REVISTA CUBANA
DE TRANSFORMACIÓN DIGITAL

RNPS 2487 • ISSN 2708-3411
Vol. 4 • Nro. 2 • abril-junio 2023 • e199

ARTÍCULO ORIGINAL

Una aplicación del algoritmo *proactive forest* para la detección de bots malignos

An Application of the Proactive Forest Algorithm for the Detection of Malicious Bots

Daniel Pardo Echevarría
dannypardo279903@gmail.com • https://orcid.org/0000-0002-2702-2846

Nayma Cepero Pérez
ncepero@ceis.cujae.edu.cu • https://orcid.org/0000-0003-3808-8135

Humberto Díaz Pando
hdiazp@ceis.cujae.edu.cu • https://orcid.org/0000-0003-1591-87811

UNIVERSIDAD TECNOLÓGICA DE LA HABANA "JOSÉ ANTONIO ECHEVERRÍA", CUJAE, CUBA

Recibido: 2023-01-26 • Aceptado: 2023-03-28

RESUMEN

Los *bots* malignos son programas informáticos que tienen la particularidad de simular la actividad humana, empleándose para ejecutar ataques cibernéticos. Estos programas resultan un problema que afecta a muchos servicios webs. Por eso se han desarrollado múltiples aproximaciones para detectarlos, con gran repercusión en la aplicación de algoritmos de aprendizaje automático, sobre todo los que generan modelos clasificadores a partir del aprendizaje supervisado. En este trabajo nos proponemos aplicar el algoritmo Proactive Forest (PF) en la detección de *bots* malignos, evaluando su rendimiento en base al porcentaje de instancias correctamente clasificadas como *bot* maligno o usuario humano y realizando adicionalmente una comparación con el algoritmo Random Forest (RF), pues también genera un bosque de decisión implementado en un artículo del estado del arte, para la detección de *bots* malignos. Los resultados permiten apreciar un máximo del rendimiento del algoritmo Proactive Forest, del 63,14 % de instancias correctamente clasificadas.

Palabras clave: detección de *bots*, clasificación, bosque de decisión, árbol de decisión.

ABSTRACT

Malicious bots are computer programs that have the particularity of simulating human activity, being used to execute cyber-attacks. These programs are a problem that affects multiple web services. As a result, multiple approaches have been developed to detect them. The application of machine learning algorithms, especially those that generate classifier models based on supervised learning, has had a great impact. The present work proposes the application of the Proactive Forest (PF) algorithm in the detection of malicious bots. Evaluating its performance, based on the percentage of instances correctly classified as malicious bot or human user. Performing additionally, a comparison with the Random Forest (RF) algorithm, being an algorithm that also generates a decision forest. Implemented in a state-of-the-art article, for the detection of malicious bots. The results achieved show a maximum performance of the Proactive Forest algorithm of 63,14 % of correctly classified instances.

Keywords: *bot detection, classification, decision forest, decision tree.*

INTRODUCCIÓN

Cada vez es más frecuente que ocurran violaciones y ataques a la seguridad informática, debido al masivo uso de Internet y la cantidad de información que en esta se maneja. Los ciberc-riminales utilizan diversas herramientas para realizar este tipo de acciones y una de las más empleadas en los últimos años son los *bots* (Vishwakarma, 2020).

Los *bots* son programas informáticos que actúan de forma autónoma y automática, simulando la actividad humana. Se dividen en dos tipos: *bots* benignos y *bots* malignos. Los *bots* benignos son empleados en múltiples actividades beneficiosas, formando parte de numerosos servicios web. Por otra parte, los *bots* malignos son utilizados para realizar ataques y violaciones a la seguridad, por lo que se consideran programas malignos (*malware*) (Rovetta, Suchaka y Masulli, 2020; Xu *et al.,* 2019).

El impacto de los *bots* en la actualidad es considerable. Según Imperva (2020) los *bots* representaron 37,2 % del tráfico total de usuarios en Internet en 2020, donde 24,1 % eran *bots* malignos. El problema de los *bots* ha afectado a múltiples servicios web, incluyendo los dedicados al comercio electrónico, ya que se detectó 18,1 % de *bots* malignos respecto al total de usuarios en estos sitios web (Imperva, 2020).

Por esta razón, diversas técnicas han sido empleadas para garantizar la detección de *bots* malignos y una de las más destacadas es la minería de datos, a través de diversos algoritmos de aprendizaje automático (Rovetta *et al.,* 2020; Xu *et al.,* 2019). La minería de datos explora

y analiza grandes cantidades de datos de ataques de *bots*, extrayendo información útil para desarrollar modelos que permitan predecir si un usuario es considerado como *bot* maligno o humano (Hernández, Ramírez y Ferri, 2004; Vishwakarma, 2020). Para ello se emplean diversas técnicas de aprendizaje automático (Vishwakarma, 2020).

Como parte de las técnicas de aprendizaje automático para la detección de *bots*, se destaca la aplicación de algoritmos de aprendizaje supervisado, debido a los buenos resultados (Doran, 2011; Rovetta *et al.,* 2020; Vishwakarma, 2020), a partir del desarrollo de modelos clasificadores construidos sobre una muestra de datos conocida (Mohammed, Khan & Mohammed Bashier, 2016; Vishwakarma, 2020). Los más destacados son: Naïve Bayes(NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decisión Tree(ID3) (Doran, 2011; Vishwakarma, 2020).

