjunto de puntuaciones P_M asignadas por la medida M a los atributos que forman el conjunto de entrenamiento D. Luego se calcula la media $m_M$de las puntuaciones asignadas por M y se recorren todas las puntuaciones en P_M, seleccionando los atributos cuya puntuación $m_a \geq \bar{m}_M$. Finalmente se retorna el conjunto de atributos seleccionados AM para la medida M. Es importante resaltar que las 3 medidas se ejecutan de forma paralela, lo cual aporta mayor eficiencia.

Luego de obtener los conjuntos de atributos seleccionados por cada medida, se realiza una unión entre ellos, obteniéndose el conjunto final A. El último paso del proceso es crear el conjunto de datos reducido $D1$ como resultado de representar a D con los atributos en A. Con el conjunto de datos reducido horizontalmente, se procede al segundo paso que consiste en la reducción de instancias. La estrategia que se propone consta de 3 etapas principales: generación de etiquetas, reetiquetado y eliminación de duplicados (ver figura

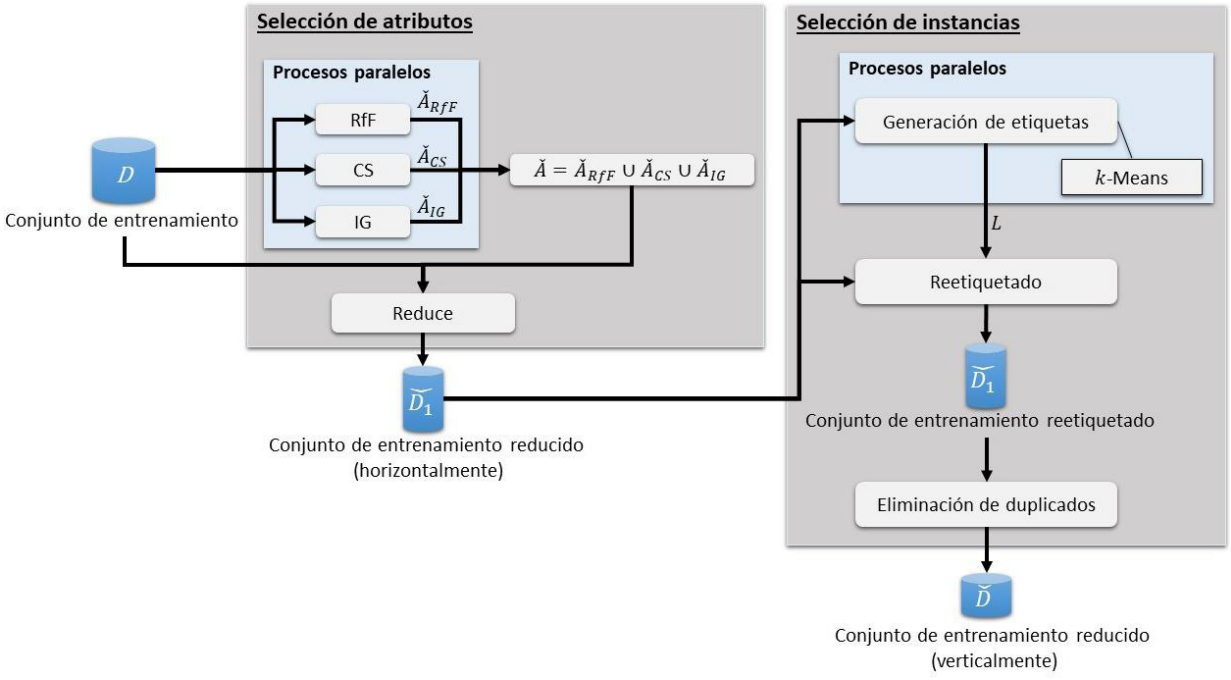

Fig. 1. Esquema del proceso de reducción de datos ejecutado por MHS

1). El proceso fundamental de la etapa de generación de etiquetas es la discretización de los atributos continuos. Con ello se busca igualar las instancias similares cuyos valores de atributos continuos sean ligeramente diferentes.

