

Suscríbete a DeepL Pro para poder editar este documento.  
Entra en [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) para más información.

Redes informáticas 214 (2022) 109073

Listas de contenidos disponibles en ScienceDirect



Redes informáticas

Página web de la revista: www.elsevier.com/locate/comnet

Aprendizaje automático adversarial para la detección de intrusiones en redes: Un estudio comparativo



Houda Jmilaa , Mohamed Ibn Khedher b,∗

* *SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, Francia*b  *IRT-SystemX, 2 Boulevard Thomas Gobert , 91120 Palaiseau, Francia*

ARTÍCULO INFO

*Palabras clave:*

Detección de intrusiones en redes

Robustez adversarial de la IA

Ataque adversario

Técnica de defensa

NSL-KDD

UNSW-NB15

RESUMEN

La detección de intrusos es un tema clave de la ciberseguridad. Su objetivo es proteger los sistemas informáticos y las redes de intrusos y ataques malintencionados. Los sistemas tradicionales de detección de intrusos (IDS) siguen un enfoque basado en firmas, pero en las dos últimas décadas se han propuesto con fuerza diversas técnicas de aprendizaje automático (ML) que han demostrado su eficacia. Sin embargo, el ML se enfrenta a varios retos, siendo uno de los más interesantes la aparición de ataques de adversarios para engañar a los clasificadores. Resolver esta vulnerabilidad es fundamental para evitar que los ciberdelincuentes aprovechen los fallos de ML para burlar los IDS y dañar datos y sistemas.

Algunos trabajos de investigación han estudiado la vulnerabilidad de los IDS basados en ML frente a ataques de adversarios, aunque la mayoría de ellos se han centrado en clasificadores basados en aprendizaje profundo. A diferencia de ellos, este artículo presta más atención a los clasificadores superficiales que siguen siendo ampliamente utilizados en los IDS basados en ML debido a su madurez y simplicidad de implementación. Más detalladamente, evaluamos la robustez de 7 NIDS basados en ML poco profundos, incluyendo Adaboost, Bagging, Gradient boosting (GB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Classifier (SVC) y también una Red de Aprendizaje Profundo, frente a varios ataques adversarios ampliamente utilizados en el estado del arte (SOA). Además, aplicamos una técnica de defensa de aumento de datos gaussiana y medimos su contribución a la mejora de la robustez del clasificador [1]. Realizamos experimentos exhaustivos en diferentes escenarios utilizando el conjunto de datos de referencia NSL-KDD [2] y el conjunto de datos UNSW-NB 15 [3]. Los resultados muestran que los ataques no tienen el mismo impacto en todos los clasificadores, que la robustez de un clasificador depende del ataque y que debe considerarse un compromiso entre rendimiento y robustez en función del escenario de detección de intrusiones en la red.

**1. Introducción**

La protección de los sistemas y redes informáticos contra los ciberataques ha sido una preocupación creciente en los últimos años. Aunque la mayoría de los sistemas se construyen con características de seguridad mejoradas, sigue existiendo un gran número de vulnerabilidades. Entre ellas figuran el acceso no deseado a sistemas e información, la destrucción o alteración de datos, etc. Los sistemas de detección de intrusos desempeñan un papel fundamental en el proceso de defensa de la red y permiten a los operadores de red identificar con precisión los ataques a la seguridad. Existen principalmente dos categorías de IDS: IDS basados en la red e IDS basados en el host, que se describen a continuación:

* + *Los IDS basados en red (NIDS)* supervisan y analizan el tráfico de red en distintas capas para detectar intrusos.
  + *Los IDS basados en host (HIDS)*, supervisan la infraestructura informática para detectar cambios internos explotando indicadores del host como archivos de registro de sensores, recursos de disco, procesos de información de cuentas de usuario, etc.

∗ Autor correspondiente.

*Dirección de correo electrónico:* ibnkhedhermohamed@hotmail.com (M.I. Khedher).

Este artículo se centra en los IDS basados en red. El continuo aumento del número y tipos de amenazas de red contemporáneas [4] motiva este interés.

Tanto los enfoques NIDS como HIDS pueden clasificarse en las siguientes categorías:

* *Los enfoques basados en el uso indebido* (también llamados *basados en firmas*) explotan indicadores (o firmas) extraídos previamente de ataques *conocidos*. Las firmas se generan manualmente para cada nuevo ataque. Por lo tanto, mantener una lista actualizada de firmas es costoso debido al creciente número y diversidad de ataques.
* *Los enfoques basados en anomalías* modelan el comportamiento normal de la red, en contraposición al comportamiento malicioso. Aunque estos enfoques son ca-paces de detectar nuevos ataques, adolecen de una alta tasa de falsas alarmas porque un *nuevo comportamiento normal* puede ser detectado como malicioso.

https://doi.org/10.1016/j.comnet.2022.109073

Recibido el 6 de octubre de 2021; Recibido en versión revisada el 26 de abril de 2022; Aceptado el 23 de mayo de 2022

Disponible en línea el 3 de junio de 2022

1389-1286/© 2022 Elsevier B.V. Todos los derechos reservados.

*H. Jmila y M.I. Khedher*

|  |  |  |
| --- | --- | --- |
| **Nomenclatura** | Precisión de la predicción de anomalías |  |
| AAC |  |
| AdvML | Aprendizaje automático adversarial |  |
| ANN | Red neuronal artificial |  |
| BIM | Método iterativo básico |  |
| C&W | Carlini y Wagner |  |
| CNN | Redes neuronales convolucionales |  |
| DDoS | Denegación de servicio distribuida |  |
| DNN | Red neuronal profunda |  |
| DT | Árbol de decisión |  |
| FFNN | Red neuronal directa |  |
| FGSM | Método de signo de gradiente rápido |  |
| FNR | Tasa de falsos negativos |  |
| GAN | Redes generativas adversariales |  |
| GAN | Red Generativa Adversarial |  |
| GB | Refuerzo de gradiente |  |
| GMM | Modelo de mezcla gaussiana |  |
| HIDS | IDS en casa |  |
| HSJ | Hop Skip Jump |  |
| IDS | Sistema de detección de intrusos |  |
| IoT | Internet de los objetos |  |
| JSMA | Ataque al mapa de saliencia basado en jacobianos |  |
| KNN | Vecinos más próximos K |  |
| ML | Aprendizaje automático |  |
| NIDS | IDS en red |  |
| PGD | Descenso gradual proyectado |  |
| RF | Aprendizaje por refuerzo |  |
| RNN | Red neuronal recurrente |  |
| SOA | Estado de la técnica |  |
| VPC | Clasificador de vectores de apoyo |  |
| SVM | Máquina de vectores soporte |  |
| TAC | Precisión total de la predicción |  |
| PAU | Perturbaciones Adversariales Universales |  |
| ZOO | Optimización de orden zerótico |  |

La detección de anomalías suele ser considerada por la comunidad como más prometedora que la detección basada en firmas, ya que es capaz de detectar *ataques desconocidos*. Por ello, este artículo se centra en los NIDS basados en anomalías.

En los últimos años, los enfoques de ML se han utilizado ampliamente para la detección de anomalías. Los enfoques existentes pueden clasificarse en *modelos superficiales (o clásicos)* [5] y *modelos de aprendizaje profundo* [6,7]. El aprendizaje profundo implica varios niveles de representación y varias capas de unidades de procesamiento no lineal. Por el contrario, todos los enfoques que no son de aprendizaje profundo pueden calificarse de aprendizaje superficial, lo que incluye la mayoría de los modelos de aprendizaje automático convencionales propuestos antes de 2006 y las redes neuronales con una sola capa oculta de nodos [8]. Los enfoques superficiales más populares son Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Hidden Markov Models (HMM) y Ensemble Learning. Se han utilizado modelos de aprendizaje superficial y profundo con resultados prometedores.

Mientras que la mayor parte de la investigación se centra en el diseño de nuevos IDS basados en ML, este artículo destaca las vulnerabilidades de los sistemas ML frente a los ataques adversarios. Los ataques de adversarios permiten que un cambio pequeño y cuidadosamente diseñado en la entrada del clasificador ML altere completamente la salida del sistema. El *Aprendizaje Automático Adversarial* (AdvML) es el área de investigación que estudia estas vulnerabilidades. Se ha explorado ampliamente en los últimos años, en particular en el campo de la visión por ordenador [9]. El estudio del AdvML en ciberseguridad también merece una gran atención dada la sensibilidad de este campo y la necesidad de preservar la confidencialidad, integridad y disponibilidad de datos y sistemas. Es esencial evaluar

*Redes informáticas 214 (2022) 109073*

la *solidez* de los sistemas de detección de intrusiones basados en ML antes de desplegarlos en la red. De este modo se evita que los ciberdelincuentes aprovechen las vulnerabilidades de los ML para eludir los IDS y dañar datos y sistemas. La *robustez* de un clasificador ML se define como su capacidad para mantener su precisión frente a *muestras adversas*. Una muestra *adversa* es una instancia de entrada con una pequeña perturbación que se predice erróneamente. En función de los resultados de la evaluación de la robustez, pueden aplicarse técnicas de defensa adecuadas para mejorar la robustez de los NIDS.

Debido a la adopción generalizada de enfoques de aprendizaje profundo para NIDS, la mayoría de los trabajos de investigación evalúan la robustez de los NIDS basados en aprendizaje profundo [10]. Sin embargo, los modelos ML superficiales siguen siendo ampliamente utilizados en NIDS debido a su simplicidad y madurez de implementación [11]. Por lo tanto, es interesante estudiar su robustez en un entorno adverso. Este artículo se centra en la evaluación de NIDS basados en ML superficial frente a varios ataques adversariales ampliamente utilizados en el estado del arte.

En este trabajo, evaluamos la robustez de 7 clasificadores superficiales, incluyendo Adaboost, Bagging, Gradient boosting, Logistic regression, Decision Tree, random forest, Support Vector Classifier y también un Deep Learning Network, frente a una amplia gama de ataques (un ataque se define como un método de generación de ejemplos adversos). En particular, consideramos ataques de caja blanca y de caja gris/negra. En los ataques de caja blanca, el atacante tiene acceso completo a toda la información sobre la NIDS basada en ML, mientras que en los ataques de caja gris/negra, el atacante tiene poco o ningún conocimiento de la NIDS basada en ML. Los ataques de caja gris/negra son interesantes porque representan el escenario más realista para los ataques de los adversarios. Examinar los ataques de caja blanca es útil para los fabricantes de IDS que tienen pleno acceso a su sistema y desean evaluar su rendimiento frente a ataques de adversarios.

Este documento aporta las siguientes contribuciones principales:

* Un estudio claro y estructurado de los ataques adversariales y las técnicas de defensa más utilizadas, además de una revisión exhaustiva de los trabajos actuales sobre NIDS ML adversariales.
* Estudio en profundidad del impacto de los ataques de adversarios en los NIDS basados en ML. Se exploran varios tipos de ataques (9 de caja blanca y ataques de caja gris/negra) con especial atención a los clasificadores poco profundos. De hecho, a diferencia de la inmensa mayoría de trabajos que estudian el comportamiento de las NIDS en un entorno adversario y se centran en enfoques de aprendizaje profundo, este trabajo se centra en algoritmos poco profundos, que siguen siendo ampliamente utilizados en las NIDS basadas en ML gracias a su sencillez de implementación y madurez. Por tanto, también merece la pena explorar la evaluación de su rendimiento en un entorno adversario.
* Evaluación de la contribución de una técnica gaussiana de defensa contra el aumento de datos a la mejora de la robustez de los clasificadores.
* Valiosos resultados y conclusiones que pueden ayudar a los investigadores de seguridad a mejorar la robustez de sus NIDS. Estos resultados se basan en exhaustivos experimentos realizados en distintos escenarios.
* Las etapas del estudio realizado representan un marco que detalla los pasos a seguir para evaluar la sensibilidad de los NIDS a los ataques de adversarios y mejorar su robustez.

