**Tipo de contribución:** Artículo Original

**Idioma:** Español

**MULTICLASIFICADOR HOMOGÉNEO**

**PARA DETECCIÓN DE BOTS EN EL COMERCIO ELECTRÓNICO**

***HOMOGENEOUS MULTICLASSIFIER FOR BOT DETECTION IN E-COMMERCE***

*.*

**RESUMEN**

Para el comercio electrónico, mitigar las amenazas de bots es una tarea relevante, debido al enorme impacto de las actividades maliciosas perpetradas por bots, a través de estos por personas mal intencionadas, las cuales, además del daño que causan a la infraestructura informática y pérdidas económicas, también agudizan la insatisfacción de los usuarios humanos. Actualmente, este problema se hace aún más complejo, porque en ocasiones, usuarios humanos emplean aplicaciones móviles con sus cuentas de usuario para tener privilegios en el acceso a determinados servicios del comercio, o sea, es cada vez más elevado el nivel de sofisticación de bots, lo que resulta que en determinadas circunstancias los patrones de actividades humanas tengan las mismas características de las actividades de bots. Con estos niveles de desarrollo, las tareas de detección se hacen cada vez más complejas y vitales. En este estudio, se propone un enfoque de detección basado en aprendizaje supervisado, con los modelos homogéneos de ensambles de clasificadores, *Bagging* y *Boosting*. Los modelos construidos con base a los estimadores ExtraTree, Cart y K-vecinos más cercanos, lograron la puntuación F1 máxima de 100 %, en determinados escenarios, en que la cantidad de ejemplos de la clase minoritaria no sobrepasa 9 % del conjunto de datos. Los resultados se comparan con otros enfoques del estado del arte.

**Palabras clave:** detección de bots, metaaprendizaje, multiclasificadores, comercio electrónico.

***ABSTRACT***

*For electronic commerce, mitigating bot threats is a relevant task, due to the enormous impact of malicious activities perpetrated by bots, through these by malicious people, whose, in addition to the damage they cause to the IT infrastructure and economic losses, also exacerbate human user dissatisfaction. Currently this problem becomes even more complex, because sometimes human users use mobile applications with their user accounts to have access privileges to certain business services, that is, the level of sophistication of the bots is increasingly higher, which results in the patterns of human activities under certain circumstances having the same characteristics as the activities of bots. With these levels of development, detection tasks become increasingly complex and vital. In this study, a detection approach based on supervised learning is proposed, with the homogeneous models of ensembles of classifiers, Bagging and Boosting. The models built based on the ExtraTree, Cart and K-nearest neighbors estimators, achieved the maximum F1 score of 100%, in certain scenarios, in which the number of examples of the minority class does not exceed 9% of the data set. The results are compared with other approaches of the state of the art.*

***Keywords:*** *bot detection, meta learning, multiclassifiers, e-commerce.*

**INTRODUCCIÓN**

La pandemia de la COVID-19 impulsó el desarrollo de servicios basados en la web. Internet Banking, teletrabajo, e-learning, comercio electrónico, marketing digital, gobierno, entre otros, son áreas con alto nivel de informatización en el mundo de hoy, lo que también es visible en países como Cuba, donde se busca profundizar el proceso de informatización como una de las claves de las estrategias de desarrollo sostenible (Bermúdez, 2022).

Los bots son caracterizados por poseer determinadas habilidades de usuarios humanos e incluso mejoran algunas en determinados contextos. De forma general, los bots pueden pertenecer a una comunidad o conjunto de bots denominado *botnet,* por otro lado, pueden ser controlados de forma remota por el *botmaster* (rol de un usuario humano), mediante canales de comando y control (C&C, del inglés, *Command and Control)* dedicados. Estos últimos basados fundamentalmente en los protocolos de comunicación: *Internet Relay Chat (IRC)*, *Peer-to-Peer (P2P)* y el protocolo de transferencia de hipertexto (HTTP, del inglés, *Hypertext Transfer Protocol).*

Pueden ser clasificados según la finalidad como bots benignos, los que realizan tareas como motores de búsquedas y otras, que son parte importante de los sistemas, y bots maliciosos, los que son utilizados por atacantes para perpetrar actividades maliciosas. Según su evolución, pueden ser clasificados los bots maliciosos en: bots simples, se conectan en una sola dirección IP asignada por el proveedor del servicio de Internet (ISP, del inglés, *Internet Service Provider),* usando *scripts* automatizados para que no sean identificados, además no se identifican como un navegador. Los bots moderados, con determinada complejidad, simula la tecnología del navegador, incluida la capacidad de ejecutar JavaScript. Por último, los bots avanzados, reproducen movimientos del *mouse* y *clics* que engañan incluso a los métodos de detección más sofisticados, imitan a los humanos, emplean comportamientos más evasivos, utilizan *software* de automatización del navegador o *malware* instalado en navegadores reales para conectarse a los sitios. Bots evasivos son una agrupación de bots maliciosos moderados y avanzados, que tienden a recorrer direcciones IP aleatorias, ingresan a través de *proxies* anónimos y redes P2P, y pueden cambiar sus agentes de usuario; utilizan una combinación de tecnologías y métodos para evadir la detección, mientras se mantiene la persistencia en los sitios objetivo; a menudo eligen tácticas «bajas y lentas», que les permitir llevar a cabo ataques significativos utilizando menos solicitudes e incluso retrasar las solicitudes, lo que les posibilita no sobresalir de los patrones de tráfico normales y evitar la activación de umbrales de detección de seguridad basados en tasas. Este método reduce el «ruido» o grandes picos de tráfico generados por muchas compañías de bots maliciosos (Imperva, 2022).

Según el *Imperva bad bot report* 2022 (Imperva, 2022), 42*,*3 % del tráfico de Internet del año 2021 es resultante de la actividad de bots, donde 27,7 % es tráfico de bots maliciosos, lo cual representa 2,1 % más que el año anterior. Estos resultados demuestran la tendencia creciente en el empleo de bots, lo cual trae consigo un crecimiento de la exposición de recursos informáticos a varias amenazas.