Entre los algoritmos de aprendizaje supervisado más relevantes para la detección de *bots*, se encuentran los basados en bosques de decisión (Vishwakarma, 2020), los cuales generan un modelo compuesto por múltiples árboles de decisión, siendo precisos y diversos (Rokach, 2015). Tal es el caso de Random Forest (RF), pues como se evidencia en Pardo, Moreno, Diaz, y Chissingui (2022), obtiene excelentes resultados en la detección de *bots*, aunque los bosques construidos a partir de este algoritmo tienden a ser poco diversos, lo que de cierta forma produce una disminución de la precisión y por ende de los resultados (Cepero-Pérez, Denis-Miranda, Hernández-Palacio, Moreno-Espino, y García-Borroto, 2018). Para contrarrestar la pérdida de diversidad de los bosques de decisión obtenidos de Random Forest, en Cepero-Pérez *et al.* (2018) los autores desarrollan el algoritmo Proactive Forest(PF), lo cual podría traducirse en mejores resultados en la clasificación de *bots* malignos y humanos.

Este trabajo presenta la aplicación del algoritmo de aprendizaje supervisado Proactive Forest para la detección de *bots* malignos, evaluando a partir de un diseño experimental, si los factores identificados influyen en el rendimiento del algoritmo, basado en el porcentaje de instancias correctamente clasificadas en *bots* malignos o humanos. Además de efectuar una comparación con el algoritmo Random Forest, implementado en Pardo *et al.* (2022) para la detección de *bots* malignos en sitios de comercio electrónico, pues ambos algoritmos generan un modelo basado en un bosque de decisión.

METODOLOGÍA

Trabajos relacionados

El alto y considerable impacto de los *bots* malignos en la actualidad, ha traído consigo un incremento del estudio de diversas aproximaciones para garantizar su detección. Por ello son muchos los trabajos que integran el estado del arte que abordan diferentes técnicas y metodologías con el fin de resolver esta tarea.

Sobre todo, se destaca la minería de datos a partir de la aplicación de diversos algoritmos de aprendizaje automático, lo cual se pone en evidencia en Vishwakarma (2020), donde los autores emplean diversos algoritmos de aprendizaje automático para la detección de *bots*

malignos. Además, en Rovetta *et al.* (2020); Xu *et al.* (2019), se emplean dichos algoritmos en sitios de comercio electrónico, destacándose los buenos resultados obtenidos por los algoritmos clasificadores, pertenecientes al aprendizaje supervisado.

Como parte de las técnicas supervisadas, los bosques de decisión son generalmente utilizados, sobre todo con el empleo del algoritmo Random Forest, como se evidencia en Pardo *et al.* (2022); Rout, Lingam y Somayajulu (2020); Vishwakarma (2020), lográndose en Pardo *et al.* (2022) un máximo de 99,88 % de instancias correctamente clasificadas. Se señala además en Haq y Singh (2018); Velasco, González, Fidalgo y Alegre (2021), la importancia de la base de datos CTU-13 en la construcción de los modelos de clasificación y la evaluación de sus resultados.

Proactive Forest para la detección de bots malignos

Generalmente, los algoritmos basados en bosques de decisión generan buenos resultados al emplearse en la detección de *bots* malignos (Pardo *et al.*, 2022; Vishwakarma, 2020). Por esta razón, en este trabajo se aplica el algoritmo Proactive Forest para la detección de *bots* malignos. Este algoritmo genera un modelo de decisión compuesto por diversos árboles, más conocido como bosque de decisión Cepero-Pérez *et al.* (2018), donde los árboles de decisión generan modelos de clasificación que tienen estructura de árbol, integrados por nodos intermedios, aristas y hojas. En los nodos intermedios, incluida la raíz, se realiza la división continua de los datos, mientras que los nodos hojas contienen el valor de clase. Las aristas, conocidas como ramas, son el enlace entre cada nodo y representan las decisiones que se toman en el modelo (Dahan, Cohen, Rokach y Maimon, 2014; Rokach, 2015).

Como todo buen bosque de decisión, los modelos obtenidos por Proactive Forest se enfocan en ser diversos y precisos (Cepero-Pérez *et al.*, 2018; Rokach, 2015), sobre todo, porque dicho algoritmo surge como una mejora a la pérdida de diversidad ocasionada por Random Forest, manteniendo la calidad de precisión de cada árbol individual, donde cada árbol se construye sobre las características menos utilizadas en la generación de los anteriores (Cepero-Pérez *et al.*, 2018).

El flujo de ejecución del algoritmo se detalla en la figura 1. Vale señalar que después de cargar los datos de ataques de *bots* y preprocesarlos, se establece el número de árboles que se van a construir. En este caso se toman 250 árboles, pues a partir de este número los resultados no mejoran. Además, se establece un criterio de selección de un subconjunto de atributos del total, que contenga la base de datos, para analizar en cada nodo de cada árbol, seleccionándose el logaritmo de total de atributos en esta ocasión, pues se establece por defecto en la implementación realizada por Cepero-Pérez *et al.* (2018). Posteriormente, el conjunto de datos para construir cada árbol es elegido aleatoriamente con reemplazo.

Como parte más importante del flujo de ejecución mostrado en la figura 1, se establece que en cada nodo de cada árbol que se va a construir, se selecciona el subconjunto de atributos para dividir la muestra, de los que más probabilidad de elección tengan. Dicha probabilidad se incrementa a partir de un cálculo de la importancia de los atributos, posterior a la construcción de cada árbol, incrementando el valor de probabilidad de elección, de aquellos

con menor valor de importancia (Cepero-Pérez *et al.,* 2018). Al culminar el proceso de construcción del modelo, cuando se genere la cantidad de árboles establecida, la clasificación de una nueva instancia es obtenida a través del voto mayoritario del conjunto de árboles (Cepero-Pérez *et al.,* 2018).