Existen 2 enfoques muy utilizados para llevar a cabo el proceso de discretización: supervisado y no supervisado. Los métodos supervisados hacen un uso intensivo de las clases para particionar los atributos continuos. [12] Por otra parte, los métodos no supervisados no tienen en cuenta la clase, lo cual los hace más eficientes que los supervisados. [12,13] En algunos trabajos comparativos reportados se utiliza el algoritmo k-Means como método de discretización no supervisado. [14,15] En estos estudios se concluye que k-Means obtiene resultados más consistentes y favorables que los demás métodos no supervisados evaluados. Además, k-Means mantiene la distribución original del atributo, lo cual hace que los resultados sean similares a los obtenidos por los métodos supervisados. Teniendo en cuenta esto y la eficiencia que brinda, k-Means fue seleccionado como parte de la estrategia para agrupar los valores continuos más cercanos y asociarlos a una única etiqueta.

El resultado de este proceso es un diccionario de etiquetas L que se utiliza en la etapa de reetiquetado para sustituir los valores continuos por sus etiquetas correspondientes. Con el proceso de reetiquetado se obtiene un conjunto de datos que contiene las mismas instancias que $D1$ solo que en el lugar de los valores continuos se encuentran sus etiquetas correspondientes.

Finalmente, se realiza la reducción de instancias sobre $D1$. La etapa de eliminación de duplicados, como su nombre indica, consiste en eliminar las instancias duplicadas de $D1$. Una instancia es considerada duplicada si existe al menos otra instancia con los mismos valores de atributos y clase. De esta forma aquellas instancias que eran semejantes y pasaron a ser iguales luego del proceso de reetiquetado serán representadas mediante una sola instancia. El resultado final es un conjunto de entrenamiento reducido D'.

RESULTADOS

La evaluación del algoritmo híbrido secuencial propuesto fue realizada sobre los conjuntos de datos KDD'99 [16] y CDMC 2013. [17] El conjunto de datos KDD'99 es considerado de referencia para tareas de detección de intrusos. [18] La cantidad de instancias que conforman el conjunto de entrenamiento es 494 021, mientras que el conjunto de prueba contiene 311 029. Cada instancia está compuesta por 41 atributos, de los cuales 9 son discretos y 32 son continuos.

En el caso de CDMC 2013 fue necesario dividir los datos en 2 conjuntos. Sin pérdida de generalidad se dividió CDMC 2013 en un conjunto de entrenamiento con 40 000 instancias y un conjunto de prueba con 37 959 instancias. Cada instancia está conformada por 7 atributos numéricos, además del atributo que define la clase.

En estos experimentos se clasifican todas las instancias en 2 tipos: "ataque" o "normal". Los resultados obtenidos por MHS fueron comparados con el algoritmo propuesto en Chen T *et al*, [6] descrito anteriormente. El mencionado algoritmo requiere de 8 nodos distribuidos, cada uno equipado con un procesador Quad-Core a 2,5 GHz y 4 GB de memoria RAM para lograr un rendimiento adecuado. Teniendo en cuenta esto, es válido resaltar que MHS fue ejecutado en un solo nodo de los utilizados por los autores citados.

El primer experimento fue dirigido a evaluar la eficiencia del proceso de reducción. Para ello se tomó el conjunto de entrenamiento de KDD'99, que posee el mayor volumen de información, y se procedió a crear 4 subconjuntos de datos de distintos tamaños. En la figura 2 se muestra el tiempo empleado por MHS y a la propuesta de Chen *et al*. [6] para reducir cada uno de los subconjuntos creados y el conjunto de entrenamiento original. En dicha figura se aprecia que independientemente del valor de k utilizado para reducir el número de instancias, el algoritmo propuesto en este capítulo es más eficiente que la propuesta de Chen T. *et al.* [6] y escala linealmente con respecto a k.

El siguiente experimento se centró en evaluar la eficacia de los clasificadores con los conjuntos de datos reducidos. El algoritmo presentado en Chen T. *et al*. [6] requiere que se defina la cantidad de atributos que se van a seleccionar para conformar su conjunto de atributos reducido. Para hacer una comparación más justa, se decidió que la cantidad de atributos fuera la misma que la seleccionada por MHS.