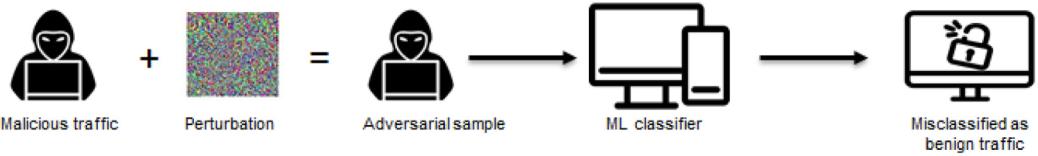
El documento se estructura de la siguiente manera. La sección 2 describe los retos en el campo de la detección de intrusiones en red. La Sección 3, proporciona un estado del arte de los ataques adversarios más comúnmente utilizados y las técnicas de defensa, así como un estudio exhaustivo de los enfoques AdvML en el campo de NIDS. La Sección 4 describe nuestro estudio de evaluación, incluyendo los parámetros de evaluación y el protocolo. La sección 5 detalla los resultados experimentales. La Sección 6 ofrece una discusión y la Sección 7 concluye el artículo.

**2. El reto de la detección de intrusiones en la red**

La detección de intrusiones en la red es una tarea compleja por muchas razones. Los retos pueden estar relacionados con la naturaleza de los datos de tráfico de red o con el modelo de decisión inherente a los NIDS, como se describe a continuación.

2

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*



**Fig. 1.** Generación de ataques adversarios en NIDS. Una muestra adversa se genera añadiendo una pequeña perturbación a la muestra original. De este modo, el tráfico perturbado malicioso puede ser clasificado erróneamente como benigno y eludir así el sistema de detección de intrusos. Esto puede tener graves consecuencias para el sistema.

*2.1. Retos relacionados con la naturaleza de los datos de tráfico de la red*

* *Retos relacionados con la explotación de datos*: en la detección de anomalías en la red, los paquetes capturados representan parcialmente todo el tráfico de la red, ya que los puntos de observación suelen estar muy repartidos. Por tanto, los datos capturados suelen ser escasos, enormes y contener datos redundantes o poco informativos, lo que dificulta su explotación.
* *Conjuntos de datos desequilibrados*: En la mayoría de los conjuntos de datos IDS, la cantidad de datos nor-males domina los datos. Esto se debe a la baja frecuencia de los ataques en comparación con el comportamiento normal. Esto hace que los conjuntos de datos disponibles no estén equilibrados. Por tanto, los algoritmos clásicos de aprendizaje automático deben adaptarse al contexto de los conjuntos de datos desequilibrados.
* *Variedad de ataques*: el panorama de los ataques cambia con frecuencia, ya que los atacantes desarrollan constantemente nuevos métodos. Las herramientas de detección y análisis de ataques deben actualizarse y evolucionar continuamente.

*2.2. Retos relacionados con el modelo de decisión*

* + *Escalabilidad*: se trata de un reto habitual para los algoritmos de aprendizaje estadístico. Se define por la capacidad del algoritmo para funcionar con normalidad incluso con datos de gran dimensión.
  + *IDS en tiempo real:* El objetivo de un IDS es detectar ataques y detenerlos antes de que dañen el sistema. Por lo tanto, el diseño de un IDS en tiempo real es muy importante. Un IDS en tiempo real también debe ser eficiente y flexible para funcionar en la mayoría de los ordenadores comerciales.
  + *Alta tasa de falsos positivos:* Se trata de la notificación de un elevado número de falsas alarmas que corresponden a actividad legítima que ha sido clasificada erróneamente por el IDS. Reconocer las alarmas verdaderas entre el enorme volumen de alarmas es una tarea complicada y que requiere mucho tiempo. Por lo tanto, reducir las falsas alarmas es un problema grave para garantizar la eficacia y el uso de los IDS.
  + *Vulnerabilidad a los ataques adversarios:* Este aspecto se ha descrito anteriormente y es el objeto de este artículo. El objetivo es examinar y reducir la vulnerabilidad de varios clasificadores NIDS a ataques de adversarios [12,13].

1. **Ataques adversarios y técnicas de defensa en NIDS: Antecedentes y revisión**

*3.1. Preliminares*

Generar un ataque adversario implica añadir una pequeña perturbación a la muestra de entrada para que la etiqueta de salida se clasifique erróneamente [14,15]. Esto se ilustra en la Fig. 1 en el contexto de los NIDS. Formalmente, sea la muestra de datos de entrada original, el clasificador y = ( ) la etiqueta asociada a . Una muestra de datos′ se considera una muestra adversaria de cuando′ está cerca de bajo una métrica de distancia específica mientras (′ ) ≠ . Los ataques adversarios en seguridad de redes pueden clasificarse en dos dimensiones: el *conocimiento del atacante* y el *objetivo del atacante*:

1. *El conocimiento del atacante*: describe el grado de conocimiento del adversario sobre el sistema NIDS. Podemos caracterizar tres niveles de peligro de ataque [16]:

* *Ataques de caja blanca*: el atacante se encuentra en la posición más favorable, en la que tiene pleno acceso a toda la información sobre la NIDS basada en ML. Esto incluye los datos de entrenamiento y la arquitectura del modelo de aprendizaje, la decisión y los parámetros (gra-diente, función de pérdida, etc.). Afortunadamente, esto no suele ser factible en la mayoría de los ataques adversarios reales.
* *Ataques de caja negra:* Este es el caso opuesto en el que el atacante ignora completamente el sistema NIDS basado en ML y sus entradas/salidas. Se puede argumentar que un verdadero ataque de caja negra es imposible y rara vez tiene éxito.
* *Ataques de caja gris:* este escenario supone un enfoque más realista, en el que el atacante tiene cierto nivel de conocimiento del NIDS basado en ML, y puede tener un acceso limitado a los datos de entrenamiento. El adversario no dispone de la información exacta, pero sí de la suficiente para poder atacar el sistema ML y provocar su fallo.

Nótese que en la literatura, por abuso del lenguaje, el término ''ataques de caja negra'' también se utiliza para ''ataques de caja gris'' (por ejemplo, el ataque ZOO se denomina ataque de caja trasera en [17]). En este artículo, utilizamos los términos ''gray/black-box'' para referirnos a los ataques de caja gris descritos a continuación, para matizar entre esta definición y el término ''black-box'' ampliamente utilizado en la literatura.

1. *El objetivo del atacante*: depende de si quiere simplemente engañar al sistema o inducir una predicción precisa para determinadas entradas. Se pueden enumerar dos formas de ataque:
   * *Ataques dirigidos:* dirigen el algoritmo ML a una clase específica, es decir, el adversario engaña al clasificador para que prediga todos los ejemplos del adversario como una clase objetivo específica.
   * *Ataque no dirigido: tiene como* objetivo clasificar erróneamente la muestra de entrada fuera de su clase original, independientemente de la nueva clase de salida. Son más fáciles de implementar porque hay más alter-nativas disponibles para reorientar la salida. Tenga en cuenta que en los problemas de clasificación binaria, los ataques dirigidos y no dirigidos son equivalentes.

*3.2. Estudio de los enfoques de generación de ataques adversarios*

Dado que muchos enfoques de generación de ataques adversariales pueden aplicarse tanto a escenarios dirigidos como no dirigidos, nos basaremos más bien en los conocimientos del atacante para clasificarlos.

*3.2.1. Ataques de caja blanca*

*Fast Gradient Sign Method (FGSM) [18].* crea ejemplos adversos añadiendo ruido a la muestra original a lo largo de las direcciones del gradiente.

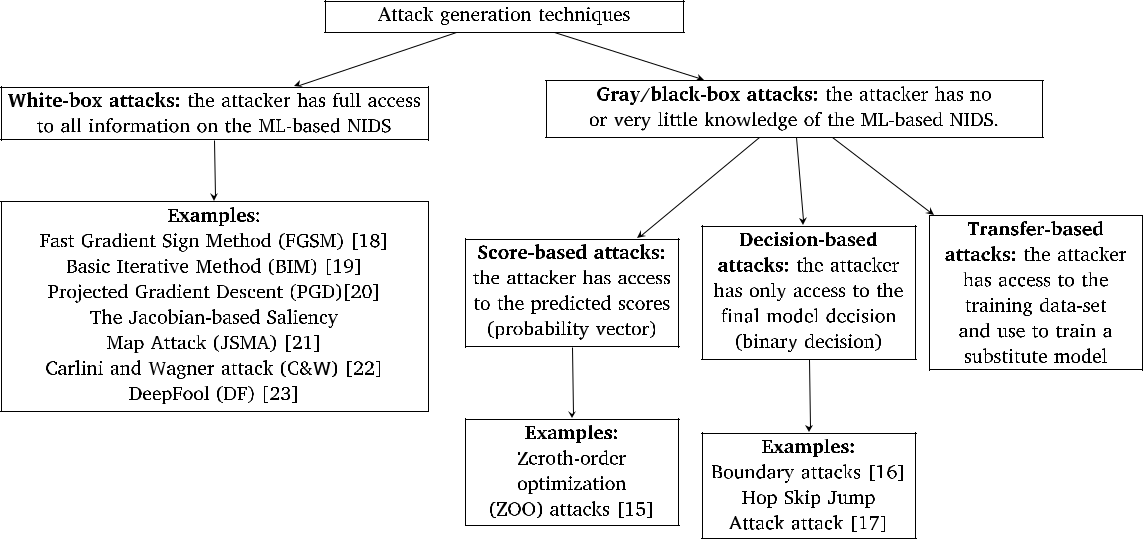
En la literatura reciente también se han utilizado dos extensiones iterativas del FGSM, a saber, el *Método Iterativo Básico (BIM) [19]* y el *Descenso Gradiente Proyectado (PGD) [20]*.

*El Ataque de Mapa de Saliencia Basado en Jacobianos (JSMA) [21]* genera ejemplos de publicidad utilizando derivadas hacia adelante (es decir, el modelo Jacobiano). JSMA perturba iterativamente características/componentes de la entrada de uno en uno en lugar de perturbar toda la entrada para engañar al clasificador.

*Las Perturbaciones Adversariales Universales (UAP) [22].* son un tipo especial de ataques no dirigidos que consisten en crear una perturbación constante que clasifique erróneamente una fracción específica de las muestras de entrada.

3

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*



**Fig. 2.** Técnicas de ataque adversario.

*DeepFool (DF) [23].* es un ataque no dirigido basado en el cálculo de la distancia mínima entre la entrada original y el límite de decisión.

*Ataque de Carlini y Wagner (C&W) [24].* Los autores formulan la búsqueda de una muestra adversarial como un problema de optimización con el siguiente objetivo:

min *,* + ) + *.*( + ) sujeto a ∈

donde denota la perturbación adversarial, (*., .*) denota la métrica de distancia ,00 o∞ , y ( + ) define la función de coste tal que ( + ) 0 si y sólo si el modelo clasifica correctamente + (es decir, le da la misma etiqueta que ).

*3.2.2. Ataques de caja gris/negra*

*Ataques basados en* puntuaciones*.* : el atacante tiene acceso a las puntuaciones previstas,

es decir, la probabilidad de que cada etiqueta predicha pertenezca a la clasifica-

del modelo. Algunos ejemplos son la *optimización de orden Zeroth*

*(ZOO) [17]*.

*Ataques basados en decisiones.* : el atacante sólo tiene acceso a la decisión final del modelo (decisión binaria) sin ninguna puntuación de confianza. Algunos ejemplos son *los ataques de límite [25]* y *el ataque de salto de salto [26]*.

*Ataques basados en la transferencia*: el atacante tiene acceso al conjunto o a una parte del conjunto de datos de entrenamiento y lo utiliza para entrenar otro modelo totalmente observable, denominado ''modelo sustituto'', que pretende emular el modelo atacado, denominado ''modelo objetivo''. Las perturbaciones adversarias que pueden sintetizarse a partir del ''modelo sustituto'' se utilizan para atacar el ''modelo objetivo''.