Una de las grandes ventajas de la web consiste en el poder de interconectividad global, lo que proporciona un aporte importante para los sistemas en dependencia de sus finalidades, y es determinante para la disponibilidad y calidad de servicio (QoS, del inglés, *Quality of Service).* Sin embargo, esta interconectividad suele ser el recurso que expone el sistema a variadas amenazas informáticas, de hecho, es el más utilizado en las actividades que violan las propiedades de seguridad de los sistemas (confidencialidad, autenticación, no repudio, control de integridad, auditabilidad e incluso la disponibilidad) de los sistemas, perpetradas por individuos que intentan dañar los recursos, confundir la opinión pública, obtener ventajas sobre los adversarios y muchas otras actividades maliciosas materializadas por medio de ataques cibernéticos.

El comercio electrónico es una de las áreas que en los últimos años se ha beneficiado de los diversos avances tecnológicos, no solo por el tema de la pandemia de la COVID-19, pues su tendencia de expansión es perceptible desde hace años con el crecimiento de la adopción de servicios basados en Internet. Posee características distinguibles, donde se pueden identificar para el lado del proveedor la rentabilidad, el gran alcance de las ventas, los ahorros de recursos para el alcance del servicio, etc., lo que convierte al comercio electrónico en un entorno de alta disputa entre adversarios. También es rehén de políticas microeconómicas y macroeconómicas de Estados, continentes, grupos, etc., e incluso de los propios clientes, lo que desencadena un conjunto de actividades maliciosas con recurso a bots y que violan las propiedades de seguridad mencionadas en el párrafo anterior.

En el estado del arte se emplean con mayor frecuencia algoritmos supervisados para clasificar el tráfico de red o las solicitudes de usuarios. Entre ellos, los algoritmos de aprendizaje supervisados tradicionales tienen como objetivo minimizar el número de errores cometidos durante la clasificación, asumiendo que los costos de los falsos positivos y falsos negativos son iguales, considerando un equilibrio de clases, al suponer una distribución equilibrada de clases y costos de error iguales, por lo tanto, no son adecuados para datos de clases desequilibradas. Los modelos multiclasificadores funcionan relativamente bien para el problema de desequilibrio de clases en dos clases, ya que los clasificadores individuales que componen el conjunto pueden incluir versiones de los enfoques de sobremuestreo y el desplazamiento del umbral (Han *et al.,* 2012).

En este trabajo se implementan los algoritmos multiclasificadores homogéneos *Bagging* y *Adaboost* para detectar Bots, con estimadores seleccionados según la frecuencia de empleo en el estado de arte, con el objetivo principal de seleccionar las configuraciones de multiclasificadores homogéneos con mejor rendimiento empleando la base de datos CTU-13.

**ANTECEDENTES**

Aquí se presenta un resumen de los trabajos relacionados con la detección de bots en el comercio electrónico. Se hace énfasis en las diferentes variantes de algoritmos empleados, además se presentarán los principios fundamentales para la construcción de multiclasificadores y la descripción de las técnicas más comunes.

**Trabajos relacionados**

Entre las diferentes variantes de enfoques de detección de bots, el aprendizaje automático es el más común, con las variantes de algoritmos: no supervisados y supervisados, siendo estos últimos los más frecuentes. La adopción del tipo de algoritmo está directamente asociada a las características propias del aprendizaje supervisado con respecto al problema de detección de bots.

Entre los algoritmos de aprendizaje supervisado para la detección de bots el más utilizado es Random Fore (Barbon *et al.,* 2018; García *et al.,* 2020). Adicionalmente, existen otros enfoques en los estudios de detección de bots en el comercio electrónico:

* En Suchacka & Sobków (2015) se emplea un enfoque bayesiano de análisis de clústeres con *accuracy* mayor de 90 %.
* En Rahman & Tomar (2020) se implementaron algoritmos de Red Neuronales Artificiales (ANN, del inglés, *Artificial Neural Networks)* para la detección de *scraping Bots* en sitios web de comercio electrónico, con un rendimiento (valores de las métricas de rendimiento de clasificación como, *accuracy*, *precision*, *f-score*, *recall*) no superior a 95 %.
* En Cabri *et al.* (2018) y Suchacka *et al.* (2021) se emplean otras variantes de ANN.
* En Rovetta *et al.* (2020) se emplea el algoritmo K-Medios, que incluso su rendimiento sobrepasó los clasificadores basados en Perceptrón Multicapas (MLP, del inglés *Multi-Layer Perceptrón)* y de Máquinas de Soporte Vectorial (SVM, del inglés *Support Vector Machine)*.
* En Suchacka & Iwanski (2020) se emplea el algoritmo no supervisado de Cuello de Botella de Información Aglomerativa (AIB, del inglés *Agglomerative Information Bottleneck).*
* En B. *et al.* (2018) se emplean arboles de decisión (DT, del inglés *Decision Tree)* y un enfoque basado en reglas para clasificar las direcciones IP de bots maliciosos y benignos.

Los estudios mencionados en el párrafo anterior tienen la similitud de que los modelos de aprendizaje automático fueron diseñados con características de sesiones de usuarios (extraídas a nivel de sesión y aplicación), donde se incluyen las características semánticas que representan los contenidos de las solicitudes HTTP en cada sesión. Además, hay otros estudios donde los modelos de clasificación para detectar bots se construyen con características de tráfico web extraídas a nivel de las capas de red y transporte del modelo OSI.