Además de la medida de Accuracy (Acc) utilizada para estimar la eficacia fueron empleadas otras 2 medidas de calidad: Recall y FPR. [4] Con las medidas utilizadas es posible comprobar si se afectó la calidad del conjunto de datos de entrenamiento reducido con respecto al original.

La tabla 1 y la tabla 2 muestran los resultados alcanzados sobre los conjuntos de datos de KDD'99 y CDMC 2013 respectivamente, utilizando los clasificadores SVM y KNN. Los resultados están dados por las 3 medidas de calidad anteriormente mencionadas, así como por la cantidad de instancias que conforman los conjuntos de datos reducidos.

En ambas tablas se puede apreciar que a medida que aumenta el valor k en el algoritmo MHS, mejora no solo la eficacia de los clasificadores, sino que también mejoran las medidas Recall y FPR. Esto se debe a que el valor de k es inversamente proporcional a la cantidad de instancias que se obtienen tras la reducción; por tanto, mientras mayor sea k, el

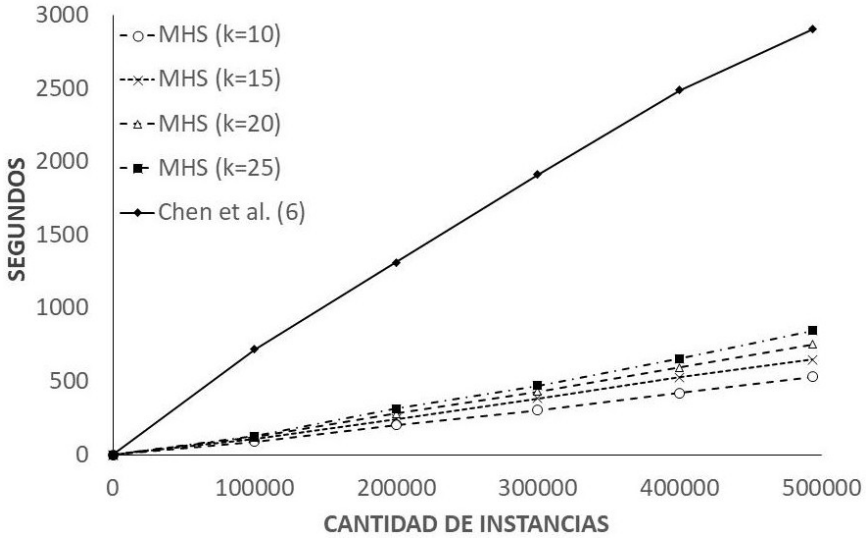

Fig. 2 Tiempo de ejecución de MHS (para diferentes valores de k) y de la propuesta de Chen et al. (6) sobre diferentes cantidades de instancias de KDD'99

conjunto de datos original se reduce en menor medida, conservándose más información que es utilizada por el clasificador para alcanzar mejores resultados.

En la tabla 1 se observa que para k = 25 el algoritmo MHS obtiene un conjunto de datos más reducido que el algoritmo propuesto en Chen T. *et al*.; [6] y al mismo tiempo, los resultados obtenidos por los clasificadores utilizando MHS (k = 25) como algoritmo de reducción superan en cuanto a eficacia Recall y FPR a los alcanzados utilizando la propuesta de Chen T. *et al*. [6] Un resultado similar se puede observar en la tabla 2 utilizando MHS (k = 20). Los resultados muestran que MHS puede reducir en mayor medida el conjunto de entrenamiento que la propuesta de Chen T. *et al*. [6] preservando más información útil, lo cual se evidencia con mejores resultados en las medidas evaluadas. Con respecto al conjunto de entrenamiento original no se aprecia una diferencia considerable en cuanto a los resultados obtenidos por los clasificadores utilizando MHS (k = 25) para el caso del conjunto de datos KDD'99 y MHS (k = 20) para CDMC 2013.