Remitimos al lector a [27-30] para más información sobre ataques adversarios. La Fig. 2 resume las distintas técnicas de generación de ataques adversarios descritas anteriormente.

*3.3. Defensa*

Una técnica de defensa tiene como objetivo mejorar la robustez del modelo frente a ataques de adversarios. En [31], se destacan las tres categorías siguientes de técnicas de defensa:

* *Modificar los datos de entrada:* Estas técnicas no se ocupan directamente de los modelos de formación, sino que se basan en la modificación de los datos de formación durante el entrenamiento o en la modificación de los datos de entrada durante las pruebas. Por ejemplo, la técnica de *aumento de datos gaussiana [32]* consiste en aumentar el conjunto de datos original con copias de las muestras originales para

a la que se ha añadido ruido gaussiano. La idea subyacente es que forzar al modelo a hacer la misma predicción para una instancia verdadera y su versión ligeramente perturbada debería aumentar su capacidad de generalización. Este método se utiliza mucho por su sencillez, su facilidad de aplicación y su eficacia tanto contra los ataques de caja gris como contra los de caja blanca.

* *Modificar el clasificador:* Se trata de modificar el modelo de clasificación original cambiando las funciones de pérdida, añadiendo capas/subredes adicionales, etc. Por ejemplo, el método *Gradient Mask-ing* modifica un modelo de aprendizaje automático para ocultar su gradiente a un atacante.
* *Añadir un modelo externo:* estos métodos mantienen intacto el modelo original y le añaden uno o más modelos externos durante las pruebas. Por ejemplo, los autores de [33] utilizaron *Redes Adversariales Generativas (*GAN) para entrenar la red a lo largo de una red generadora que intenta generar una perturbación a esa red.

La Fig. 3 resume las diferentes técnicas de defensa descritas anteriormente.

*3.4. Defensa y ataques en IDS*

La Tabla 1 presenta y compara investigaciones recientes sobre NIDS basados en ML en entornos adversarios. Para cada trabajo de investigación, destacamos (i) los clasificadores ML evaluados (ii) el conjunto de datos de evaluación (iii) los algoritmos de ataque adversario y (iv) las técnicas de defensa, en su caso. En particular, clasificamos los clasificadores ML evaluados en dos categorías: aprendizaje *superficial* y *profundo*. También dividimos las técnicas de generación de ataques adversarios en *técnicas del estado del arte*, es decir, técnicas inspiradas en el campo de la visión por computador, y *nuevas técnicas diseñadas por los autores*.

La primera fila de la tabla presenta una estadística que revela las tendencias de la bibliografía. Se observa que la mayoría de la literatura (95%) evalúa la robustez de las técnicas de aprendizaje profundo, mientras que una minoría (37%) evalúa el aprendizaje superficial. La mayoría de estos últimos se centran en un único tipo de ataque adversario, propuesto por los autores, y no abordan los diversos ataques adversarios ampliamente utilizados en la literatura.

La evaluación de las NIDS basadas en ML superficial en entornos adversarios requiere más estudios. Para llenar este vacío, este artículo evalúa las NIDS basadas en ML superficial frente a los métodos de generación de ataques más utilizados en la literatura. Sólo [41] ha tratado este tema, pero los autores no exploraron las técnicas de defensa.

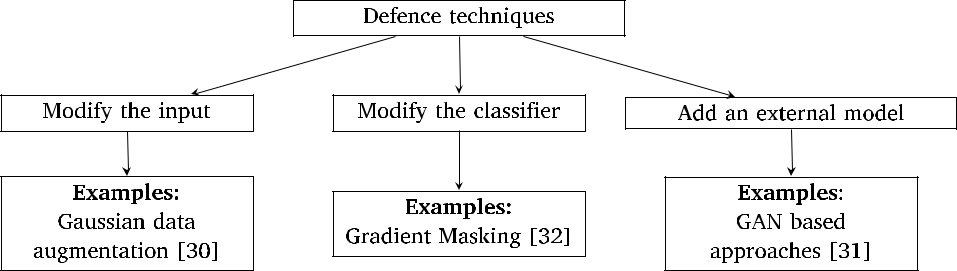
Más detalladamente, evaluamos diversos algoritmos ML ampliamente utilizados [76] en el ámbito de los NIDS cuando se exponen a pruebas de caja blanca y de

4

+ + +

+

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*



**Fig. 3.** técnicas de defensa (Ref. [34]).

ataques adversarios gray/back-box generados por algoritmos SOA bien conocidos. Además, exploramos el efecto de la técnica de defensa de aumento de datos gaussianos en diferentes configuraciones de clasificación.

**4. Evaluación de la robustez de los IDS frente a ataques de adversarios**

Los ataques de caja blanca investigados en este trabajo son: Ataque FGSM, BIM, ataque PGD, JSMA, ataque DeepFool, ataque Carlini y Wagner. Los ataques de caja gris/negra examinados son: Zoo Attack, Boundary at-tack y Hop Skip Jump Attack. Además, generamos ataques adversarios adicionales de forma ingenua utilizando ruido gaussiano, con diferentes intensidades (los valores son 0,01, 0,1 y 0,2).

Para comparar equitativamente la robustez de diversos clasificadores, tenemos que aplicar los mismos ataques, con la misma configuración y los mismos hiperparámetros a todos los clasificadores. Sin embargo, los ataques de caja blanca dependen en gran medida del tipo de clasificador al que atacan; por ejemplo, FGSM, BIM, PGD y JSMA utilizan el gradiente del clasificador para generar los ataques, por lo que solo pueden aplicarse a clasificadores basados en gradientes.

Para superar este problema, proponemos utilizar un clasificador sustituto externo basado en DNN, que llamamos '' *Generador*'' al que aplicamos todos los ataques de caja blanca para generar las muestras adversarias, *en las mismas condiciones*. A continuación, las muestras generadas por cada tipo de *ataque de caja blanca* se introducen en los clasificadores para medir su robustez frente a dicho ataque. Esta idea se basa en la *propiedad transferible de los ataques adversarios* [77], que muestra que el efecto del ataque puede ser transferido a otros modelos ML, incluyendo el ''Generador'' en nuestro caso. Utilizamos una DNN compuesta por 7 capas completamente conectadas con dimensiones que van de 1024 a 32. De una capa a otra, la dimensión se divide por 2. La arquitectura del generador es diferente de la red neuronal evaluada. Es más compleja para hacer más compleja la generación de las muestras adversarias.

Para mejorar la robustez de los IDS, se pueden aplicar técnicas defensivas. Para obtener el sistema NIDS más robusto, el fabricante debe seguir un procedimiento iterativo:

1. Construir un NIDS básico
2. Evaluar su robustez frente a un conjunto de ataques adversos.
3. Aplicar una técnica de defensa para aumentar su robustez
4. Repita (2) y (3) hasta que el nivel de robustez del modelo sea aceptable.

En este trabajo, nos centramos en los pasos (2) y (3), es decir, evaluamos la robustez de los clasificadores frente a ataques adversarios y, a continuación, aplicamos **la** técnica de defensa de **aumento de datos gaussianos** y medimos su contribución a la mejora de su robustez.

A continuación describimos el montaje experimental y presentamos los resultados.

*4.1. Descripción de los conjuntos de datos*

*4.1.1. Conjunto de datos NSL-KDD [2]*

El conjunto de datos NSL-KDD se deriva del conjunto de datos KDDCup 99 y aborda los problemas de este último, a saber, los registros irrelevantes y el desequilibrio de datos entre registros normales y anormales. Un registro se define como

por 41 características, incluidas 9 características básicas de conexiones TCP individuales, 13 características de contenido dentro de una conexión, 9 características temporales calculadas dentro de una ventana de tiempo de dos segundos y otras 10 características. El conjunto de datos contiene 24 tipos de ataques, agrupados en 4 categorías: denegación de servicio (DoS), remoto a local (R2L), usuario a root (U2R) y sondeo. El conjunto de datos se divide en subconjuntos de entrenamiento y prueba que contienen 100.778 y 25.195 muestras, respectivamente. En este estudio se utilizan los conjuntos originales de entrenamiento NSL-KDD Train+ y de prueba NSL-KDD Test+ del conjunto de datos NSL-KDD.

*4.1.2. UNSW-NB15 [3]*

El conjunto de datos fue creado por el grupo de investigación en ciberseguridad del Centro Australiano de Ciberseguridad (ACCS) en 2015. Contiene aproximadamente 100 GB de datos sin procesar (tráfico normal y malicioso). Se utilizaron 9 tipos diferentes de ataques para generar el tráfico malicioso: Fuzzers, Análisis, Backdoors, DoS, Exploits, Genéricos, Reconocimiento, Shell-code y Gusanos. Cada muestra se describe mediante 49 características generadas por varias herramientas de extracción de características. El conjunto de datos se divide en subconjuntos de entrenamiento y de prueba que contienen 175.341 y 82.332 muestras respectivamente. En este estudio se utiliza el conjunto original de entrenamiento UNSW\_NB15\_training-set y el 10% (muestras seleccionadas al azar) del conjunto de prueba UNSW\_NB15\_testing-set.

*4.1.3. Preprocesamiento de datos*

Además, el conjunto de entrenamiento se divide en conjuntos de entrenamiento y validación según **8:2**. Con el fin de proporcionar datos más adecuados para el clasificador, se realizan los pasos de codificación en caliente y normalización de datos. La codificación en caliente consiste en codificar los atributos categóricos, como protocolo, servicio y estado, en una matriz numérica en caliente. A continuación, estos datos se normalizan entre 0 y 1 para producir valores más homogéneos.

*4.2. Los clasificadores evaluados*

Evaluamos el rendimiento de varios algoritmos de ML bien conocidos: Adaboost, Bagging, Gradient boosting, Logistic regression, Decision Tree, Random Forest, Support Vector Classifier (SVC) y también una Deep Learning Network. Se considera la clasificación binaria. Todos nuestros modelos se implementaron utilizando los paquetes TensorFlow [78], Keras [79] y scikit-learn [80]. Para generar muestras adversariales, utilizamos el framework de código abierto IBM Robustness Toolbox (ART) [81]. Todos los hiperparámetros de los clasificadores y algoritmos utilizados para generar las muestras adversariales *se han establecido en sus valores predeterminados* para facilitar la comparación de los distintos escenarios de evaluación.