Experimentos

En esta sección del documento se define la estrategia de validación, a partir de un diseño experimental, con el objetivo de evaluar la influencia de los factores identificados en el rendimiento del algoritmo, rendimiento que se mide en base al porcentaje de instancias correctamente clasificadas en *bots* malignos o humanos

Descripción de la base de datos

Se selecciona CTU-13 como base de datos para proceder con la experimentación, la cual fue desarrollada en la Universidad CTU de República Checa. Está integrada por trece escenarios, cada uno de los cuales contiene datos etiquetados correspondientes al tráfico de *bots*, obtenidos en tiempo real (Vishwakarma, 2020). Cada escenario consta de quince columnas y poseen diferentes tamaños en cuanto al flujo de datos, como se muestra en la tabla 1.

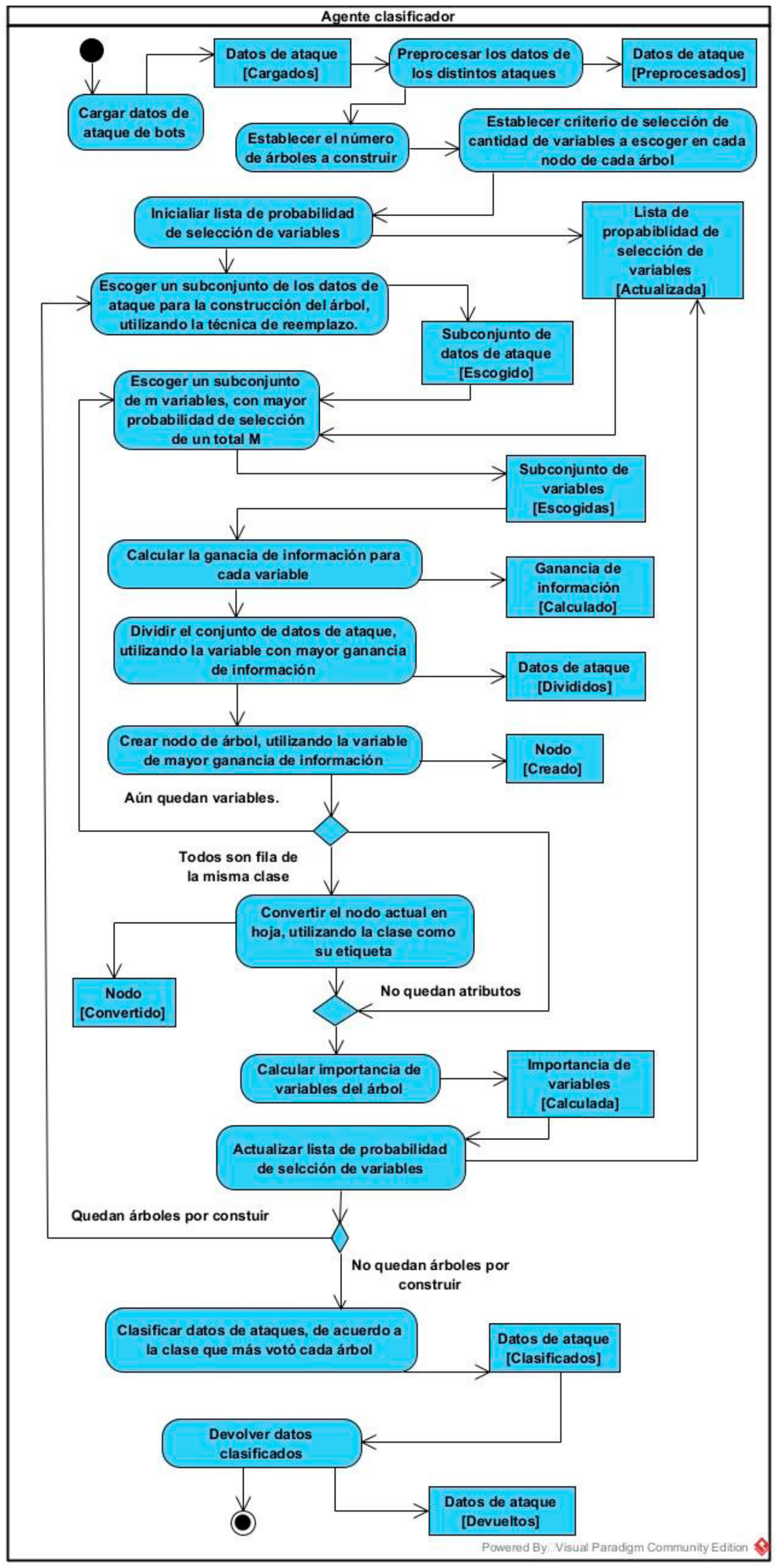

Fig. 1 Flujo de ejecución del algoritmo Proactive Forest.

Tabla 1. Flujos de datos de los escenarios de CTU-13 (tomado de Vishwakarma (2020))

Escenario	Flujos de fondo (%)	Flujos de bots (%)	Flujos normales (%)	Flujos totales
1	97,47	1,41	1,07	2 824 636
2	98,33	1,15	0,50	1 808 122
3	96,94	0.561	2,48	4 710 638
4	97,58	0,154	2,25	1 121 076
5	95,70	1,68	3.60	129 832
6	97,83	0,82	1,34	558 919
7	98,47	1,50	1,47	114 077
8	97,32	2,57	2,46	2 954 230
9	91,70	6,68	1,57	2 753 884
10	90,67	8,112	1.20	1 309 791
11	89,85	7,602	2,53	107 251
12	96,99	0,657	2,34	325 471
13	96,26	2,07	1,65	1 92 .149

Como se muestra en la tabla 1, se destacan tres flujos de datos principales: flujos normales, flujos de fondo y flujos de *bots*. Los flujos normales corresponden al tráfico de usuarios humanos. Los flujos de fondo corresponden al tráfico propio de la red, ya sea cuando no existe una actividad directa de usuarios o para ocultar la presencia de *bots* benignos, el cual presenta datos reales de tráfico indirecto de usuarios. Por otro lado, los flujos de *bots* corresponden a la navegación visible de los *bots* malignos (Vishwakarma, 2020). Como se aprecia en los flujos totales, los escenarios tienen diferentes dimensiones, destacándose los que contienen menos de un millón de instancias y los que superan dicha cifra.