No obstante, para comprobar si los resultados alcanzados, en términos de eficacia, por los clasificadores evaluados con distintos conjuntos de datos en este experimento son estadísticamente diferentes, se realizó la prueba de Friedman [19] y posteriormente el procedimiento post-hoc Bergmann-Hommel, [20] como se sugiere en varios trabajos. [21-23] Los resultados estadísticos obtenidos se presentan mediante un diagrama de diferencia crítica (CD, por sus siglas en inglés). El diagrama CD muestra de manera compacta el orden de los clasificadores a partir del ranking obtenido por la prueba de Friedman, la magnitud de las diferencias entre ellos y la significación de dichas diferencias. La posición de los clasificadores en el segmento representa su valor en el ranking, siendo el clasificador que se encuentra más a la derecha el de mejor resultado. Si 2 o más clasificadores comparten una línea gruesa indica que tienen un comportamiento estadísticamente similar.

Como se puede observar en la figura 3, la posición que ocupa el conjunto de datos original en el segmento indica que con él se obtienen los mejores resultados durante la clasificación. Como se señaló anteriormente, una mayor reducción del conjunto de datos implica una mayor pérdida de información que conlleva a tener una afectación significativa en la eficacia, lo cual se puede apreciar en los resultados obtenidos con MHS (k = 10) y MHS (k = 15). No obstante, aún con una reducción considerable del conjunto de datos, los valores de eficacia alcanzados con MHS (k = 20), Chen *et al*., [6] MHS (k = 25) y el conjunto de datos original, tienen un comportamiento estadísticamente similar.

DISCUSIÓN

Los sistemas de detección de intrusos son considerados sistemas bajo ataque, donde los atacantes modifican la forma de llevar a cabo sus "ataques" con el propósito de hacer fallar el sistema de detección. [24] Por tanto, los sistemas deben ser capaces de adaptarse a los cambios en los datos, actualizando los patrones característicos de "ataques" periódicamente para mantener la eficacia al paso del tiempo. La incorporación de MHS, como etapa de preprocesamiento, aporta eficiencia a la etapa de generación o descubrimiento de patrones, haciendo posible que se puedan actualizar con mayor frecuencia y se conserve la eficacia de los clasificadores.

Tabla 1. Resultados alcanzados sobre el conjunto de datos KDD'99

Clasificador	Algoritmo	Acc.	Recall	FPR	Instancias
SVM	Original	92,30	90,79	1,75	494 021
	MHS(k=10)	87,03	86,13	2,68	22 822
	MHS(k=15)	88,31	86,92	2,42	38 679
	MHS(k=20)	88,83	87,32	2,14	55 999
	MHS(k=25)	90,26	88,54	1,97	60 540
	Chen *et al.* [6]	90,05	87,88	2,12	63 234
KNN	Original	92,81	91,21	0,73	494 021
	MHS(k=10)	87,12	86,68	2,23	22 822
	MHS(k=15)	87,93	87,59	1,62	38 679
	MHS(k=20)	89,10	88,43	1,18	55 999
	MHS(k=25)	89,97	89,13	0,98	60 540
	Chen *et al.* [6]	89,51	88,81	1,12	63 234

El empleo de MHS supera en varios aspectos al trabajo reportado en la literatura para estos escenarios. En términos de eficiencia, los resultados experimentales mostraron un mayor rendimiento con respecto a Chen T. *et al.* [6] Es válido resaltar que los recursos de cómputo que requiere MHS para ejecutarse de forma adecuada son inferiores a los reportados en Chen T. *et al.*, [6] lo cual contribuye a que bien pudiera ser utilizado en entornos que no cuenten con grandes recursos de cómputo para llevar a cabo este tipo de tareas.

Otro aspecto en el que sobresale MHS es que la eficacia que alcanzan los clasificadores evaluados empleando el conjunto de datos reducido, no muestra diferencia significativa con respecto a la reportada utilizando el conjunto de entrenamiento original.

Además, MHS fue evaluado, en otro trabajo, como etapa de preprocesamiento en un escenario real para la detección de intrusos basada en reglas. [4] Los resultados muestran que el empleo de MHS tuvo un impacto significativo en cuanto al desempeño de los sistemas de detección, ya que contribuye a generar reglas menos complejas, permitiendo que el proceso de detección de intrusos sea más rápido, aspecto fundamental en el procesamiento de datos en tiempo real.