*4.3. Indicadores de robustez*

Consideramos las dos métricas siguientes:

* **Precisión**: mide la capacidad del IDS para clasificar correctamente el tráfico malicioso y el legítimo. Representa el porcentaje de

el número de registros clasificados correctamente sobre el número total de registros. =

5

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*

**Cuadro 1**

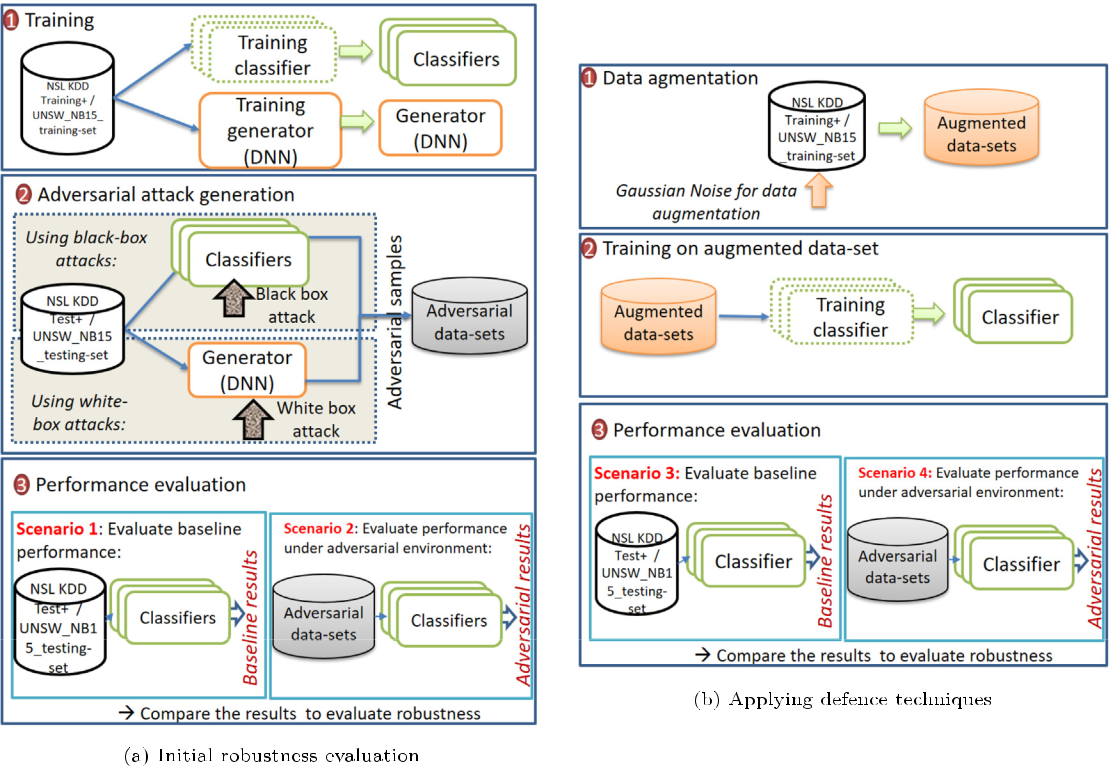
Resumen de trabajos de investigación relacionados con la alineación de documentos.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Trabajo | Año | Algoritmo |  | Conjunto de datos |  | Ataque adversario |  | técnica de defensa |  |
|  |  | Clásico ML | Profundo |  |  | SOA | Diseñado |  |  |
|  |  |  | Aprender |  |  |  |  |  |  |
| (%) | 2018 | 37% | 95% | NS-LKDD | 53% | | 47% | 53% |  |
| [35] | ✗ | DNN | ZOO, con sede en Gan | | ✗ | ✗ |  |
| [36] | 2018 |  | DNN | NSL-KDD | ataque | |  |  |  |
| ✗ | FGSM, JSMA | | ✗ | ✗ |  |
| [37] | 2019 | AdaBoost | MLP | CTU [38] | DeepFool, C&W | | Alterar algunos | Eliminar |  |
| ✗ | |  |
|  |  | DT, GB |  | CICIDS [39] |  |  | características valores | características alteradas |  |
|  |  | KNN, LR |  | BOTNET [40] |  |  |  |  |  |
| [41] | 2019 | RF, SVM | DNN | NSL-KDD | FGSM, PGD | |  |  |  |
| LR, RF | ✗ | ✗ |  |
| [42] | 2019 | SVM | KitNET [43] | Kitsune [43] | L-BFGS,SPSA | |  |  |  |
| ✗ | FGSM | | ✗ | ✗ |  |
|  |  |  |  |  | JSMA | |  |  |  |
|  |  |  |  |  | C&W | |  |  |  |
| [44] | 2019 |  | FFNN, SNNs | BoT-IoT [45] | Método de la red elástica | |  | Normalización de características |  |
| ✗ | FGSM | | ✗ |  |
| [46] | 2019 | KitNET [43] |  | CICIDS [39] | BIM, PGD | | Alterar el | Reconstrucción |  |
| ✗ | ✗ | |  |
| [47] | 2020 | DAGMM [48] |  |  |  |  | paquetes a | de Parcial |  |
|  |  | BiGAN [49] |  |  |  |  | imitar el tráfico benigno | Observación |  |
| [50] | 2020 | KNN, LR | ✗ | DARPA SYN | ✗ | | Perturbar algunos | ✗ |  |
|  |  | RF, SVM |  | inundación [51] |  |  | características importantes |  |  |
| [52] | 2020 | DT, LR | MPL | CICIDS [39] |  |  | Ataque basado en GAN |  |  |
| ADFA-LD [53] | ✗ | | ✗ |  |
| [16] | 2020 | NB, RF | Kitnet [43] | DREBIN [54] |  |  | Ataque de fuerza bruta | Formación contenciosa |  |
| DT, IF | Kitsune [43] | ✗ | | Mutación aleatoria |  |
|  |  | LR, SVM | MLP |  |  |  | y duplicación | Selección de características |  |
|  |  |  |  |  |  |  | de las características seleccionadas | Característica adversarial |  |
| [55] | 2020 |  | CNN, AE | MNIST [56] | FGSM, JSMA | |  | reducción |  |
| ✗ | ✗ | ✗ |  |
| [57] | 2020 | ✗ | CNN, RNN | UNSW-NB15 | FGSM, BIM, C&W | | ✗ | Min-max |  |
| [58] | 2020 |  | ANN | UNSW-NB15 | PGD, DeepFool | |  | Min-max |  |
| ✗ | DNN | FGSM, BGA, BCA | | ✗ |  |
| [47] | 2020 | ✗ | KitNET [43] | MIRAI [59] | ✗ | | Utilizar el mapa de saliencia | ✗ |  |
|  |  |  |  |  |  |  | para identificar |  |  |
| [60] | 2020 |  | RNN | NSL-KDD | JSMA | | características y perturbarlas |  |  |
| ✗ | ✗ | ✗ |  |
| [61] | 2020 | ✗ | MLP, CNN | CICIDS [39] |  | JSMA | ✗ | Modelo de votación |  |
|  |  |  | CNN, LSTM |  |  |  |  | Ensamblaje |  |
|  |  |  |  |  |  |  |  | Ensemble Adversarial |  |
|  |  |  |  |  |  |  |  | Formación |  |
|  |  |  |  |  |  |  |  | Consulta adversarial |  |
| [62] | 2020 | RF | Ancho | CTU [38] |  |  | RL profundo | Detección |  |
| ✗ | | Basado en RL profunda |  |
| [63] | 2021 |  | y Profundo | BOTNET [40] |  |  | ataca | formación contradictoria |  |
| [64] | 2021 | J48 DT, RF | ✗ | EclipseIoT [65] | ✗ | | Alteración de algunos | ✓ |  |
| [66] | 2021 | NB, SVM |  | CSE-CIC-IDS2018 [67] |  | FGSM | características seleccionadas |  |  |
| LSTM, CNN, | ✗ |  | ✗ | ✓ |  |
| [68] | 2021 | GRU | LSTM, CNN, | CSE-CIC-IDS2018 [67] |  | FGSM |  |  |  |
| ✗ |  | ✗ | ✓ |  |
| [69] | 2021 |  | GRU | CSE-CIC-IDS2018 [67] |  | FGSM, BIM |  |  |  |
| ✗ | DNN |  | ✗ | ✓ |  |
|  |  |  |  |  | JSMA, DeepFool | |  |  |  |
| [70] | 2022 | ✗ | MPL, LSTM, | CSE-CIC-IDS2018 [67] |  | Nes, Límite | ✗ | ✓ |  |
|  |  |  | CNN |  | HopSkipJump, | |  |  |  |
|  |  |  |  |  | puntualmente, | |  |  |  |
| [71] | 2022 | RF | MLP | CIC-IDS2017 [67], | Opt-attack | | Adaptativo |  |  |
| ✗ | | ✗ |  |
|  |  |  |  | IoT-23 [72] |  |  | perturbación |  |  |
| [73] | 2022 |  | LSTM, RNN | MedBIoT [74] y |  |  | método del patrón |  |  |
| ✗ | ✗ | | Alterar | ✗ |  |
|  |  |  |  | Conjunto de datos IoTID [75]) |  |  | características |  |  |
|  |  |  |  | [75]. |  |  |  | Datos gaussianos |  |
| Este | - | Adaboost | DNN | NSL-KDD | GN,ZOO,BIM | | ✗ |  |
| papel |  | Embolsado |  | UNSW-NB15 | FGSM, JSMA, | |  | aumento |  |
|  |  | DT, GB |  |  | C&W, PGD | |  |  |  |
|  |  | LR, RF |  |  | BA, HSJ | |  |  |  |
|  |  | VPC |  |  |  |  |  |  |  |

6

+

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*



**Fig. 4.** Protocolo de evaluación.

* **Tasa de falsos negativos**: es un indicador más específico que destaca el porcentaje de tráfico malicioso que ha superado con éxito el

IDS. =

Donde **TP son** los Verdaderos Positivos y representan el número de registros anómalos que se identifican correctamente como anomalías. Los **TN son los Verdaderos Negativos** y calculan el número de registros normales que se identifican correctamente como normales. Los **FP son los** Falsos Positivos y representan el número de registros normales clasificados erróneamente como anómalos. Los **TN son los Verdaderos Negativos y representan el número de registros** anómalos que se identifican como normales.

Un buen clasificador es aquel que tiene una *alta precisión y una baja tasa de falsos negativos.*

*4.4. Protocolo de evaluación*

*4.4.1.* ***Evaluación inicial de la robustez***

Este escenario (véase la Fig. 4(a)) evalúa el rendimiento de los clasificadores frente a ataques de adversarios.

Para generar muestras adversarias utilizando ataques de caja blanca, las muestras de los conjuntos de datos de prueba (NSL-KDD Test+ y UNSW\_NB15\_testing-set) se perturban utilizando ataques de caja gris/negra aplicados *directamente al clasificador* y ataques de caja blanca aplicados *al modelo ''Generador''*. Las muestras adversarias generadas forman nuevos conjuntos de datos de prueba, que denominamos '**'Conjuntos de datos** adversarios**''** (un conjunto de datos adversarios generado a partir de NSL-KDD Test+ y otro generado a partir de UNSW\_NB15\_testing-set). El rendimiento de los clasificadores se evalúa en los siguientes escenarios:

* *Hipótesis 1: medición de los resultados en la hipótesis de referencia*

**- Tren:** NSL-KDD Tren+ / UNSW\_NB15\_training-set

**- Test:** Prueba NSL-KDD+ / UNSW\_NB15\_testing-set

* *Escenario 2: medir el rendimiento en un entorno adverso*

**- Tren**: NSL-KDD Tren+ / UNSW\_NB15\_training-set

**- Test:** Conjuntos de datos adversarios

Las primeras sub-tablas de las Tablas 4 y 5 del apéndice muestran los resultados de la *precisión,* mientras que las segundas sub-tablas muestran los resultados de la *Tasa de Falsos Negativos*.

Para cada tabla, la primera fila muestra los resultados del primer escenario, mientras que las demás filas ilustran los resultados del segundo escenario. La diferencia de rendimiento entre el escenario base y el adversario se resalta **en rojo** para el aumento y en **amarillo** para la disminución, para cada clasificador. Por ejemplo, si la precisión de un clasificador es del 50% en el escenario base, si su rendimiento disminuye al 47% en el escenario adverso, el 3% se resalta en amarillo, si su rendimiento aumenta al 55% en el escenario adverso, el 5% se resalta en rojo.

*4.4.2.* ***Aplicación de técnicas de defensa***

En este escenario (ver Fig. 4(b)), medimos la contribución del aumento de datos gaussianos a la mejora de la robustez de los clasificadores NIDS. Por lo tanto, se generan dos nuevos conjuntos de datos denominados ''conjuntos de datos aumentados'' aplicando el aumento de datos gaussiano en KDD Train+ y UNSW\_NB15\_training-set. A continuación, se evalúa el rendimiento de los clasificadores en los dos escenarios siguientes:

* *Escenario 3: medir el rendimiento en un entorno no adversarial*

*con entrenamiento en datos aumentados* **Train:** Conjuntos de datos aumentados, Test**:** Prueba NSL-KDD+ / UNSW\_NB15\_testing-set

* *Escenario 4: medición del impacto de las técnicas de defensa en adver-*

*sarial* **Train:** Conjuntos de datos aumentados, **Test:** Conjuntos de datos adversarios

Los resultados obtenidos se describen en la tercera y cuarta subtablas de las Tablas 4 y 5, para la precisión y el FNR respectivamente. Para cada subtabla, la primera fila muestra los resultados del escenario 3, y las demás filas ilustran los resultados del escenario 4.