**Construcción de conjuntos de clasificadores**

En Dietterich (2000) se resumen las razones para el diseño de multiclasificadores:

* Estadística: si los datos de entrenamiento son pequeños en comparación con el espacio de hipótesis. Si los datos son insuficientes, el algoritmo de aprendizaje puede encontrar variadas hipótesis en H con igual precisión sobre los datos. Si los clasificadores se combinan el algoritmo puede promediar sus votos y reducir el riesgo de seleccionar una hipótesis errónea.
* Computacional: en entornos donde no hay problema estadístico, como muchos de los algoritmos funcionan por medio de búsquedas locales, esto tiene como consecuencia que dichos algoritmos pueden cometer errores en los máximos locales, lo que puede representar una enorme dificultad para llegar al objetivo desde la perspectiva computacional. Pero una combinación de clasificadores cuyas búsquedas locales sean realizadas desde diferentes puntos de partida del espacio de hipótesis, puede llegar a una mejor aproximación que uno de los clasificadores individuales que componen el conjunto.
* Representación: en la mayor parte de las aplicaciones de aprendizaje automático, su función real *f* no puede ser representada por alguna hipótesis de H, la combinación de clasificadores puede permitir la expansión del espacio H de tal modo que se pueda incluir f en dicho espacio. Se hace referencia también a algoritmos de ANN y árboles de decisión (DT, del inglés *Decision Trees),* que son muy flexibles en la exploración de todas las hipótesis posibles de determinado espacio.

Estas razones están centradas en el hecho de que un conjunto de clasificadores puede representar mayor seguridad de clasificación, o sea, puede ser más preciso que un clasificador individual en un determinado entorno. Al considerar varios clasificadores (del mismo tipo, como de tipos diferentes), le confiere más diversidad (por usar diferentes formas de exploración del campo de hipótesis), en dependencia de la función de decisión del multiclasificador se torna el aporte determinante en la decisión. Es normal que en determinado entorno un clasificador individual tenga mejor rendimiento que un multiclasificador; pero teniendo en cuenta la capacidad de generalización de los modelos para clasificación, un modelo multiclasificador tendrá una respuesta más robusta que un clasificador individual.

Existen muchos métodos para generar el conjunto de clasificadores, teniendo en cuenta lo planteado en Dietterich (2000), a continuación se describen algunos métodos que son de propósitos generales:

* Voto Bayesiano: aplicable en contextos donde se pueden enumerar las hipótesis hl y calcular probabilidad *a posteriori.* Se centra fundamentalmente en el teorema de Bayes.
* Manipulación de los ejemplos de entrenamiento: con la finalidad de generar diferentes hipótesis, el algoritmo puede ser ejecutado varias veces con un conjunto de datos diferentes. Esta técnica funciona mejor con algoritmos de aprendizaje inestables, es decir, aquellos cuyo modelo resultante puede variar mucho al cambiar en menor grado los ejemplos de entrenamiento. Por ejemplo, los árboles de decisión, las redes de neuronales artificiales y los algoritmos de inducción de reglas, son algoritmos inestables. En cambio, los métodos de regresión lineal y el vecino más cercano, suelen ser muy estables.
* Manipulación de los atributos de entrada: se centra en la manipulación de los atributos de entrada disponibles a la hora de utilizar el algoritmo de aprendizaje. Una debilidad de esta técnica es que solo funciona cuando los atributos de entrada son altamente redundantes.
* Manipulación de las salidas: en esta técnica el conjunto de clasificador es generado por medio de la manipulación de clase de instancia, como es el caso de la técnica ECOC *(error correcting output code).*
* Introducción de aleatoriedad: otra técnica muy común para la generación de conjuntos consiste en la incorporación de aleatoriedad dentro del algoritmo de aprendizaje.
* En el estudio Dietterich (2000) no se hace referencia a los conjuntos de clasificadores formados por diferentes algoritmos de aprendizaje, centrándose únicamente en multiclasificadores homogéneos. En el estado del arte no hay consenso en cuanto a la taxonomía de algoritmos de construcción de multiclasificadores, pero de forma genérica se destacan los métodos de *Boosting, Bagging, Voting* y *Stacking*. Los dos primeros se refieren a algoritmos homogéneos que se basan en la manipulación de ejemplos de entrenamiento, lo cual constituye el objetivo de este estudio.
* *Bagging* es un método que logra la diversidad en función de la aleatoriedad que se introduce en la fase de entrenamiento del modelo, por ejemplo, el muestreo aleatorio con reemplazo es utilizado para generar los subconjuntos de datos de entrenamiento de los clasificadores bases, a su vez las predicciones de los clasificadores bases son combinadas con una función basada en votación o promedio. Del mismo modo, el método *Boostin*g logra la diversidad en el entrenamiento; pero con una estrategia diferente: los modelos clasificadores bases son formados secuencialmente, donde el objetivo del clasificador base posterior seria corregir los errores de clasificación del modelo anterior, o sea, se introducen en su entrenamiento las instancias clasificadas de forma errónea por el clasificador base anterior al actual.

**METODOLOGÍA**

Este trabajo fue desarrollado, empleando un enfoque de Minería de Datos cuyos pasos se describen en los apartados a continuación (tabla 1).

Tabla 1, Características de los escenarios de botnet de CTU-13 (CF: Clics fraudulentos, PS: Escaneo de puertos, FF: FastFlux, US: Compilado y controlado por los desarrolladores) (Fuente: García *et al.,* 2014)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **IRC** | **SPAM** | **CF** | **PS** | **DDoS** | **FF** | **P2P** | **US** | **HTTP** | **Nota** |
| 1 | √ | √ | √ |  |  |  |  |  |  |  |
| 2 | √ | √ | √ |  |  |  |  |  |  |  |
| 3 | √ |  |  | √ |  |  |  | √ |  |  |
| 4 | √ |  |  |  | √ |  |  | √ |  | UDP and ICMP DDoS |
| 5 |  | √ |  | √ |  |  |  |  | √ | Scan web proxies |
| 6 |  |  |  | √ |  |  |  |  |  | Proprietary C&C, RDP |
| 7 |  |  |  |  |  |  |  |  | √ | Chinese hosts |
| 8 |  |  |  | √ |  |  |  |  |  | Proprietary C&C, Net-BIOS, STUN |
| 9 | √ | √ | √ | √ |  |  |  |  |  |  |
| 10 | √ |  |  |  | √ |  |  | √ |  | UDP DDoS |
| 11 | √ |  |  |  | √ |  |  | √ |  | ICMP DDoS |
| 12 |  |  |  |  |  |  | √ |  |  | Synchronization |
| 13 |  | √ |  | √ |  |  |  |  | √ | Captcha, Web mail |

**Descripción de los datos**

Los datos son el componente fundamental para la construcción de modelos basados en aprendizaje automático. Existe bastante escasez de conjuntos de datos de comercio electrónico disponibles en la web, de hecho, la mayor parte de los algoritmos de detección de bots en comercio electrónico del estado del arte son implementados con conjuntos de datos formados a partir de archivos de registro de servidores web, los cuales no están disponibles por diversas razones. Por otro lado, las plataformas de comercio electrónico son desarrolladas con tecnologías diferentes, lo que reduce la posibilidad de tener un conjunto de datos que sea genérico.