Estos resultados permiten incorporar conocimientos propios a nuestros sistemas, evitando la inversión en sistemas foráneos con los altos costos de compra y soporte que implican. Todo lo expresado anteriormente coadyuva a un mejor procesamiento y análisis de la información generada diaria-

Tabla 2. Resultados alcanzados sobre el conjunto de datos CDMC 2013

Clasificador	Algoritmo	Acc.	Recall	FPR	Instancias
SVM	Original	96,42	95,92	0,21	40000
	MHS (k=10)	91,47	91,32	2,11	239
	MHS (k=15)	94,16	92,68	1,54	500
	MHS (k=20)	94,76	93,13	0,88	825
	Chen et al. (6)	94,21	92,97	1,23	984
	MHS (k=25)	95,41	94,04	0,37	1086
KNN	Original	96,57	96,10	0,11	40000
	MHS (k=10)	92,25	91,87	1,83	239
	MHS (k=15)	94,72	93,30	1,08	500
	MHS (k=20)	95,11	94,19	0,62	825
	Chen et al. (6)	94,78	93,89	1,02	984
	MHS (k=25)	95,87	94,93	0,27	1086

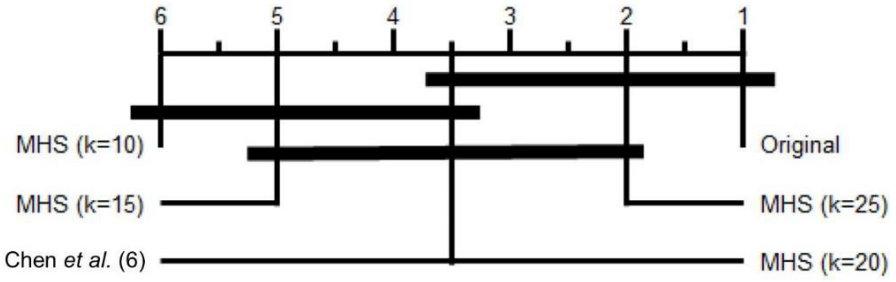

Fig. 3 Diagrama CD con una comparación estadística de la eficacia alcanzada por los clasificadores utilizando distintos algoritmos híbridos

mente por la prestación de servicios en el sector financiero y de las telecomunicaciones, contribuyendo a un mayor control y desarrollo en dichos sectores.

Conclusiones

A partir de una eficaz combinación de estrategias y heurísticas que garantizan la consistencia de MHS, se alcanza una eficiencia superior a otro algoritmo diseñado para escenarios de detección de intrusos. Lo anterior permite a los clasificadores generar los modelos de clasificación en un menor tiempo, lo cual permite una respuesta más rápida del sistema de detección ante nuevos ataques.

Además, MHS permite a los clasificadores evaluados alcanzar una eficacia superior que la obtenida utilizando el algoritmo comparado. Todo esto sin afectar significativamente la eficacia alcanzada utilizando el conjunto de entrenamiento original. La aplicación de MHS en escenarios de detección de intrusos tiene un impacto significativo en la práctica ya que su empleo influye en que el proceso de detección de intrusos sea más rápido, lo cual es un aspecto fundamental en el procesamiento de datos en tiempo real.