7

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*

**Cuadro 2**

Resumen de los resultados de la hipótesis de evaluación 2.

Comportamiento frente a un ataque de ruido gaussiano

* DT y GB son muy vulnerables en comparación con otros clasificadores
* Para GB, los ataques han provocado un aumento de las falsas alarmas.
* Adaboost es el más robusto aunque tiene el peor rendimiento en el escenario de referencia.
* Los clasificadores clasifican más tráfico malicioso como legítimo en UNSW-NB15 que en NSL-KDD (altas tasas de falsos negativos), especialmente para Bagging y Random Forest.

Comportamiento frente a ataques Gray/black-box

* El rendimiento de los clasificadores cae drásticamente ante los ataques HopSkipJump y Boundary
* Los clasificadores son bastante robustos al ataque ZOO.
* DNN, con el mejor rendimiento de referencia, es el más vulnerable al ataque ZOO.
* Adaboost es robusto frente a tres ataques gray/back-box en la NSL-KDD, pero extremadamente sensible a los ataques Boundary y HopSkipJum en la UNSW-NB15.
* Desde el punto de vista de la precisión, los clasificadores tienen el mismo comportamiento frente a los ataques en ambas bases de datos. Sin embargo, el comportamiento de FNR es diferente de una base de datos a otra: aumenta para la mayoría de los clasificadores en la base de datos NSL-KDD, pero sigue siendo muy bajo para algunos clasificadores (DNN, LR, RF SVC) en la base de datos UNSW-NB15.

Comportamiento frente a ataques de caja blanca

* Los clasificadores son bastante robustos al ataque de FGSM, PGD, BIM y C&W también tiene un efecto muy similar.
* JSMA es el ataque más poderoso.
* Adaboost y RF son los únicos clasificadores resistentes a todos los ataques de caja blanca.
* Los clasificadores fueron más vulnerables a estos ataques en la base de datos UNSW-NB15 que en la base de datos NSL-KDD, tanto para la precisión como para la tasa de falsos negativos.

**5. 5. Resultados experimentales**

*5.1. Evaluación inicial de la robustez*

*5.1.1. Escenario 1: medición del rendimiento en el escenario de referencia* Los resultados de las primeras filas de las primeras subtablas muestran que casi

Todos los clasificadores funcionan bien, la precisión varía entre el 74% y el 77% en la NSL-KDD, excepto Adaboost, que tiene el peor rendimiento (precisión del 55,13% en la NSL-KDD y del 60,6% en la UNSW-NB15). DNN es el más eficiente con una precisión del 77,68% en NSL-KDD y del 78,56% en UNSW-NB15. En el conjunto de datos NSL-KDD, estos resultados se ven confirmados por las tasas de falsos negativos (primer crudo de la segunda subtabla). En efecto, Adaboost clasifica erróneamente como benigno el 71,35% del tráfico malicioso, mientras que los demás clasificadores tienen un FNR que varía entre el 29,68% y el 39,32%. Sin embargo, todos los clasificadores tienen tasas de falsos negativos extremadamente bajas (por debajo del 6%) en la base de datos UNSW-NB15.

*5.1.2. Escenario 2: medición del rendimiento en un entorno adverso* En este escenario interesan dos análisis: (i) medir el impacto de *cada ataque adverso* sobre los distintos clasificadores y (ii) medir el rendimiento de *cada clasificador* frente a distintos tipos de ataques. El primer análisis identifica el clasificador más/menos vulnerable a un ataque adversario dado, mientras que el segundo encuentra el más/menos potente.

ataque adversario para cada clasificador.

*5.1.2.1.* ***Precisión****.*

*Ataque de ruido gaussiano:*

* Se observan dos comportamientos diferentes; algunos clasificadores (DT, GB y bagging en UNSW-NB15) son muy vulnerables al ataque del ruido gaussiano, incluso a baja intensidad ( = 0*,01*), mientras que los demás clasificadores son más robustos. En particular, para = 0*,01, el* rendimiento de SVC (resp. regresión logística) permanece estable en NSL-KDD, (resp. UNSW-NB15).
* Para un ataque de alta intensidad ( = 0*,*2), el comportamiento es similar, donde, en el conjunto de datos NSL-KDD, DT pierde casi la mitad de su rendimiento y GB cae a un tercio de su eficacia. Sin embargo, la disminución de la precisión de los demás algoritmos oscila entre el 4,7% de LR y el 11,5% de Bagging. Los resultados son similares en la base de datos UNSW-NB15.
* Sorprendentemente, Adaboost, que tiene el peor rendimiento de referencia, es el menos vulnerable a los ataques de ruido gaussiano en el conjunto de datos NSL-KDD, superando a DT y GB cuando todos ellos se enfrentan a este tipo de ataque. Y lo que es más sorprendente, una pequeña perturbación de ruido gaussiano incluso *aumenta el* rendimiento de Adaboost en un 5,6%.

*Ataques de caja gris/negra:*

* En comparación con el ataque de ruido gaussiano, todos los clasificadores son bastante robustos al ataque ZOO, con una pérdida de rendimiento que oscila entre el 0,3% de Adaboost y el 5% de DNN en NSL-KDD. Así, el DNN con el mejor rendimiento de referencia es el más vulnerable al ataque ZOO.
* Los ataques HopSkipJump y Boundary tienen casi el mismo impacto en todos los clasificadores (cuyo rendimiento cae drásticamente). Esto puede explicarse por el hecho de que ambos ataques son de la misma familia, es decir, HopSkipJump es una extensión del ataque Boundary.
* A excepción de Adaboost, cuya precisión disminuye sólo un 21% en NSL-KDD para ambos ataques, los demás clasificadores son mucho más vulnerables. De hecho, en NSL-KDD, DNN y RF pierden un 71% y un 68% de su rendimiento respectivamente cuando se enfrentan al ataque Boundary. En cuanto al ataque HopSkipJUmp, tanto Bagging como DT pierden casi el 68% de su rendimiento.
* El comportamiento de la mayoría de los clasificadores frente a estos ataques es similar en ambas bases de datos.

*Ataques de caja blanca:*

* FGSM, PGD y BIM, que son ataques de la misma familia, tienen casi el mismo impacto en los clasificadores.
* JSMA es el ataque más potente; en NSL-KDD, la disminución de la precisión alcanza el 36% para GB. C&W es ligeramente más potente que el trío FGSM, PGD, BIM, pero todos ellos son ataques débiles que provocan una caída de la precisión no superior al 4,7%.
* Adaboost y Random Forest son los únicos clasificadores resistentes a todos los ataques de caja blanca. La disminución de la precisión se limita al 1,4% y al 3,7% para Adaboost y RF respectivamente en NSL-KDD.
* Los clasificadores son más vulnerables a estos ataques en la base de datos UNSW-NB15 que en la base de datos NSL-KDD.

*5.1.2.2.* ***Tasa de falsos negativos****.* Los resultados se muestran en las segundas subtablas de las Tablas 4 y 5. El objetivo de esta evaluación es medir la capacidad de los clasificadores para bloquear el tráfico malicioso. Recordemos que el FNR mide el porcentaje de tráfico malicioso clasificado como legítimo, por lo que cuanto menor sea el FNR, mejor será el rendimiento del clasificador.

8

*H. Jmila y M.I. Khedher*

*Ataque de ruido gaussiano:*

* Los resultados muestran que FNR puede disminuir tras la perturbación de los datos, es decir, los datos maliciosos clasificados inicialmente como ilegítimos se clasifican erróneamente como legítimos tras la perturbación por ruido gaussiano. Esto refleja una mejora en el rendimiento del clasificador, como ocurre con Adaboost, GB, LR y SVC en el conjunto de datos NSL-KDD. En cuanto a los clasificadores Bagging, DT, DNN y RF, el FNR aumenta un 23%, 13,2%, 22% y 11% para a = 0*,*02, lo que refleja la disminución general de la precisión descrita en la primera subtabla de la Tabla 4.
* El resultado de GB en el conjunto de datos NSL-KDD es especialmente interesante porque el ruido gaussiano tiene un impacto contradictorio en la precisión y el FNR. De hecho, el rendimiento general del clasificador se degradó (la precisión disminuyó hasta un 21,68% para = 0*,2*), mientras que el FNR también disminuyó (un 25,6% para = 0*,*2), lo que significa que se bloqueó más tráfico malicioso. Así pues, la degradación del rendimiento general del clasificador puede deberse a la clasificación errónea del tráfico benigno como ilegítimo, lo que provoca *falsas alarmas*.
* Observamos que en la base de datos UNSW-NB15, en la que todos los clasificadores tenían bajos índices de falsos negativos en ausencia de ataques adversarios, los índices de falsos negativos aumentan significativamente, lo que demuestra su sensibilidad a estos ataques, especialmente en el caso de los clasificadores Bagging y RF.

*Ataques de caja gris/negra:*

* Adaboost es robusto frente a los tres ataques gray/back-box en el conjunto de datos NSL-KDD. Su aumento de FNR no supera el 3,1% (en el caso del ataque HopSkipJump). Sin embargo, en UNSW-NB15, el FNR aumenta exponencialmente contra Boundary y HopSkipJum y alcanza el 98,23%.
* Los ataques HopSkipJump y Boundary tienen el mismo impacto en los demás clasificadores. De hecho, el FNR alcanza el 100%, lo que significa que el ruido adverso consigue clasificar erróneamente todas las muestras maliciosas en benignas.
* HopSkipJump y Boundary son más potentes que ZooAttack en la mayoría de los casos.
* A diferencia de NSL-KDD, algunos algoritmos (DNN, LR, RF, SVC) son muy robustos a estos ataques en el UNSW-NB15 desde el punto de vista del FNR, que no aumenta.

*Ataques de caja blanca:*

* en el conjunto de datos NSL-KDD, JSMA tiene la mayor influencia en la desviación de FNR, ya sea positiva o negativa. Se observa principalmente que el FNR de DT aumenta un 41% para DT y un 41% y un 10% para Adaboost y LG respectivamente.
* Aunque los resultados no difieren significativamente de los resultados de referencia para FGSM, PDG y BIM en el conjunto de datos NSL-KDD, observamos que DNN es el clasificador más afectado (un aumento del FNR de hasta el 8,7%), mientras que el FNR de Adaboost y Gradient Boosting disminuye ligeramente, lo que refleja una mejor clasificación del tráfico malicioso.
* El impacto de estos ataques en el FNR de los clasificadores es mucho más notable en el conjunto de datos UNSW-NB15, que aumenta exponencialmente para la mayoría de los clasificadores excepto para Adabooost, que mantiene un FNR bajo.

*5.2. Aplicación de técnicas de defensa*

*5.2.1. Escenario 3: medición del rendimiento en un entorno no adversarial con entrenamiento en datos aumentados*

Comparando el rendimiento de los clasificadores entrenados en la base de datos aumentada en un entorno no adversarial (primeras filas de las Tablas 4 y 5) con su rendimiento cuando se entrenaron en conjuntos de datos de entrenamiento iniciales (primera fila de las subtablas 3), observamos que el rendimiento de la mayoría de ellos (Bagging, DT, DNN, RF) ha permanecido estable en el conjunto de datos NSL-KDD. El rendimiento de Adaboost aumentó del 55,13% al 66,33%,

*Redes informáticas 214 (2022) 109073*

pero el rendimiento de LR y SVC disminuyó del 75% al 67,27% y del 74,29% al 65,36%, respectivamente. Los resultados FNR (primera fila de la Tabla 4) reflejan las mismas conclusiones. La degradación del rendimiento es más notable en la base UNSW-NB15.

*5.2.2. Escenario 4: medición del impacto de las técnicas de defensa*

*5.2.2.1.* ***Precisión****.* Los resultados se muestran en las terceras sub-tablas y se compararán con los resultados de las primeras sub-tablas, donde se midió la precisión sin aplicar técnicas de defensa.