Todos los escenarios contienen un total de catorce atributos. De ellos, siete son de tipo numérico y el resto de tipo categórico. Estos atributos representan características de red de un usuario, las cuales se describen en: dirección *ip,* transacciones de bytes y paquetes, tipos de estados y protocolos, duración de un supuesto ataque de *bot* maligno, entre otras (Vishwakarma, 2020).

Preprocesamiento

En la etapa de preprocesamiento de datos, primeramente se transforman todos los datos a tipo numérico, para trabajar con un solo tipo y debido a que la mitad de los atributos presentes son numéricos. Además, se sustituyen los valores vacíos por el número 0, en vez de eliminar las filas que contengan columnas de esta característica, corrigiendo así inconsistencias presentes en la muestra y para evitar la pérdida de información al eliminar una instancia. Como parte de este proceso, se cambia la etiqueta de clase a un valor de tipo numérico. Asignando a las instancias con etiqueta *background* y normal valor 0, y aquellas consideradas *bots* malignos con etiqueta *botnet* valor 1. Posteriormente se realiza una normalización de los datos, aplicando una técnica de estandarización de datos, dividiéndolos en escalas que faciliten la comparación entre las instancias y para que el algoritmo no otorgue más importancia a un atributo con respecto a los demás. A esta primera fase del preprocesamiento se le denominó limpieza de datos

Al concluir la primera fase del preprocesamiento, se realiza un análisis de componentes principales (PCA), con el objetivo de reducir la dimensionalidad de los escenarios de la base de datos, tomando una varianza acumulada explicativa a 98 % de los datos, que dio lugar a diez o doce componentes principales, dependiendo del escenario. Además, se efectuó un balance de las clases presentes, debido a la existencia de escenarios con pocas instancias de etiqueta *botnet*. Para ello se aplica la técnica SMOTE, que basándose en vecinos más cercanos crea ejemplos sintéticos de clase minoritaria, aunque esto trae como desventaja un aumento del conjunto de datos, principalmente cuando hay muy pocos ejemplos de la clase minoritaria para muchos ejemplos de la clase mayoritaria. La tabla 2 muestra los resultados al aplicar cada fase del preprocesamiento.

Señalar como parte del preprocesamiento, la selección de solo cinco mil instancias para cada clase, debido a la alta complejidad algorítmica que presenta el algoritmo Proactive Forest al entrenar el modelo de clasificación. Además, a partir de un valor porcentual de los datos de la muestra, se crea un subconjunto de datos para la prueba del algoritmo. Los ejemplos que integran este conjunto son escogidos aleatoriamente del escenario preprocesado para emplear, aunque

Tabla 2. Resultados de las fases del preprocesamiento

	Número de instancias por clase antes del preprocesamiento (con 14 atributos)			Fases del preprocesamiento				
				Limpieza de datos	Transformación y selección			
					Componentes al aplicar PCA	Balanceo		
Id	Botnet (1)	Normal (0)	Total	Instancias		Botnet (1)	Normal (0)	Total
1	40 961	2 783 675	2 824 636	2 824 636	11	2 783 675	2 783 675	5 567 350
2	20 941	1 787 181	1 808 122	1 808 122	11	1 787 181	1 787 181	3 574 362
3	26 822	4 683 816	4 710 638	4 710 638	11	4 683 816	4 683 816	9 367 632
4	2 580	1 118 496	1 121 076	1 121 076	11	1 118 496	1 118 496	2 236 992
5	901	128 931	129 832	129 832	11	128 931	128 931	257 862
6	4 630	554 289	558 919	558 919	11	554 289	554 289	1 108 578
7	63	114 014	114 077	114 077	11	114 014	114 014	228 028
8	6 127	2 948 103	2 954 230	2 954 230	11	2 948 103	2 948 103	5 896 206
9	184 987	1 902 521	2 087 508	2 087 508	12	1 902 521	1 902 521	3 805 042
10	106 352	1 203 439	1 309 791	1 309 791	10	1 203 439	1 203 439	2 406 878
11	8 164	99 087	107 251	107 251	10	99 087	99 087	198 174
12	2 168	323 303	325 471	325 471	11	323 303	323 303	646 606
13	40 003	1 885 146	1 925 149	1 925 149	11	1 885 146	1 885 146	3 770 292

es una desventaja considerar datos sintéticos generados por el balanceo de clases en el conjunto de validación, lo que añadiría un sesgo ficticio a dicho conjunto; para el caso de los escenarios de CTU-13 puede ser válido hacer esta elección, debido al principal problema que presentan sus escenarios de tener tan pocos ejemplos de datos de *bots* malignos, por lo que de no hacerlo, se podría seleccionar una cantidad de datos de *bots* en el conjunto de validación muy baja o por el contrario dejar pocos ejemplos de estos para entrenar el modelo. Además, con la técnica SMOTE empleada, solo se crearán ejemplos sintéticos a partir de datos reales de *bots* malignos.