CTU-13 es un conjunto de datos de tráfico real de botnet, capturado en la universidad CTU de la República Checa en 2014. Consiste en trece capturas o escenarios de diferentes muestras de tráfico de botnets y contiene datos de siete diferentes botnets: Neris, Rbot, Virut, Murlo, NSIS, Donbot y Sogou. Cada escenario fue analizado y etiquetado manualmente; la columna ID de la tabla 1 corresponde a los escenarios del conjunto de datos, cuyas características se describen en esa tabla.

**Preprocesamiento**

Para adecuar los datos al proceso de minería, se aplicó un conjunto de procedimientos que buscan eliminar irregularidades, inconsistencias y ruidos en los datos, convertir, escalar y reducir la dimensionalidad. El conjunto de datos posee instancias de clases de tráfico de *botnet,* *normal* y *background*. Las clases de *normal* y *background* fueron tratadas como una sola clase, porque la primera representa el tráfico de usuarios humanos y la segunda es el tráfico propio de la red cuando no existen actividades de usuarios humanos y de bots. Tras la carga y limpieza de los datos, y con el objetivo de garantizar que los algoritmos de clasificación no sean inducidos a atribuir más peso a un atributo en relación con los otros, con una técnica de estandarización se estableció una misma escala de valores para cada atributo. Para la reducción de dimensionalidad se implementó el Análisis de Componentes Principales (PCA, del inglés *Principal Component Analysis),* considerando una varianza explicativa acumulada correspondiente a 98 %, que resultó en 10-12 componentes principales en dependencia del escenario. Se observa que los escenarios de CTU-13 son muy desbalanceados, donde la clase positiva (clase de tráfico de botnet) es la clase minoritaria, con un porcentaje máximo inferior a 10 % de los ejemplos en un escenario, como se describe en la tabla 2.

Tabla 2. Dimensiones de los escenarios de CTU-13 antes y después del preprocesamiento (PC: componentes principales)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Número de instancias por clase antes del preprocesamiento**  **(con 14 atributos)** | | | **Fases del preprocesamiento** | | **Porcentaje de**  **la clase minoritaria** |
| **Limpieza e integración** | **Transformación y selección** |
| ID | Botnet | Normal | Total | Instancias | PC |
| 1 | 40 961 | 2 783 675 | 2 824 636 | 2 824 636 | 11 | 1,45 % |
| 2 | 20 941 | 1 787 181 | 1 808 122 | 1 808 122 | 11 | 1,15 % |
| 3 | 26 822 | 4 683 816 | 4 710 638 | 4 710 638 | 11 | 0,57 % |
| 4 | 2 580 | 1 118 496 | 1 121 076 | 1 121 076 | 11 | 0,23 % |
| 5 | 901 | 128 931 | 129 832 | 129 832 | 11 | 0,70 % |
| 6 | 4 630 | 554 289 | 558 919 | 558 919 | 11 | 0,83 % |
| 7 | 63 | 114 014 | 114 077 | 114 077 | 11 | 0,05 % |
| 8 | 6 127 | 2 948 103 | 2 954 230 | 2 954 230 | 11 | 0,20 % |
| 9 | 184 987 | 1 902 521 | 2 087 508 | 2 087 508 | 12 | 8,45 % |
| 10 | 106 352 | 1 203 439 | 1 309 791 | 1 309 791 | 10 | 8,11 % |
| 11 | 8 164 | 99 087 | 107 251 | 107 251 | 10 | 7,61 % |
| 12 | 2 168 | 323 303 | 325 471 | 325 471 | 11 | 0,67 % |
| 13 | 40 003 | 1 885 146 | 1 925 149 | 1 925 149 | 11 | 2,07 % |

**Diseño de experimento**

Tras el conjunto de tareas a las que fueron sometidos los datos en la etapa de preprocesamiento, para la validación de los algoritmos de multiclasificadores homogéneos, se realiza un diseño de experimento basado en determinar los modelos multiclasificadores adecuados para clasificar el tráfico de usuarios de humanos y de Bots con base a métricas de rendimiento. El proceso consistió en las siguientes actividades:

* Construcción individual de algoritmos supervisados: en este experimento se busca encontrar un modelo clasificador individual de referencia (según los algoritmos supervisados de la tabla 3), para posteriormente compararlo con los modelos multiclasificadores Bagging y AdaBoost.
* Construcción de ensambles: se construyen los ensambles Bagging y AdaBoosting, considerando como estimadores los algoritmos supervisados de la tabla 3 y se busca encontrar el modelo ensamble más adecuado para clasificar el tráfico normal e de botnet.
* Comparación con el estado del arte: se comparan los resultados de los mejores modelos ensambles con los enfoques del estado del arte donde se emplea el mismo conjunto de datos.

Tabla 3. Algoritmos de aprendizaje supervisado seleccionados

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Variante** | **Identificador** |
| Modelos lineares | Logística | LR |
| Descenso de gradientes estocástico | SGD |
| Árbol de decisión | Cart | CART |
| ExtraTree | EXTRA |
| Red bayesiana | Naive Bayes | NAIVE |
| Basado en instancias | k-vecinos más cercanos | KNN |
| Máquinas de soporte vectorial | LinearSVC | LSVC |
| Red neuronal artificial | Perceptrón multicapas | MLP |

**RESULTADOS Y DISCUSIÓN**

Antes del empleo de los ensambles se evaluaron de forma individual los algoritmos de aprendizaje supervisado mencionados en la tabla 3. Para la evaluación del rendimiento de los algoritmos se empleó la métrica puntuación F1,que permite tener una noción más precisa de la influencia de los falsos positivos y falsos negativos en escenarios desbalanceados. Con la referida métrica se puede verificar que los valores F1 = 0,00 implican que el algoritmo no logró clasificar correctamente ningún ejemplo de la clase positiva (clase de tráfico de botnet).