* *Ataque de ruido gaussiano:* La técnica de defensa ha mejorado la robustez de la mayoría de los clasificadores. La mejora es aún más notable para una perturbación gaussiana de alta intensidad ( = 0*,*2). La mejora es casi perfecta para SVC y LR que se han vuelto más robustos pero a costa de una degradación en el rendimiento, incluso en condiciones no adversariales. Lo más interesante es que el DNN se ha vuelto más robusto manteniendo casi el mismo rendimiento original en el conjunto de datos NSL-KDD, es el clasificador que tiene la mejor precisión en condiciones normales y adversariales en el conjunto de datos NSL-KDD.
* *Ataques de caja gris/negra:* La técnica de defensa tiene efectos diferentes según el clasificador. Por ejemplo, en el conjunto de datos NSL-KDD:
  1. contribuye a la mejora de la robustez para algunos clasificadores como para LR y SVC cuya precisión disminuyó, en comparación con el rendimiento de referencia, en sólo un 55,3% y un 53,8% (resp.) cuando se enfrentaron a un ataque Boundary, frente a un 67% y un 66,8% (resp.) de disminución antes de haber aplicado la técnica de defensa.
  2. degrada la robustez de algunos algoritmos como es el caso de Adaboost cuya precisión disminuye un 53% para un ataque Boundary en lugar de un 22% sin defensa. (iii) la técnica de defensa tiene muy poco efecto en los demás clasificadores que mantienen casi el mismo nivel de robustez que antes. El comportamiento de los clasificadores es similar en ambas bases de datos.
* *Ataques de caja blanca:* La técnica de defensa es eficaz para casi todos los clasificadores frente a FGSM, PGD y BIM. En cuanto a JSMA, la robustez también mejora para todos los clasificadores, y Gradient Boosting tiene el mejor rendimiento. El comportamiento de los clasificadores es similar en ambas bases de datos.

*5.2.2.2.* ***Tasa de falsos negativos*** *.* Los resultados se muestran en la cuarta subtabla, y se compararán con los de la segunda subtabla, donde se mide el FNR sin técnica de defensa.

* *Ataque de ruido gaussiano:* El aumento del FNR es menos importante con la técnica de defensa para la mayoría de los algoritmos, por lo que su robustez ha mejorado. Sin embargo, la capacidad de estos clasificadores para bloquear el tráfico malicioso ha disminuido en general, por ejemplo para SVC, el FNR es del 50% mientras que no superaba el 37,52% sin defensa en NSL-KDD. Además, la técnica de defensa no mejoró el rendimiento de Gradient Boosting y RF en NSL-KDD, ya que su FNR aumentó del 10,2% al 48,51% y del 45,07% al 55,51% respectivamente, por lo que se bloqueó más tráfico malicioso sin defensa. DNN tiene el FNR más bajo (la disminución del FNR no supera el 2,8%) y una robustez muy buena. La técnica de defensa ha mejorado su robustez.Curiosamente, el método de defensa disminuyó la robustez de algunos algoritmos a los ataques de ruido gaussiano en UNSW-NB15 (FNR aumenta más) como es el caso de DT, Gradient Boost y Adaboost.
* *Ataques de caja gris/negra:* en el conjunto de datos NSL-KDD, la defensa ha degradado la robustez de Adaboost frente a Boundary y Hop-SkipJump (el FNR aumenta un 49%, mientras que el aumento se limitaba al 3,1% antes de la defensa). Ha mejorado ligeramente la robustez de Bagging, (DNN,RF). La mejora es más significativa para LR y SVC. FNR aumentó sólo un 52% y un 49% para LR y SVC respectivamente, frente a un aumento del 62% para ambos sin defensa. En el conjunto de datos UNSW-NB15, el método de defensa demostró ser eficaz y mejoró significativamente la robustez de los clasificadores en términos de FNR.

9

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*

**Cuadro 3**

Resumen de los resultados de la hipótesis de evaluación 4.

Comportamiento frente a un ataque de ruido gaussiano

* La técnica de defensa ha mejorado la robustez de la mayoría de los clasificadores.
* La robustez de SVC y LR ha mejorado, pero a costa de una degradación del rendimiento.
* La robustez de la DNN se ha mejorado en el conjunto de datos NSL-KDD manteniendo su buen rendimiento, por lo que la DNN tiene la mejor precisión en condiciones normales y adversariales.
* El método de defensa disminuyó la robustez de algunos algoritmos en el UNSW-NB15 (FNR aumenta más) como es el caso de DT, Gradient Boost y Adaboost.

Comportamiento frente a ataques Gray/black-box

* La robustez de Bagging, DT, LR, SVC ha mejorado en el conjunto de datos NSL-KDD
* La robustez de Adaboost ha disminuido en el conjunto de datos NSL-KDD
* La técnica de defensa casi no tiene efecto sobre los demás clasificadores. En términos de FNR, el método de defensa fue más eficaz en la base UNSW-NB15, que NSL KDD (las tasas de FNR son más bajas).

Comportamiento frente a ataques de caja blanca

* La robustez de los clasificadores frente a los ataques FGSM, PGD, BIM es prácticamente la misma, puesto que ya son bastante robustos.
* La robustez frente a JSMA ha mejorado para la mayoría de los clasificadores, pero a costa de una degradación del rendimiento para la mayoría de ellos. En términos de FNR, el método de defensa no fue eficaz en UNSW-NB15 e incluso degradó el rendimiento de algunos clasificadores (por ejemplo, Gradient Boost, DT).
* *Ataques de caja blanca:* Después de la defensa, la robustez de los clasificadores para FGSM, PGD, BIM y C&W es estable en NSL-KDD, ya que los clasificadores ya son bastante robustos. Pero la mejora de la robustez es más visible para JSMA, aunque el rendimiento global se ha degradado para la mayoría de los clasificadores (mayor FNR, en comparación con la Tabla 2). El método de defensa no fue efectivo en UNSW-NB15 e incluso degradó el rendimiento de algunos clasificadores (por ejemplo, Gradient Boost, DT).

*5.3. Análisis y discusión*

En esta sección resumimos y analizamos nuestros resultados.

*5.3.1. Conclusiones generales*

* Un ataque no afecta a todos los clasificadores de la misma manera. Del mismo modo, la robustez de un clasificador depende de los ataques. En el mismo sentido, una técnica de defensa no es eficaz contra todos los ataques y no tiene el mismo efecto sobre los distintos clasificadores (puede mejorar o disminuir la robustez del clasificador o ser ineficaz). Del mismo modo, el comportamiento de un clasificador ante un ataque o un método de defensa depende de la base de datos.
* La robustez y el rendimiento general de los clasificadores pueden ser contradictorios. Como se observa en los resultados del conjunto de datos NSL-KDD, Ad-aboost es un clasificador muy robusto, pero no tiene una gran precisión.

Por el contrario,˙ DNN es muy eficaz pero es el más vulnerable al ataque ZOO. Por lo tanto, según la situación y la necesidad del IDS, se puede privilegiar la robustez o el rendimiento. A saber, si el IDS opera en un entorno determinado, es natural favorecer el rendimiento, sin embargo, si el entorno es incierto, la robustez cobra importancia. Del mismo modo, las técnicas de defensa pueden mejorar la robustez de los clasificadores, pero a costa de degradar su rendimiento. Por lo tanto, hay que considerar un compromiso entre estos dos objetivos. Una buena técnica de defensa, mejora la robustez del clasificador sin degradar su rendimiento. Además, la eficacia de un método de defensa contra un ataque a una base de datos puede variar de una base de datos a otra.

*5.3.2. Observaciones específicas*

* Los ataques de la misma familia (Boundary y HopSkipJump) tienen el mismo impacto en cada clasificador.
* A veces, un ataque puede tener el efecto contrario: el ataque de ruido gaussiano mejoró el rendimiento de Adaboost en NSL-KDD. Además de clasificar erróneamente más tráfico malicioso como legítimo, que es el principal objetivo de un ataque de IDS adversario, un ataque adversario también puede aumentar la tasa de falsas alarmas, como ocurrió con Gradient Boosting contra el ruido gaussiano.
* Los ataques Boundary y HopSkipJump consiguen clasificar erróneamente el 100% del tráfico malicioso en NSL-KDD.
* La defensa contra el aumento de datos gaussianos fue especialmente eficaz contra el ataque de ruido gaussiano, probablemente porque son de la misma familia.
* Un método de defensa puede incluso degradar la robustez de un clasificador (por ejemplo, DT contra ataques de caja blanca en el UNSW-NB15), por lo que debe elegirse adecuadamente.

También es interesante observar que, en nuestro experimento, los ataques gray/back-box son más eficaces que los ataques white-box, lo que es contrario a la intuición, ya que estos últimos tienen acceso a más información sobre el clasificador. Esto puede explicarse por el hecho de que los ataques gray/back-box se aplicaron directamente sobre los clasificadores, mientras que los ataques white-box se aplicaron sobre una red de generadores. Por otra parte, el rendimiento de los ataques depende en gran medida de la configuración de los hiperparámetros. Un mejor ajuste de los hiperparámetros probablemente mejoraría el rendimiento de los ataques de caja blanca.

*5.4. Comparación con el estado de la técnica*

Como se menciona en 3.4, el artículo [41] es el más cercano a nuestro trabajo ya que, al igual que nosotros, los autores evalúan la robustez de *los clasificadores superficiales* frente a *ataques adversarios del estado del arte* utilizando la base de datos NSL-KDD.

Dado que la información hiperparamétrica no está disponible en [41], además de la diferencia en la lista de clasificadores y ataques considerados, optamos por comparar cualitativamente nuestro marco de evaluación con el enfoque presentado en [41].