Diseño de los experimentos

Como estrategia de validación se efectúa un diseño experimental, para evaluar el rendimiento del algoritmo Proactive Forest a partir de factores identificados, tomando como referencia el utilizado en Pardo *et al.* (2022) y así comparar los resultados alcanzados con el algoritmo Random Forest, empleando el preprocesamiento descrito en la anterior sección del documento, identificándose como rendimiento el porcentaje de las instancias correctamente clasificadas en *bots* malignos o humanos, teniendo en cuenta los siguientes factores:

- **Escenario:** tamaño de los escenarios de la base de datos CTU-13, identificando como escenarios pequeños a aquellos que tiene menos de un millón de instancias y como escenarios grandes los que superan el millón. Se toma en cuenta este factor, debido a que una característica destacable en los escenarios de la base de datos es la gran cantidad de ejemplos que contienen muchos de estos y la muy poca cantidad que contienen otros, al establecer un valor de un millón de datos como un umbral. En este trabajo se elige al onceno escenario como pequeño y al tercero como grande.
- **Porcentaje de datos empleados para la prueba del algoritmo:** el resto de dichos datos serán empleados para la construcción del modelo. Estos presentan valores que se encuentran en una escala de 0 a 1. Este factor se tiene en cuenta para realizar un análisis

del algoritmo, según la cantidad de datos con la que se entrene y pruebe el modelo generado, en base a si es mejor entrenar un predictor a un menor porcentaje de datos o si es más efectivo emplear una cantidad mayor.

La tabla 3 muestra los niveles asociados a cada factor identificado.

Tabla 3. Factores y niveles del diseño experimental

Factor	Nivel Bajo	Nivel Alto
Escenario	Pequeño	Grande
% de prueba	0,2	0,5

Para desarrollar la experimentación se realiza un diseño factorial completo, identificando el impacto de los factores en el rendimiento del algoritmo. Para los dos factores y dos niveles identificados, se generan cuatro tratamientos. Para cada tratamiento se realizaron tres réplicas, obteniendo un total de doce ejecuciones; se logró una mayor precisión en los resultados. Los tratamientos son aleatorios: se garantiza así una independencia en las observaciones (Fernández, Baptista y Hernández, 1998).

RESULTADOS Y DISCUSIÓN

Resultados del experimento

En esta sección se analizan los resultados alcanzados durante la experimentación. La tabla 4 muestra los resultados en base al porcentaje de instancias correctamente clasificadas.

Tabla 4. Resultados de la experimentación de Proactive Forest

Ejecución	Escenario	% de prueba	Proactive Forest
1	Pequeño	0,5	61,26
2	Pequeño	0,2	50,80
3	Grande	0,2	53,35
4	Grande	0,5	63,14
5	Pequeño	0,2	52,25
6	Grande	0,5	62,24
7	Pequeño	0,5	60,74
8	Grande	0,2	51,75
9	Grande	0,2	52,95
10	Grande	0,5	61,01
11	Pequeño	0,5	60,54
12	Pequeño	0,2	49,8

Como se evidencia en la tabla 4, el algoritmo *Proactive Forest* obtiene sus mejores resultados cuando se toma un valor de porcentaje de prueba de 0,5. El mejor resultado fue un 63,14 % de instancias correctamente clasificadas, aplicándose sobre un escenario grande, mientras que se observa una disminución de las instancias correctamente clasificadas cuando se emplea un valor de porcentaje de prueba de 0,2. Se llegó a obtener un mínimo de 49,8 % de instancias correctamente clasificadas, al aplicar el algoritmo en un escenario pequeño.

Con el fin de comprobar si lo antes planteado es cierto, se deben ejecutar las pruebas estadísticas correspondientes, identificando si los factores e interacciones influyen significativamente en el rendimiento del algoritmo. Para ello se construye un diagrama de Pareto (figura 2), donde se demuestra que no todos los factores e interacciones influyen significativamente en el rendimiento del algoritmo, a pesar de que individualmente los factores escenario y el porcentaje de prueba sí influyen en el rendimiento, pues sobrepasan el valor de significancia.