Los resultados de la evaluación individual de los algoritmos supervisados en la tabla 4 demuestran una tendencia de bajo rendimiento en los escenarios con un desbalance de clases severo, como los escenarios 6, 7, 8 y 12, mientras que los escenarios 10 y 11 fueron los más favorables, donde los todos los algoritmos lograron un rendimiento no inferior a 85 %. Los clasificadores basados en los algoritmos ExtraTree, CART y KNN con los promedios de rendimiento (usando el conjunto de test) de 91 %, 93 % y 91 % respectivamente, han presentado mejor estabilidad del rendimiento en los diferentes escenarios.

Tabla 4. Desempeño de los clasificadores individuales según la métrica puntuación F1, La columna ID corresponde a los escenarios de CTU-13 mencionados en la tabla 1, Las celdas con valores en negrita, corresponden a los menores rendimientos alcanzados por los algoritmos de aprendizaje en el respectivo escenario (*entr*: Entrenamiento, *test:* Test)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Clasificadores** | | | | | | | | | | | | | | | |
| **LR** | | **EXTRA** | | **SGD** | | **CART** | | **NAIVE** | | **KNN** | | **LSVC** | | **MLP** | |
| ID | entr | test | Entr | test | entr | test | entr | test | entr | test | entr | test | entr | test | entr | Test |
| 1 | 0,27 | 0,27 | **1,00** | **0,96** | 0,00 | 0,00 | **1,00** | **0,97** | 0,08 | 0,08 | **0,97** | **0,96** | 0,27 | 0,26 | 0,91 | 0,90 |
| 2 | 0,03 | 0,02 | **1,00** | 0,88 | 0,00 | 0,00 | **1,00** | 0,90 | 0,20 | 0,20 | 0,91 | 0,88 | 0,02 | 0,02 | 0,80 | 0,79 |
| 3 | 0,00 | 0,00 | **1,00** | **0,99** | 0,00 | 0,00 | **1,00** | **0,99** | 0,08 | 0,08 | **0,99** | **0,98** | 0,00 | 0,00 | 0,96 | 0,96 |
| 4 | 0,22 | 0,24 | **1,00** | **0,96** | 0,01 | 0,03 | **1,00** | **0,97** | 0,03 | 0,03 | **0,96** | **0,96** | 0,05 | 0,04 | 0,67 | 0,69 |
| 5 | 0,27 | 0,25 | **1,00** | 0,91 | 0,62 | 0,64 | **1,00** | **0,94** | 0,20 | 0,20 | **0,94** | **0,93** | 0,00 | 0,00 | 0,66 | 0,68 |
| 6 | 0,00 | 0,00 | **1,00** | **0,96** | 0,00 | 0,00 | **1,00** | **0,97** | 0,33 | 0,35 | **0,96** | **0,94** | 0,00 | 0,00 | 0,90 | 0,90 |
| 7 | 0,00 | 0,00 | **1,00** | 0,56 | 0,00 | 0,00 | **1,00** | 0,65 | 0,04 | 0,02 | 0,77 | 0,54 | 0,00 | 0,00 | 0,00 | 0,00 |
| 8 | 0,00 | 0,00 | **1,00** | **0,97** | 0,00 | 0,00 | **1,00** | **0,97** | 0,10 | 0,10 | **0,97** | **0,96** | 0,00 | 0,00 | **0,93** | **0,93** |
| 9 | 0,83 | 0,83 | **1,00** | **0,97** | 0,84 | 0,84 | **1,00** | **0,98** | 0,46 | 0,46 | **0,98** | **0,97** | 0,84 | 0,84 | **0,94** | **0,94** |
| 10 | **0,95** | **0,95** | **1,00** | **1,00** | **0,95** | **0,95** | **1,00** | **1,00** | **0,93** | **0,93** | **1,00** | **1,00** | **0,95** | **0,95** | **1,00** | **1,00** |
| 11 | **0,98** | **0,99** | **1,00** | **1,00** | **0,99** | **0,99** | **1,00** | **1,00** | 0,86 | 0,85 | **1,00** | **1,00** | **0,99** | **0,99** | **1,00** | **1,00** |
| 12 | 0,00 | 0,00 | **1,00** | 0,74 | 0,00 | 0,00 | **1,00** | 0,73 | 0,06 | 0,06 | 0,81 | 0,76 | 0,00 | 0,00 | 0,51 | 0,45 |
| 13 | 0,39 | 0,40 | **1,00** | **0,98** | 0,00 | 0,00 | **1,00** | **0,98** | 0,20 | 0,20 | **0,99** | **0,98** | 0,35 | 0,36 | **0,94** | **0,93** |
|  | Promedio del rendimiento | | | | | | | | | | | | | | | |
| 0,30 | 0,30 | **1,00** | **0,91** | 0,26 | 0,27 | **1,00** | **0,93** | 0,27 | 0,27 | **0,94** | **0,91** | 0,27 | 0,27 | 0,79 | 0,78 |

Tabla 5. Desempeño del modelo *Bagging,* según la métrica puntuación F1, La columna ID corresponde a los escenarios de CTU-13 mencionados en la tabla 1, Las celdas con valores en negrita, corresponden a los mayores rendimientos alcanzados por los algoritmos de aprendizaje