REFERENCIAS BIBLIOGRÁFICAS

1. García S, Luengo J, Herrera F. Data preprocessing in data mining: Springer; 2016.
2. Aggarwal CC. Data mining: the textbook: Springer; 2015.
3. Tsai CF, Eberle W, Chu CY. Genetic algorithms in feature and instance selection. Knowledge-Based Systems. 2013;39:240-7.
4. Herrera-Semenets V, Pérez-García OA, Hernández-León R, van den Berg J, Doerr C. A data reduction strategy and its application on scan and backscatter detection using rule-based classifiers. Expert Systems with Applications. 2018;95:272-9.
5. Marinescu DC. Cloud computing: theory and practice: Morgan Kaufmann; 2017.
6. Chen T, Zhang X, Jin S, Kim O. Efficient classification using parallel and scalable compressed model and its application on intrusion detection. Expert Systems with Applications. 2014;41(13):5972-83.
7. Holte RC. Very simple classification rules perform well on most commonly used datasets. Machine learning. 1993;11(1):63-90.
8. Bodenhofer U, Kothmeier A, Hochreiter S. APCluster: an R package for affinity propagation clustering. Bioinformatics. 2011;27(17):2463-4.
9. Liu W, Liu S, Gu Q, Chen J, Chen X, Chen D. Empirical studies of a two-stage data preprocessing approach for software fault prediction. IEEE Transactions on Reliability. 2015;65(1):38-53.
10. Herrera-Semenets V, Bustio-Martínez L, Hernández-León R, van den Berg J. A multi-measure feature selection algorithm for efficacious intrusion detection. Knowledge-Based Systems. 2021 September; 227.
11. George Forman. An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research. 2003;3:1289-305.
12. Mizianty M, Kurgan L, Ogiela M. Comparative analysis of the impact of discretization on the classification with naive bayes and semi-naive bayes classifiers. Seventh International Conference on Machine Learning and Applications. 2008;823-8.
13. Daniela Joicta. Unsupervised static discretization methods in data mining. Titu Maiorescu University, Bucharest, Romania. 2010.
14. Dash R, Paramguru RL, Dash R. Comparative analysis of supervised and unsupervised discretization techniques. International Journal of Advances in Science and Technology. 2011;2(3):29-37.
15. Maslove M, Podchiyska T, Lowe HJ. Discretization of continuous features in clinical datasets. Journal of the American Medical Informatics Association. 2012;20(3):544-53.
16. Ozgur Atilla and Erdem Hamit. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ PrePrints. 2016;4.
17. Song J. CDMC2013 intrusion detection dataset. Department of Science & Technology Security, Korea Institute of Science and Technology Information (KISTI). 2013.
18. Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A Survey of Network-based Intrusion Detection Data Sets. Computers & Security. 2019;86:147-67.
19. Demvsar J. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research. 2006;7:1-30.
20. García S, Herrera F. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research. 2008;9:2677-94.
21. García S, Fernández A, Luengo J, Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences. 2010;180(10):2044-64.

22. Derrac, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation. 2011;1(1):3-18.
23. Loyola-González. Supervised classifiers based on emerging patterns for class imbalance problems. Tesis doctoral. Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Tonantzintla, Puebla, Mexico. 2017.
24. Herrera-Semenets V, Pérez-García OA, Gago-Alonso A, Hernández-León R. Classification rule-based models for malicious activity detection. Intelligent Data Analysis. 2017;21(5):1141-54.

Recibido: 21/02/2024
Aprobado: 21/03/2024

Agradecimientos

Nuestro sincero agradecimiento al Dr. Jan van den Berg y al Dr. Christian Doerr por facilitar el uso de datos reales para evaluar nuestra propuesta y por sus sugerencias para el desarrollo de la investigación. Agradecemos además a los doctores Lázaro Bustio Martínez, José K. Febrer Hernández, Niusvel Acosta Mendoza y al Ms. C. Mario A. Prado Romero por su colaboración durante las investigaciones realizadas.

Conflictos de intereses

Los autores declaran que no tienen conflictos de intereses económicos o relaciones personales que pudieran haber influido en el resultado del trabajo reportado en este documento.

Contribuciones de los autores

Conceptualización: Vitali Herrera Semenets
Análisis formal: Raudel Hernández León
Investigación: Vitali Herrera Semenets, Raudel Hernández León, Osvaldo Andrés Pérez García, Andrés Gago Alonso
Metodología: Osvaldo Andrés Pérez García
Administración del proyecto: Vitali Herrera Semenets
Software: Vitali Herrera Semenets
Validación: Vitali Herrera Semenets
Redacción-borrador original: Vitali Herrera Semenets
Redacción-revisión y edición: Raudel Hernández León, Osvaldo Andrés Pérez García, Andrés Gago Alonso

Financiamientos

Esta investigación no utilizó una fuente de financiamiento específica.

Cómo citar este artículo

Herrera-Semenets V, Hernández-León R. Andrés Pérez-García O, Gago-Alonso A. La reducción de datos y el procesamiento en tiempo real aplicados a la detección de intrusos. An Acad Cienc Cuba [internet] 2024 [citado en día, mes y año];14(1):e1540. Disponible en: http://www.revistaccuba.cu/index.php/revacc/article/view/1540