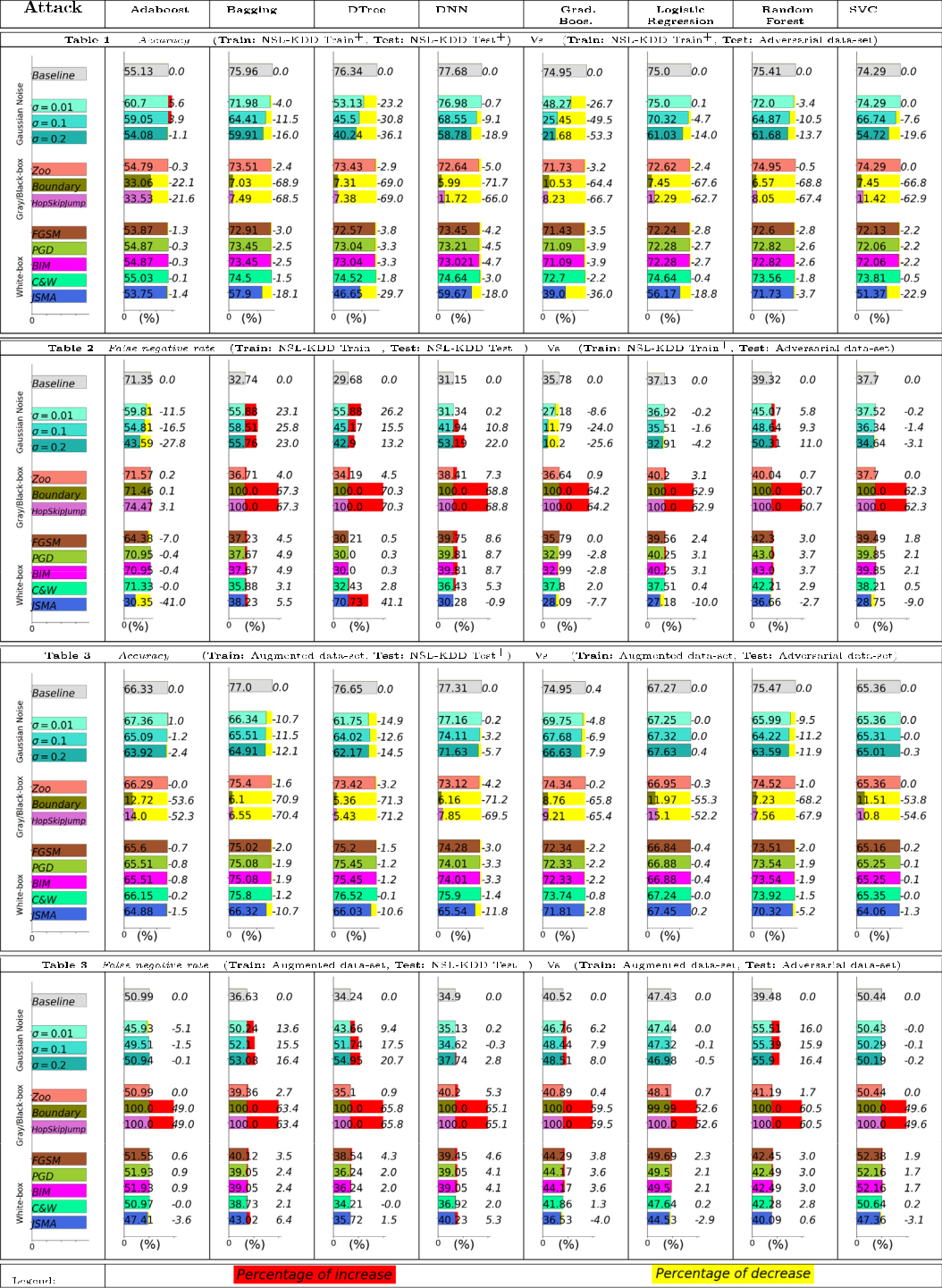
Los autores de [41] evalúan la robustez de 3 clasificadores superficiales (LR, RF, SVM) frente a tres ataques adversarios de caja blanca (FGSM, PDG, L-BFGS) y un ataque de caja negra SPSA, y no proponen un método de defensa (Cf. Tabla 1). Este trabajo, sin embargo, evalúa una colección más rica y variada de clasificadores ML (Adaboost, Bagging, DT, GB, LR, RF, SVC), contra ataques adversarios más variados y numerosos (ruido gaussiano, ataques de caja blanca: FGSM, PGD, BIM, C&W, JSMA, Ataques de caja gris/negra: Zoo, Boundary, HopSkipJump). Además, completamos nuestro estudio evaluando el impacto de una técnica de defensa (aumento de datos gaussianos) en la mejora de la robustez de los clasificadores. Esto nos permitió extraer conclusiones interesantes sobre el equilibrio entre robustez y precisión de los clasificadores, como se ha explicado anteriormente. Nótese, sin embargo, que los resultados obtenidos en [41] confirman nuestras conclusiones sobre la vulnerabilidad de los clasificadores ML a diferentes ataques con distintos grados de sensibilidad.

10

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*

**Cuadro 4**

Resultados de la evaluación NSL-KDD.



**6. Debate**

En este trabajo, los cambios sobre las muestras de entrada se realizan sobre el vector de características, sin embargo, un atacante no tiene acceso al vector de características de entrada del algoritmo ML para poder modificarlo, sino que debe generar tráfico real que respete las características descritas por el vector de características generado por el ataque adversario (FGSM, PGD, etc.). Este tráfico también debe conservar su funcionalidad original (maliciosa o benigna). Esta transición de vector de características a instancia real se conoce en la literatura como la transición de ''espacio de características'' a ''espacio de problemas'' [82],

es específico de determinados tipos de datos, en los que, a diferencia de las imágenes, esta transición no es trivialmente reversible y puede ser compleja.

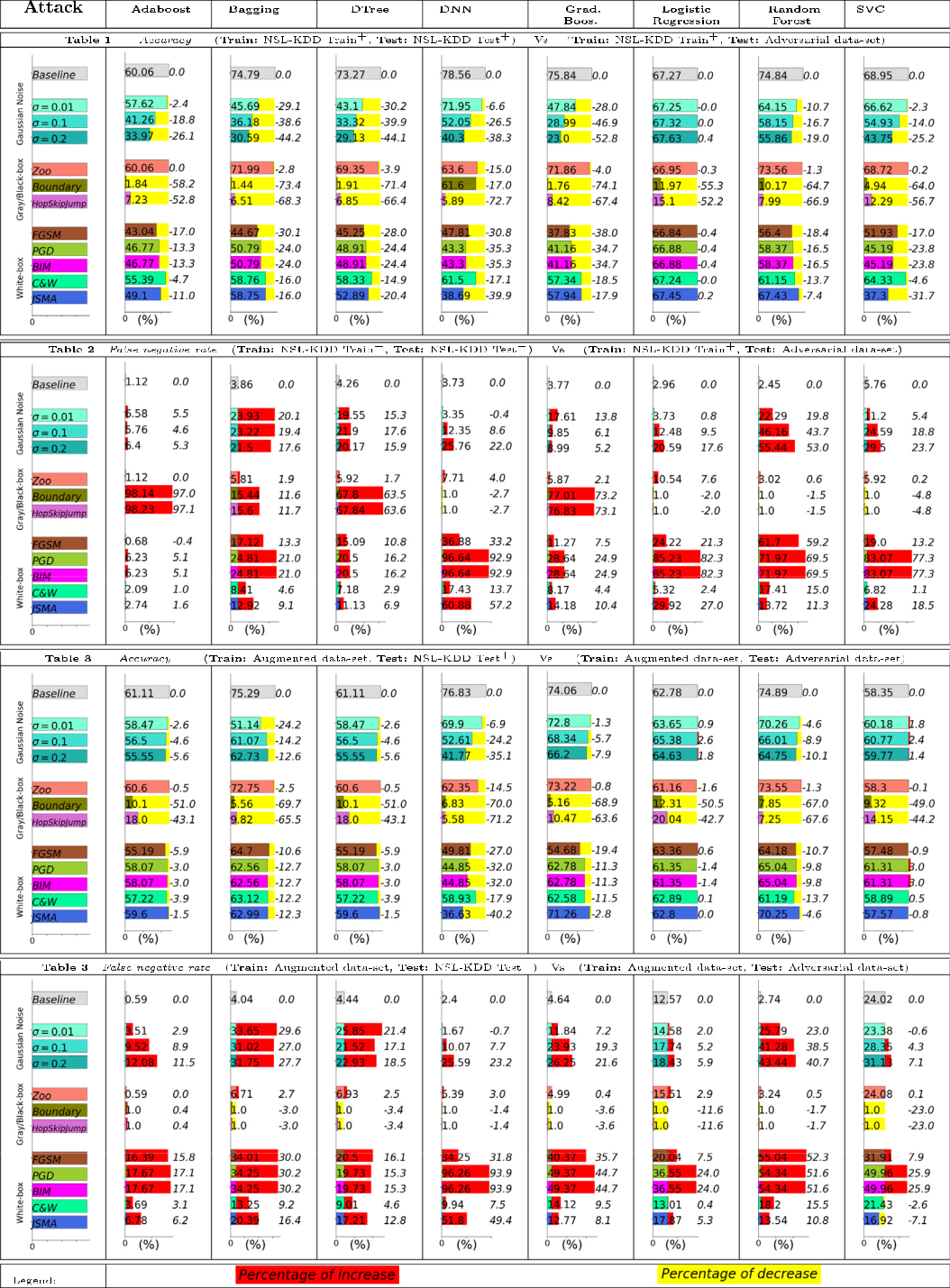
Una forma de facilitar esta transición consiste en realizar juiciosamente algunas modificaciones en la muestra de tráfico original, con el fin de obtener una muestra con características próximas a las de la muestra perturbada (muestra adversaria generada), sin alterar la función primaria del tráfico. Algunos ejemplos de manipulaciones posibles son (i) rellenar y fragmentar o duplicar las unidades de datos de protocolo (PDU, por sus siglas en inglés, p. ej., paquete, segmento, dato-grama, etc.) para modificar sus características volumétricas (p. ej., tamaño del flujo, número de paquetes, etc.), (ii) retrasar la transmisión de las PDU, para actuar sobre sus características temporales (p. ej., tiempo entre llegadas de los paquetes), (iii)

11

*H. Jmila y M.I. Khedher* *Redes informáticas 214 (2022) 109073*

**Cuadro 5**

Resultados de la evaluación UNSW-NB15.



modificar los valores de algunos campos, etc. Para no alterar la función principal del tráfico, las modificaciones deben hacerse sólo en los campos que no tengan impacto en esta función. Para ello, es necesario explorar y mejorar las herramientas de manipulación de PDU. Afortunadamente, ya existen herramientas prometedoras como Scapy [83], que es un programa de manipulación de paquetes. También se pueden explorar generadores de tráfico sintético personalizados [84].

**7. Conclusión y trabajo futuro**

Este artículo se centra en el área de investigación del aprendizaje automático adversarial. Estudiamos la robustez de varios clasificadores ML ampliamente utilizados

contra ejemplos adversos en el contexto de los IDS de red. Consideramos tanto los ataques de caja gris/negra como los de caja blanca. Se ha utilizado un clasificador externo basado en DNN para generar ejemplos adversarios basados en white-box. Además, estudiamos el impacto de una técnica de defensa basada en el aumento gaussiano de datos para mejorar la robustez de diferentes NIDS. Para la evaluación, consideramos tanto la precisión como la tasa de falsos negativos. Esta última mide el porcentaje de tráfico malicioso que consigue eludir el NIDS. Para la evaluación se utilizaron los conjuntos de datos NSL-KDD y UNSW-NB15. Los resultados muestran que los ataques no tienen el mismo impacto en todos los clasificadores y que la robustez de un clasificador depende del ataque. Del mismo modo, una técnica de defensa no es eficaz para todos los clasificadores ni contra todos los ataques. Además,

12

*H. Jmila y M.I. Khedher*

una técnica de defensa puede mejorar la robustez de un clasificador pero degradar su rendimiento general, por lo que debe considerarse un compromiso entre rendimiento y robustez en función del escenario de aplicación de la NIDS.

En futuros trabajos, pretendemos generar ataques adversarios más realistas que se proyecten más fácilmente en el espacio del problema. Para ello, seguiremos algunas recomendaciones encontradas en la literatura, [85-87], a saber (i) restringir el espacio de características a perturbar, es decir, evitar perturbar características no diferenciables para que la transformación sea reversible, y las características directamente relacionadas con la funcionalidad del flujo para no impactarlo, (ii) realizar perturbaciones de pequeña amplitud y comprobar que los valores de las características modificadas siguen siendo válidos (restricciones de dominio), y (iii) analizar la consistencia de los valores tomados por las características correlacionadas.

**Declaración de contribución a la autoría de CRediT**

**Houda Jmila:** Concepción y diseño del estudio, Adquisición de datos, Análisis y/o interpretación de datos, Redacción - borrador original, Redacción

- revisión y edición. **Mohamed Ibn Khedher:** Concepción y diseño del estudio, Adquisición de datos, Análisis y/o interpretación de datos, Redacción - borrador original, Redacción - revisión y edición.

**Agradecimientos**

Agradecemos a los revisores anónimos sus constructivos comentarios y sugerencias, que nos han ayudado a mejorar considerablemente la calidad de este trabajo. Todos los autores aprobaron la versión del manuscrito que se publicará.

**Anexo**

Véanse los cuadros 1-5.

**Referencias**

1. M.I. Khedher, M. Mziou-Sallami, M. Hadji, Improving decision-making-process for robot navigation under uncertainty, en: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1105-1113.
2. M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE symposium on computational intelligence for security and defense applications, Ieee, 2009, pp. 1-6.
3. N. Moustafa, J. Slay, UNSW-NB15: un amplio conjunto de datos para la red

sistemas de detección de intrusiones (conjunto de datos de red UNSW-NB15), en: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1-6.