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Multiclasificadores | | | | | | | | | | | | | | | |
| **Bag-LR** | | **Bag-EXTRA** | | **Bag-SGD** | | **Bag-CART** | | **Bag-NAIVE** | | **Bag-KNN** | | **Bag-LSVC** | | **Bag-MLP** | |
| ID | entr | test | entr | test | entr | test | entr | test | entr | test | entr | test | entr | test | entr | Test |
| 1 | 0,27 | 0,27 | **1,00** | **0,97** | 0,00 | 0,00 | **1,00** | **0,98** | 0,08 | 0,08 | **0,97** | **0,97** | 0,27 | 0,26 | **0,92** | **0,92** |
| 2 | 0,02 | 0,02 | **0,99** | **0,90** | 0,00 | 0,00 | **0,99** | **0,92** | 0,02 | 0,02 | 0,91 | 0,88 | 0,00 | 0,00 | 0,80 | 0,80 |
| 3 | 0,00 | 0,00 | **1,00** | **1,00** | 0,00 | 0,00 | **1,00** | **0,99** | 0,08 | 0,08 | **0,99** | **0,99** | 0,00 | 0,00 | **0,97** | **0,97** |
| 4 | 0,20 | 0,20 | **1,00** | **0,97** | 0,10 | 0,08 | **1,00** | **0,98** | 0,03 | 0,03 | 0,96 | 0,95 | 0,50 | 0,40 | 0,71 | 0,74 |
| 5 | 0,34 | 0,26 | **0,99** | **0,94** | 0,62 | 0,60 | **0,99** | **0,93** | 0,20 | 0,18 | **0,96** | **0,93** | 0,00 | 0,00 | 0,67 | 0,70 |
| 6 | 0,00 | 0,00 | **1,00** | **0,96** | 0,00 | 0,00 | **1,00** | **0,98** | 0,36 | 0,37 | **0,96** | **0,95** | 0,00 | 0,00 | 0,89 | 0,89 |
| 7 | 0,00 | 0,00 | 0,96 | 0,69 | 0,00 | 0,00 | **1,00** | 0,78 | 0,04 | 0,04 | 0,71 | 0,73 | 0,00 | 0,00 | 0,00 | 0,00 |
| 8 | 0,00 | 0,00 | **1,00** | **0,98** | 0,00 | 0,00 | **1,00** | **0,98** | 0,10 | 0,09 | **0,97** | **0,96** | 0,00 | 0,00 | **0,93** | **0,92** |
| 9 | 0,83 | 0,84 | **1,00** | **1,00** | 0,84 | 0,84 | **1,00** | **0,99** | 0,44 | 0,46 | **0,98** | **0,97** | 0,84 | 0,84 | **0,94** | **0,94** |
| 10 | **0,95** | **0,95** | **1,00** | **1,00** | **0,95** | **0,95** | **1,00** | **1,00** | 0,93 | 0,92 | **1,00** | **1,00** | **0,95** | **0,95** | **1,00** | **1,00** |
| 11 | **0,98** | **0,98** | **1,00** | **0,98** | **0,99** | **0,99** | **1,00** | **1,00** | 0,86 | 0,85 | **1,00** | **1,00** | **0,99** | **0,99** | **1,00** | **1,00** |
| 12 | 0,00 | 0,00 | **0,97** | **0,77** | 0,00 | 0,00 | 0,97 | 0,73 | 0,05 | 0,05 | 0,81 | 0,72 | 0,00 | 0,00 | 0,38 | 0,34 |
| 13 | 0,39 | 0,39 | **1,00** | **0,99** | 0,00 | 0,00 | **1,00** | **0,99** | 0,02 | 0,02 | **0,99** | **0,98** | 0,39 | 0,38 | **0,94** | **0,94** |
|  | Promedio del rendimiento | | | | | | | | | | | | | | | |
| 0,31 | 0,30 | **0,99** | **0,93** | 0,27 | 0,27 | **1,00** | **0,94** | 0,25 | 0,25 | **0,94** | **0,93** | 0,30 | 0,29 | 0,78 | 0,78 |

Tabla 6. Desempeño del modelo *AdaBoost,* según la métrica Puntuación F1, La columna ID corresponde a los escenarios de CTU-13 mencionados en la tabla 1, Las celdas con valores en negrita, corresponden a los mayores rendimientos alcanzados por los algoritmos de aprendizaje

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Multiclasificadores | | | | | | | | | | | |
| **Ada-LR** | | **Ada-EXTRA** | | **Ada-SGD** | | **Ada-CART** | | **Ada-NAIVE** | | **Ada-LSVC** | |
| ID | entr | test | entr | test | entr | test | entr | test | entr | test | entr | test |
| 1 | 0,00 | 0,00 | **1,00** | **0,96** | 0,51 | 0,51 | **0,60** | **0,60** | 0,09 | 0,09 | 0,00 | 0,00 |
| 2 | 0,01 | 0,01 | **1,00** | 0,88 | 0,00 | 0,00 | **0,48** | **0,49** | 0,20 | 0,20 | 0,10 | 0,10 |
| 3 | 0,00 | 0,00 | **1,00** | **1,00** | 0,00 | 0,00 | **0,83** | **0,83** | 0,07 | 0,08 | 0,00 | 0,00 |
| 4 | 0,00 | 0,00 | **1,00** | **0,98** | 0,00 | 0,00 | **0,36** | **0,35** | 0,08 | 0,08 | 0,10 | 0,12 |
| 5 | 0,07 | 0,08 | **1,00** | **0,90** | 0,00 | 0,00 | **0,61** | **0,61** | 0,30 | 0,30 | 0,54 | 0,54 |
| 6 | 0,00 | 0,00 | **1,00** | **0,96** | 0,00 | 0,00 | **0,86** | **0,86** | 0,36 | 0,36 | 0,00 | 0,00 |
| 7 | 0,00 | 0,00 | **1,00** | **0,67** | 0,00 | 0,00 | **0,83** | **0,63** | 0,00 | 0,00 | 0,00 | 0,00 |
| 8 | 0,00 | 0,00 | **1,00** | **0,97** | 0,00 | 0,00 | **0,54** | **0,55** | 0,10 | 0,10 | 0,00 | 0,00 |
| 9 | 0,80 | 0,80 | **1,00** | **0,98** | 0,73 | 0,74 | **0,88** | **0,88** | 0,45 | 0,45 | 0,54 | 0,54 |
| 10 | **0,92** | **0,92** | **1,00** | **1,00** | **0,89** | **0,89** | **0,97** | **0,97** | **0,93** | **0,93** | **0,91** | **0,91** |
| 11 | **0,98** | **0,98** | **1,00** | **1,00** | **0,91** | **0,91** | **0,99** | **0,99** | **0,95** | **0,95** | **0,95** | **0,95** |
| 12 | 0,00 | 0,00 | **1,00** | 0,76 | 0,00 | 0,00 | **0,00** | **0,00** | 0,04 | 0,05 | 0,00 | 0,00 |
| 13 | 0,14 | 0,14 | **1,00** | **0,98** | 0,00 | 0,00 | **0,82** | **0,82** | 0,21 | 0,21 | 0,29 | 0,29 |
|  | Promedio del rendimiento | | | | | | | | | | | |
| 0,22 | 0,23 | **1,00** | **0,93** | 0,23 | 0,23 | **0,67** | **0,66** | 0,29 | 0,29 | 0,26 | 0,27 |