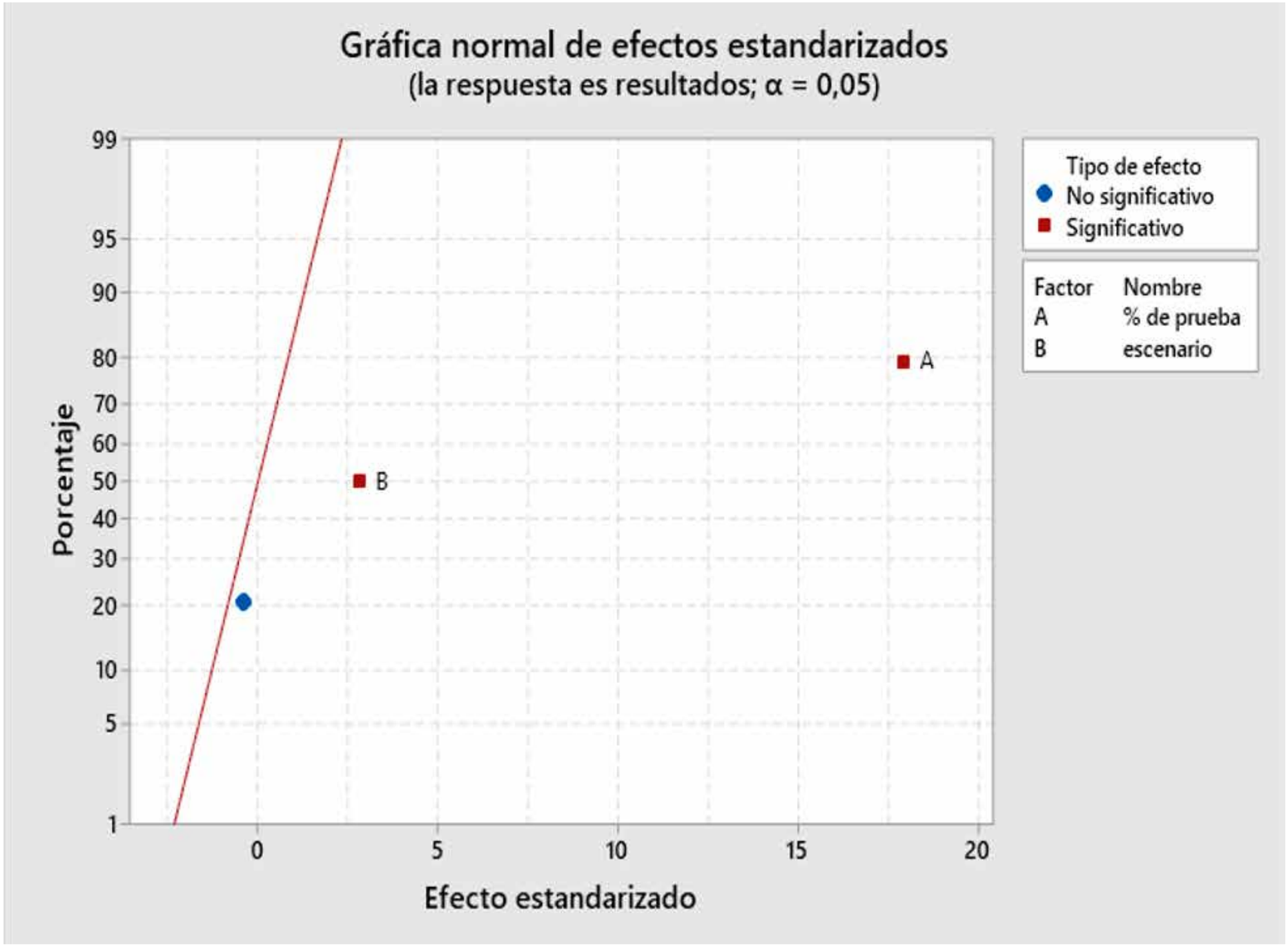

Fig. 3 Gráfica normal de efectos.

Como la combinación de los factores identificados no es estadísticamente influyente en el rendimiento del algoritmo, no se puede definir qué combinación de sus niveles inciden en los resultados. A pesar de esto, se comprueba que individualmente los dos factores identificados sí influyen de forma significativa en el rendimiento del algoritmo Proactive Forest, en la clasificación de un usuario en *bot* maligno o humano.

Comparación con Random Forest

Como parte de la experimentación realizada, se comparan los resultados de Proactive Forest (PF), con los de Random Forest (RF), empleado para la detección de *bots* malignos en Pardo *et al.* (2022), pero se utilizará el preprocesamiento de datos descrito aquí. La tabla 5 muestra los resultados logrados para cada algoritmo, particionados en igual combinación de niveles de los factores, para que su análisis sea más comprensible.

Tabla 5. Resultados de la experimentación de los algoritmos

Ejecución	Escenario	Porcentaje de prueba	Proactive Forest (PF)	Random Forest (RF)
1	Pequeño	0,5	61,26	62,32
2	Pequeño	0,5	60,74	61,12
3	Pequeño	0,5	60,54	62,34
4	Pequeño	0,2	50,80	57,21
5	Pequeño	0,2	52,25	55,01
6	Pequeño	0,2	49,8	56,85
7	Grande	0,5	63,14	61,92
8	Grande	0,5	62,24	61,44
9	Grande	0,5	61,01	62,10
10	Grande	0,2	53,35	55,65
11	Grande	0,2	51,75	55,55
12	Grande	0,2	52,95	55,70

Al analizar la tabla 5 se llega puede concluir que:

- Para escenarios pequeños y grandes, con valor de 0,5 en porcentaje de prueba, se aprecian resultados similares.
- Para escenarios pequeños y grandes, con valor de 0,2 en porcentaje de prueba, se aprecia una disminución de los resultados. La diferencia para ambos algoritmos es más distante, sobre todo en escenarios pequeños.

Para comprobar si lo anteriormente planteado es cierto, se aplican las pruebas estadísticas correspondientes. Se efectúan treinta ejecuciones de cada algoritmo en un escenario pequeño con un valor de porcentaje de prueba de 0,2, pues fue donde más diferencia se evidenció entre los resultados. La tabla 6 resume los valores máximos, mínimos y promedio de las ejecuciones realizadas.

Tabla 6. Resultados de máximo, mínimo y promedio para las treinta ejecuciones

Algoritmos	Mayor	Menor	Promedio
PF	53 750	48 800	51 130
RF	57 200	54 350	55 943

Para determinar si las muestras de los resultados para las treinta ejecuciones de cada algoritmo siguen una distribución normal, se realiza la prueba de normalidad de Shapiro-Wik y se planean las siguientes hipótesis:

H_0: La muestra sigue una distribución normal
H_1: La muestra no sigue una distribución normal

Como resultado de esta prueba se obtuvo un valor *p-value* > 0,1, superior al nivel de significa $\alpha = 0,05$, para los resultados del algoritmo Proactive Forest, mientras que para Random Forest se obtuvo un valor de *p-value* = 0,088, también superior a $\alpha = 0,05$, por lo que no existe

evidencia suficiente que rechace la hipótesis nula y se puede asumir que las muestras de resultados de ambos algoritmos siguen una distribución normal. Al asumir esto se efectúa una prueba paramétrica. En este caso se utiliza la prueba *t-student* para comparar el valor de la media de las dos muestras, determinando si existen diferencias significativas entre ellas. Si existen diferencias, se determina cuál de los dos algoritmos presenta mejor rendimiento. Las hipótesis planteadas son:

H_0: $\mu_{PF} - \mu_{RF} = 0$
H_1: $\mu_{PF} - \mu_{RF} < 0$

Al realizar la prueba se obtiene como resultado un valor de *p-value* = 0,00, menor al nivel de significancia $\alpha = 0,05$. Por lo que se rechaza la hipótesis nula y se puede afirmar que existen diferencias significativas entre los resultados de cada algoritmo. Donde, como muestra la figura 4 con un diagrama de valores individuales de cada algoritmo, el algoritmo *Random Forest* presenta mejores resultados que *Proactive Forest.*

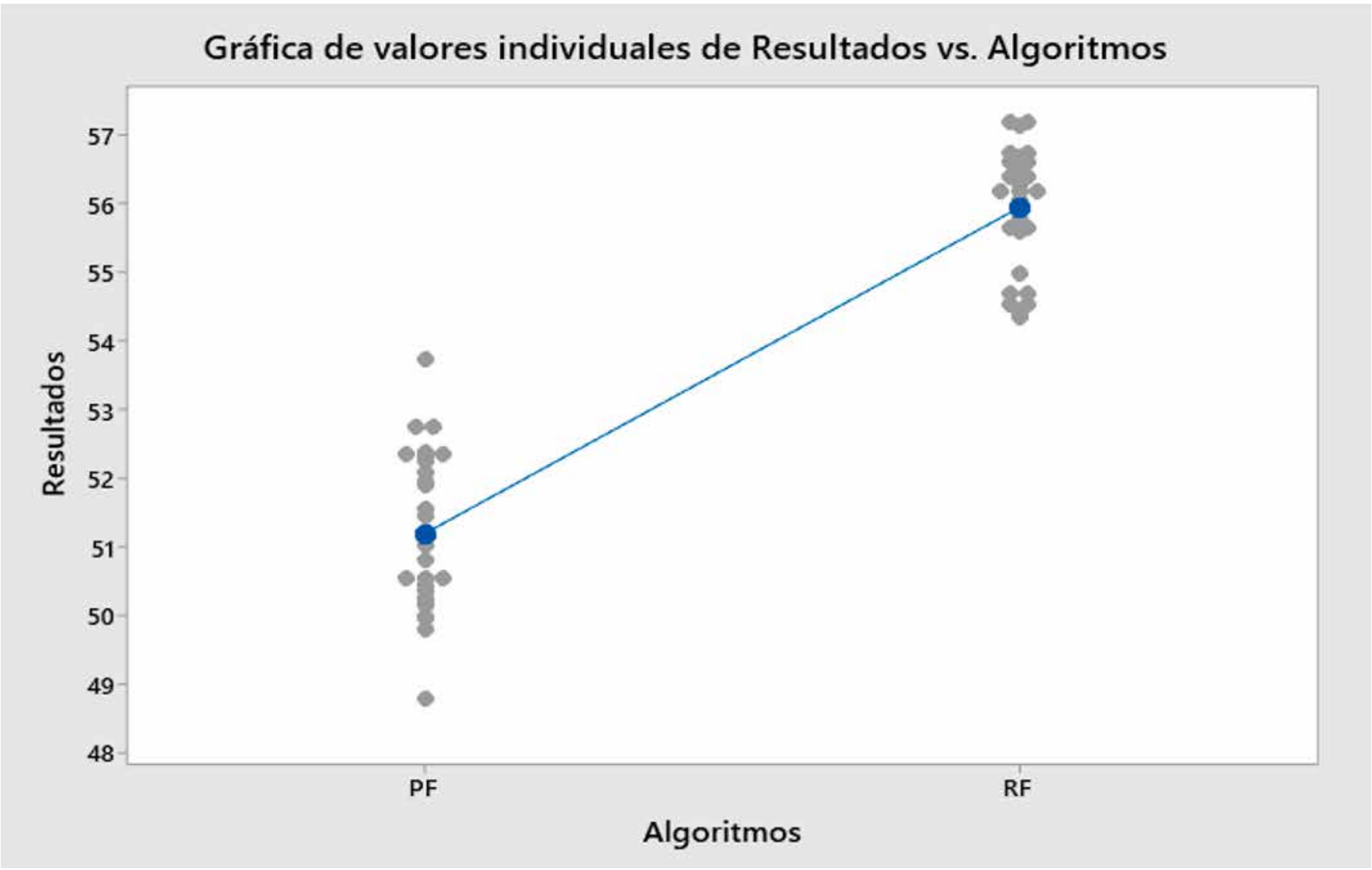

Fig. 4 Gráfica de valores individuales.

Adicionalmente, sin considerar una suposición de que la muestra sea normal, se realiza la prueba no paramétrica Man-Whitney, la cual permite conocer si hay diferencias significativas en cuanto al valor de la mediana de los resultados de ambos algoritmos. En caso de que existan diferencias significativas, se obtendrá cuál de los dos algoritmos presenta mejor rendimiento en base a la muestra de resultados analizada. Las hipótesis planteadas son:

H_0: $\eta_{PF} - \eta_{RF} = 0$
H_1: $\eta_{PF} - \eta_{RF} < 0$

Como resultado de la prueba se obtuvo un valor *p-value* = 0,000, menor al nivel de significancia $\alpha = 0,05$, por lo que existe evidencia suficiente para rechazar la hipótesis nula. Por tanto, se reafirma que existe una diferencia significativa entre los resultados de ambos algoritmos cuando se emplean en la detección de *bots*, siendo superior el rendimiento mostrado en Random Forest.