En las tablas 5 y 6 se describe el comportamiento de los ensambles *Bagging* y *Adaboost* respectivamente. En los resultados de la evaluación del modelo *Bagging* se mantiene la tendencia anterior de la evaluación individual de los algoritmos, donde se destacan los modelos basados en ExtraTree, CART y KNN con los rendimientos promedios de 93 %, 94 % y 93 % respectivamente (para el conjunto de datos de test). Se evidencia un crecimiento mínimo de 1 % y máximo de 2 % del promedio de rendimiento. Para el modelo *Adaboost*, los estimadores ExtraTree y CART presentaron los mejores rendimientos promedios de 93 % y 66 % respectivamente, lo que representa un descenso de rendimiento del clasificador basado en CART de evaluación individual.

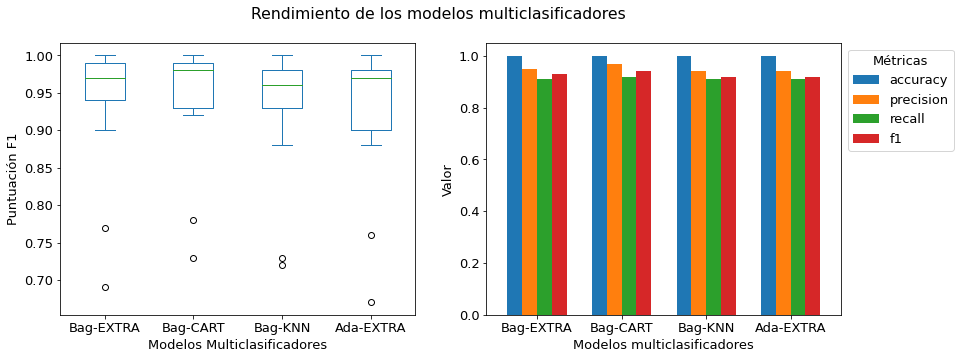


Fig. 1 Representación del intervalo de valores que la métrica puntuación F1 para los modelos multiclasificadores seleccionados considerando los escenarios de CTU-13 (imagen a la izquierda). Valor promedio de las métricas de rendimiento de los modelos seleccionados (imagen a la derecha). En ambos casos empleando el conjunto de datos de *test.*

En la figura 1 se representa el comportamiento de los modelos que ha alcanzado mejor rendimiento. Los patrones atípicos en las gráficas están relacionados con valores de *puntuación F1* que están fuera da normalidad del conjunto de valores hallados en los 13 escenarios.

Teniendo en cuenta los factores seleccionados para experimento se pudo confirmar la influencia de ambos en rendimiento de un modelo clasificador, individual o multiclasificador. Los escenarios más complejos o donde se obtuvo muy poco rendimiento tienen características similares, ya que están conformados fundamentalmente por tráficos de ataques de botnets, que son: Escaneo de puertos (PS), tráfico P2P y HTTP, y se puede asociar el hecho de que sean escenarios con acentuado desequilibrio de clases.

Los resultados sugieren que los algoritmos de aprendizaje supervisados bases o individuales son la parte importante para el rendimiento del modelo multiclasificador homogéneo, pues se mantuvo estable la tendencia de comportamiento de los clasificadores individuales en los multiclasificadores homogéneos. Del mismo modo, el mecanismo de ensamble puede influir negativa o positivamente, ya que se puede observar en la tabla 6, que con el modelo Adaboost el rendimiento mínimo para los mejores modelos descendió en 20 %.

**Comparación con los algoritmos del estado del arte**

En esta sección se comparan algunas propuestas del estado del arte, donde se emplea el conjunto de datos CTU-13 muy utilizado en enfoques genéricos de detección de tráfico de Botnet. La comparación se describe en la tabla 7.

Tabla 7. Comparación con los resultados de los enfoques del estado del arte

|  |  |  |  |
| --- | --- | --- | --- |
| **Referencia** | **Algoritmos** | **Breve descripción** | **Rendimiento (%)** |
| (Echevarría *et al.,* 2022) | Random Forests | Ensemble homogeneo | Exactitud = 98,83 |
| (Zhao *et al.,* 2020) | Bot-AHGCN | Basado en redes neuronales profundas. | Puntuación F1 = 98,48 |
| (Daya *et al.,* 2019) | SOM | Enfoque basado en grafos | Exactitud = 99,95 |
| (Haq & Singh, 2018) | Naive Bayes | Enfoque Bayesiano | Exactitud = 19,43 |
| Rule base table | Enfoque basado en reglas | Exactitud = 87,79 |
| ID3 | Árbol de decisión | Exactitud = 90,27 |
| **Presente trabajo** | *Bagging* y  *Adaboost* | Empleando estimados basados  En ExtraTree, CART, y KNN. | **Exactitud = 100**  **Precisión = 97**  **Recall = 92**  **Puntuación F1 = 94** |

**CONCLUSIONES**

En este trabajo se obtuvieron cuatro modelos multiclasificadores para predecir el tráfico de diferentes botnets, basados en tráficos de IRC, SPAM, *clics* fraudulentos (CF), escaneo de puertos (PS), denegación de servicio (DDoS), FastFlux (FF), *peer-to-peer* (P2P), compilado y controlado por los desarrolladores (US) y HTTP. Tres modelos se obtuvieron empleando el algoritmo *Bagging*, con estimadores basados en los algoritmos ExtraTree, Cart y KNN (Bag-EXTRA, Bag-CART, Bag-KNN), y con el algoritmo Adaboost con el estimador basado en ExtraTree (Ada-EXTRA). Hay una influencia significativa del tipo de algoritmo supervisado usado como estimador en el rendimiento del ensamble. El escenario 7 resulta el más severo en desbalance de clases. Se verificó un descenso significativo de rendimiento de los ensambles, lo que implica poca efectivad de los ensambles cuando el desbalance de clase es severo.

**TRABAJOS FUTUROS**

Para los trabajos futuros se planea implementar los algoritmos seleccionados, empleando otros conjuntos de datos del estado del arte, además de implementar algunas medidas estadísticas para evaluar mejor los resultados de los experimentos. Se plantea también enfocarlo en la captura de datos en una plataforma web de comercio electrónico real y la construcción de un sistema de detección, con dos fases de detección, basado en metaaprendizaje, utilizando los algoritmos Proative Forests y multiclasificadores híbridos.

**REFERENCIAS**

B, H. X., Li, Z., Chu, C., Chen, Y., Yang, Y., Lu, H., Wang, H., & Stavrou, A. (2018). Detecting and Characterizing Web Bot Traffic in a Large E-commerce Marketplace. *ESORICS 2018*, 1: 143-163. https://doi.org/10,1007/978-3-319-98989-1

Barbon S., J., Campos, G. F. C., Tavares, G. M., Igawa, R. A., Proença M.L., J., & Guido, R. C. (2018). Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets. *ACM Transactions on Multimedia Computing, Communications and Applications*, *14*(1s). https://doi.org/10,1145/3183506

Bermúdez, M. D.-C. (2022). Gestión de Gobierno basada en ciencia e innovación: avances y desafíos. *Anales de la Academia de Ciencias de Cuba*, *12*(2): 12-35. http://www.revistaccuba.cu/index.php/revacc/article/view/e1235

Cabri, A., Suchacka, G., Rovetta, S., & Masulli, F. (2018). Online Web Bot Detection Using a Sequential Classification Approach. *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th Intl. Conference on Data Science and Systems*. https://doi.org/10,1109/HPCC/SmartCity/DSS.2018.00252

Daya, A. A., Salahuddin, M. A., Limam, N., & Boutaba, R. (2019). A Graph-Based Machine Learning Approach for Bot Detection. *IFIP/IEEE International Symposium on Integrated Network Management, Washington DC, USA, April 2019*, *April*.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Multiple Classifier Systems*, pp. 1-15.

Echevarría, D. P., Espino, M. M., Pando, H. D., & Chissingui, H. J. (2022). Comercio Electrónico Random Forest For Bot Detection In E-Comerce. *Infomática - XVIII Convención y Feria Internacional*.

Garcia, S., Grill, M., Stiborek, J., & Zunimo, A. (2014). An empirical comparison of botnet detection methods. *Computers and Security Journal, Elsevier*, 45: 100-123. https://doi.org/http://dx.doi.org/10,1016/j.cose.2014.05.011

Garcia, S., Grill, M., Stiborek, J., Zunimo, A., Dietterich, T. G., Suchacka, G., Wotzka, D., Chen, H., He, H., Starr, A., Deng, J., Dong, W., Socher, R., Li, L. L.-J., Li, K., Fei-Fei, L., Balla, A., Stassopoulou, A., Dikaiakos, M. D., … Greensmith, J. (2020). Artificial Intelligence - A Modern Approach. *Computers & Security*, *8*(1): 1-6. https://doi.org/10,1007/s11416-020-00368-6

Han, J., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques, third edition*. Morgan Kaufmann Publishers. http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm\_hrd\_title\_0?ie=UTF8&qid=1366039033&sr=1-1

Haq, S., & Singh, Y. (2018). Botnet Detection using Machine Learning. *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 240-245. https://doi.org/10,1109/PDGC.2018.8745912

Imperva. (2022). *2022 Imperva Bad Bot Report - Evasive Bots Drive Online Fraud*. www.imperva.com

Rahman, R. U., & Tomar, D. S. (2020). Threats of price scraping on e-commerce websites: attack model and its detection using neural network. *Journal of Computer Virology and Hacking Techniques*, *17*(1): 75-89. https://doi.org/10,1007/s11416-020-00368-6

Rovetta, S., Suchacka, G., & Masulli, F. (2020). Bot recognition in a Web store: An approach based on unsupervised learning. *Journal of Network and Computer Applications*, *157*, 102577. https://doi.org/https://doi.org/10,1016/j.jnca.2020,102577

Suchacka, G., Cabri, A., Rovetta, S., & Masulli, F. (2021). Efficient on-the-fly Web bot detection. *Knowledge-Based Systems*, *223*, 107074. https://doi.org/https://doi.org/10,1016/j.knosys.2021,107074

Suchacka, G., & Iwanski, J. (2020). Identifying legitimate Web users and bots with different traffic profiles — an Information Bottleneck approach. *Knowledge-Based Systems*, *197*, 105875. https://doi.org/https://doi.org/10,1016/j.knosys.2020,105875

Suchacka, G., & Sobków, M. (2015). Detection of Internet Robots Using a Bayesian Approach. *IEEE*.

Zhao, J., Liu, X., Yan, Q., Li, B., Shao, M., & Peng, H. (2020). Multi-attributed heterogeneous graph convolutional network for bot detection. *Information Sciences*, 537: 380-393. https://doi.org/https://doi.org/10,1016/j.ins.2020,03.113