CONCLUSIONES

Al culminar este trabajo se puede concluir que:

1. El algoritmo de Proactive Forest, que surge como una vía de solución a las desventajas planteadas por el algoritmo Random Forest, ya empleado en la detección de *bots*, puede ser aplicable en la detección de estos programas maliciosos, a partir de la clasificación de un usuario en humano o *bot* maligno. Para ello se emplea una base de datos que contiene datos de *bots* malignos tomados en tiempo real, conocida como CTU-13.
2. Para los factores identificados en el diseño experimental se evidenció que el algoritmo Proactive Forest presentaba su mejor rendimiento en escenarios grandes y utilizando 50 % de los datos para la prueba y entrenamiento, donde alcanza un máximo de 63,14 % de instancias correctamente clasificadas. A pesar de ello se pudo confirmar que estos factores no tienen una influencia directa en el rendimiento del algoritmo, en base al porcentaje de instancias correctamente clasificadas como *bots* malignos o humanos.
3. Al realizar una comparación del algoritmo Proactive Forest, contra el algoritmo Random Forest, se obtiene como resultado que este último presenta mejor rendimiento cuando es aplicado en la detección de *bots* malignos, al clasificar un usuario en *bot* maligno o humano.
4. Se propone para futuros trabajos:

- Evaluar el comportamiento del algoritmo Proactive Forest sobre otras bases de datos ligadas a la detección de *bots*, además de comparar los resultados alcanzados con otros algoritmos reportados en el estado del arte.
- Analizar el rendimiento del algoritmo empleando otras métricas como el caso de *f1* o *recall.* Esta última es de gran importancia para detectar que tan bien se enfoca el algoritmo en clasificar correctamente usuarios *bots* malignos, además de realizar un análisis profundo de los resultados utilizando la matriz de confusión.
- Analizar otros factores influyentes en el rendimiento del algoritmo; pero más ligado a los hiperparámetros, como es el caso de cantidad de árboles del bosque, criterios de parada, criterios de selección de variables, entre otros.

- Emplear la técnica de validación cruzada para evaluar el rendimiento del algoritmo, a partir de diversos conjuntos de prueba y entrenamiento.

5. Al término de este trabajo se tienen las siguientes limitaciones:

- En la fase de preprocesamiento se emplea la técnica de SMOTE para aumentar la cantidad de ejemplos de la clase minoritaria de *bots* malignos. Esta técnica trae consigo la creación de datos sintéticos, que fueron considerados en la validación del algoritmo. En la sección que describe este proceso se explican las razones de la decisión, aunque es válido destacar que la validación podría realizarse con los datos reales antes de efectuar el balanceo de clases y evitar así la introducción de ejemplos ficticios a este conjunto, por lo que es un elemento importante que se debe tener en cuenta para trabajos futuros.
- Al aplicar la técnica de PCA solo se obtuvieron los componentes con mayor porcentaje de varianza acumulativa explicada y se hizo un análisis del comportamiento de la reducción de dimensionalidad de cada escenario, pero sin estudiar a este para declarar un número fijo de componentes principales, ya que se seleccionaba un tipo de escenario según los factores de experimentación, por lo que se debería aplicar la reducción de dimensionalidad de los escenarios de CTU-13 en los conjuntos de entrenamiento; se obtuvieron los parámetros de proyección, para aplicarlos en el conjunto de prueba.

REFERENCIAS

Cepero-Pérez, N., Denis-Miranda, L. A., Hernández-Palacio, R., Moreno-Espino, M., y García-Borroto, M. (2018). Proactive Forest for Supervised Classification. *International Workshop on Artificial Intelligence and Pattern Recognition,* pp. 255–262.

Dahan, H., Cohen, S., Rokach, L., y Maimon, O. (2014). Proactive Data Mining with Decision Trees. *Proactive Data Mining with Decision Trees,* pp. 21-33.

Doran, D. (2011). Web robot detection techniques: Overview and limitations. *Data Mining and Knowledge Discovery, 22*(1), 183-210.

Fernández, C., Baptista, P., y Hernández, R. (1998). *Metodología de la investigación* (T. M.-H. C. Inc. Ed. Vol. Segunda Edición). México.

Haq, S., y Singh, Y. (2018). *Botnet detection using machine learning.* Paper presented at the In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC).

Hernández, J., Ramírez, J., y Ferri, C. (2004). *Introducción a la Minería de Datos* (Vol. 9). Madrid.

Imperva. (2020). *Bad Bot Report.* Retrieved from California, USA

Mohammed, M., Khan , M. B., y Mohammed Bashier, E. B. (2016). *Machine Learning Algorithms and Applications:* Crc Press.

Pardo, D., Moreno, M., Diaz, H., y Chissingui, H. J. (2022). *RANDOM FOREST PARA LA DETECCIÓN DE Bots EN EL COMERCIO ELECTRÓNICO.* Paper presented at the X Congreso Internacional de Tecnologías, Comercio Electrónico y Contenidos Digitales.

Rokach, L. (2015). Decision forest: Twenty years of research. *Information Fusion, 27,* 111-125.

Rout, Lingam, R. R., y Somayajulu, D. V. (2020). Detection of malicious social *bots* using learning automata with url features in twitter network. *IEEE Transactions on Computational Social Systems, 7*(4), 1004-1018.

Rovetta, S., Suchacka, G., y Masulli, F. (2020). *Bot* recognition in a Web store: An approach based on unsupervised learning. *Journal of Network and Computer Applications., 157,* 102577.

Velasco, J., González, V., Fidalgo, E., y Alegre, E. (2021). *Efficient Detection of Botnet Traffic by features selection and Decision Trees.* Paper presented at the Preprint submitted to IEEE Access.

Vishwakarma, A. R. (2020). *Network Traffic Based Botnet Detection Using Machine Learning.* (Master of Science (MS)), San Jose State University, SJSU Scholar Works.

Xu , H., Li , Z., Chu, C., Chen, Y., Yang , Y., Lu, H., . . . Stavrou, A. (2019). Detecting and Characterizing Web *Bot* Traffic in a Large E-commerce Marketplace. *European Symposium on Research in Computer Security,* pp. 143-163.