1. N. Moustafa, J. Hu, J. Slay, A holistic review of network anomaly detection systems: A comprehensive survey, J. Netw. Comput. Appl. 128 (2019) 33-55.
2. A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2) (2015) 1153-1176.
3. S. Gamage, J. Samarabandu, Deep learning methods in network intrusion detection: A survey and an objective comparison, J. Netw. Comput. Appl. 169 (2020) 102767.
4. H. Jmila, M.I. Khedher, G. Blanc, M.A. El-Yacoubi, Siamese network based fea-ture learning for improved intrusion detection, en: Neural Information Processing - 26th International Conference, ICONIP 2019, 11953, 2019, pp. 377-389.
5. Y. Xu, Y. Zhou, P. Sekula, L. Ding, Machine learning in construction: From shallow to deep learning, Dev. Built Environ. 6 (2021) 100045.
6. H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, A. Jain, Adversarial attacks and defenses in images, graphs and text: A review, Int. J. Autom. Comput. 17 (2020) 151-178.
7. I. Rosenberg, A. Shabtai, Y. Elovici, L. Rokach, Adversarial learning in the cyber security domain, 2020, arXiv preprint arXiv:2007.02407.
8. N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Encuesta sobre el sistema de detección de intrusiones de red basado en SDN utilizando enfoques de aprendizaje automático, Peer-to-Peer Netw. Appl. 12 (2) (2019) 493-501.
9. M.I. Khedher, H. Ibn-Khedher, M. Hadji, Dynamic and scalable deep neural net-work verification algorithm, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1122-1130.
10. M. Mziou-Sallami, M.I. Khedher, A. Trabelsi, S. Kerboua-Benlarbi, D. Bettebghor, Safety and robustness of deep neural networks object recognition under generic attacks, en: Neural Information Processing - 26th International Conference, ICONIP, 1142, 2019, pp. 274-286.

*Redes informáticas 214 (2022) 109073*

1. H. Ibn-Khedher, M.I. Khedher, M. Hadji, Mathematical programming approach for adversarial attack modelling, en: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 343-350.
2. M.I. Khedher, M. Rezzoug, Analyzing adversarial attacks against deep learning for robot navigation, en: Proceedings of the 13th International Conference on Agents and Artificial Intelligence, ICAART, Volume 2, 2021, pp. 1114-1121.
3. D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, X. Yin, Evalu-ating and improving adversarial robustness of machine learning-based network intrusion detectors, 2020, arXiv: Cryptography and Security.
4. P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, Zoo: Zeroth order optimiza-tion based black-box attacks to deep neural networks without training substitute models, en: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 15-26.
5. I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, en: International Conference on Learning Representations, 2015, URL http://arxiv.org/abs/1412.6572.
6. A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, 2017, URL https://arxiv.org/abs/1611.01236.
7. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, en: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canadá, 30 de abril - 3 de mayo de 2018, Conference Track Proceedings, 2018, URL https://openreview.net/forum? id=rJzIBfZAb.
8. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, en: 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016, pp. 372-387, http://dx. doi.org/10.1109/EuroSP.2016.36.
9. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 86-94, http://dx.doi.org/10.1109/CVPR.2017.17.
10. S. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate

método para engañar a las redes neuronales profundas, en: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574-2582, http://dx.doi.org/ 10.1109/CVPR.2016.282.

1. N. Carlini, D. Wagner, Hacia la evaluación de la robustez de las redes neuronales, en: 2017 Ieee Symposium on Security and Privacy (Sp), IEEE, 2017, pp. 39-57.
2. W. Brendel, J. Rauber, M. Bethge, Ataques adversarios basados en decisiones: Reliable

ataques contra modelos de aprendizaje automático de caja negra, en: International Conference on Learning Representations, 2018, URL https://arxiv.org/abs/1712.04248.

1. J. Chen, M.I. Jordan, Boundary attack++: Query-efficient decision-based ad-versarial attack, 2019, CoRR arXiv:1904.02144, URL http://arxiv.org/abs/1904. 02144.
2. A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Ad-versarial attacks and defenses: A survey, 2018, CoRR arXiv:1810.00069, URL http://arxiv.org/abs/1810.00069.
3. K. Ren, T. Zheng, Z. Qin, X. Liu, Adversarial attacks and defenses in deep learning, Engineering 6 (3) (2020) 346-360.
4. M. Ozdag, Adversarial attacks and defenses against deep neural networks: a survey, Procedia Comput. Sci. 140 (2018) 152-161.
5. X. Wang, J. Li, X. Kuang, Y.-a. Tan, J. Li, The security of machine learning in an adversarial setting: A survey, J. Parallel Distrib. Comput. 130 (2019) 12-23.
6. N. Akhtar, A. Mian, Amenaza de ataques de adversarios en el aprendizaje profundo en visión por computador: A survey, Ieee Access 6 (2018) 14410-14430.
7. V. Zantedeschi, M.-I. Nicolae, A. Rawat, Efficient defenses against adversarial attacks, en: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, en: AISec '17, ACM, Nueva York, NY, Estados Unidos, 2017, pp. 39-49.
8. H. Lee, S. Han, J. Lee, Generative adversarial trainer: Defense to adversarial perturbations with gan, 2017, arXiv preprint arXiv:1705.03387.
9. L. Nguyen, S. Wang, A. Sinha, A learning and masking approach to secure learning, en: International Conference on Decision and Game Theory for Security, Springer, 2018, pp. 453-464.
10. K. Yang, J. Liu, C. Zhang, Y. Fang, Adversarial examples against the deep learning based network intrusion detection systems, en: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), 2018, pp. 559-564, http: //dx.doi.org/10.1109/MILCOM.2018.8599759.
11. Z. Wang, Detección de intrusiones basada en aprendizaje profundo con adversarios, IEEE Access 6 (2018) 38367-38384.
12. G. Apruzzese, M. Colajanni, M. Marchetti, Evaluación de la eficacia de los ataques de adversarios contra los detectores de botnets, en: 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA), IEEE, 2019, pp. 1-8.
13. S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100-123.
14. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Hacia la generación de un nuevo conjunto de datos de detección de intrusiones y caracterización del tráfico de intrusiones, en: ICISSp, 2018, pp. 108-116.
15. E.B. Beigi, H.H. Jazi, N. Stakhanova, A.A. Ghorbani, Towards effective feature selection in machine learning-based botnet detection approaches, en: 2014 IEEE Conference on Communications and Network Security, IEEE, 2014, pp. 247-255.
16. Y. Peng, J. Su, X. Shi, B. Zhao, Evaluating deep learning based network intrusion detection system in adversarial environment, en: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), 2019, pp. 61-66.

13

*H. Jmila y M.I. Khedher*

1. J. Clements, Y. Yang, A. Sharma, H. Hu, Y. Lao, Rallying adversarial techniques against deep learning for network security, 2019, arXiv preprint arXiv:1903. 11688.
2. Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: an ensemble of autoencoders for online network intrusion detection, 2018, arXiv preprint arXiv: 1802.09089.
3. O. Ibitoye, M.O. Shafiq, A. Matrawy, Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks, en: 2019 IEEE Global Communications Conference, GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019, IEEE, 2019, pp. 1-6.
4. N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779-796.
5. M.J. Hashemi, G. Cusack, E. Keller, Towards evaluation of nidss in adversarial setting, en: Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks, 2019, pp. 14-21.
6. H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, M. Qiu, Adversarial attacks against network intrusion detection in IoT systems, IEEE Internet Things J. (2020) 1.
7. B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, en: International Conference on Learning Representations, 2018.
8. H. Zenati, C.S. Foo, B. Lecouat, G. Manek, V.R. Chandrasekhar, Efficient gan-based anomaly detection, 2018, arXiv preprint arXiv:1802.06222.
9. J. Aiken, S. Scott-Hayward, Investigating adversarial attacks against network intrusion detection systems in SDNs, en: 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2019, pp. 1-7, http://dx.doi.org/10.1109/NFV-SDN47374.2019.9040101.
10. L. Labrotary, DARPA Intrusion Detection Evaluation Data Set, Vol. 12, Mas-sachusetts Institute of Technology, Cambridge, MA, 1999, p. 2009, consultado en enero.
11. S. Zhang, X. Xie, Y. Xu, A brute-force black-box method to attack machine learning-based systems in cybersecurity, IEEE Access 8 (2020) 128250-128263.
12. G. Creech, J. Hu, Generación de un nuevo conjunto de datos de prueba IDS: Time to retire the KDD

colección, en: 2013 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2013, pp. 4487-4492.

1. D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, C. Siemens, Drebin: Effective and explainable detection of android malware in your pocket, en: Ndss, Vol. 14, 2014, pp. 23-26.
2. J. Jeong, S. Kwon, M. Hong, J. Kwak, T. Shon, Adversarial attack-based security vulnerability verification using deep learning library for multimedia video surveillance, Multim. Tools Appl. 79 (23-24) (2020) 16077-16091.
3. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278-2324.
4. R.A. Khamis, A. Matrawy, Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs, 2020, arXiv:2007.04472.
5. R.A. Khamis, M.O. Shafiq, A. Matrawy, Investigating resistance of deep learning-based IDS against adversaries using min-max optimization, en: 2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020, IEEE, 2020, pp. 1-7.
6. M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, et al., Understanding the mirai botnet, en: 26th {USENIX} Security Symposium ({USENIX} Security 17), 2017, pp. 1093-1110.
7. A.U.H. Qureshi, H. Larijani, M. Yousefi, A. Adeel, N. Mtetwa, An adversarial approach for intrusion detection systems using Jacobian saliency map attacks (JSMA) algorithm, Computers 9 (3) (2020).
8. C. Zhang, X. Costa-Pérez, P. Patras, Tiki-Taka: atacar y defender en profundidad

learning-based intrusion detection systems, en: Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop, 2020, pp. 27-39.

1. G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, M. Colajanni, Deep reinforcement adversarial learning against botnet evasion attacks, IEEE Trans. Netw. Serv. Manag. 17 (4) (2020) 1975-1987.
2. A. Venturi, G. Apruzzese, M. Andreolini, M. Colajanni, M. Marchetti, Drelab - deep reinforcement learning adversarial botnet: A benchmark dataset for adversarial attacks against botnet intrusion detection systems, Data Brief 34 (2021) 106631.
3. D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, X. Yin, Evalu-ating and improving adversarial robustness of machine learning-based network intrusion detectors, IEEE J. Sel. Areas Commun. 39 (8) (2021) 2632-2647.
4. E. Anthi, S. Ahmad, O. Rana, G. Theodorakopoulos, P. Burnap, EclipseIoT: A secure and adaptive hub for the Internet of Things, Comput. Secur. 78 (2018) 477-490.
5. E. Anthi, L. Williams, A. Javed, P. Burnap, Hardening machine learning denial of service (DoS) defenses against adversarial attacks in IoT smart home networks, Comput. Secur. 108 (2021) 102352.
6. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108-116.
7. X. Fu, N. Zhou, L. Jiao, H. Li, J. Zhang, The robust deep learning-based schemes for intrusion detection in Internet of Things environments, Ann. Telecommun. 76 (5) (2021) 273-285.
8. J. Wang, J. Pan, I. AlQerm, Y. Liu, Def-IDS: An ensemble defense mechanism against adversarial attacks for deep learning-based network intrusion detection, en: 2021 International Conference on Computer Communications and Networks (ICCCN), IEEE, 2021, pp. 1-9.

*Redes informáticas 214 (2022) 109073*

1. C. Zhang, X. Costa-Pérez, P. Patras, Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms, IEEE/ACM Trans. Netw. (2022).
2. J. Vitorino, N. Oliveira, I. Praça, Adaptative perturbation patterns: Realistic adversarial learning for robust intrusion detection, Future Internet 14 (4) (2022) 108.
3. S. García, A. Parmisano, M.J. Erquiaga, IoT-23: A Labeled dataset with mali-cious and benign IoT network traffic, 2020, Más detalles aquí https://www. stratosphereips.org/datasets-iot23.
4. H. Jiang, J. Lin, H. Kang, Fgmd: A robust detector against adversarial attacks in the IoT network, Future Gener. Comput. Syst. 132 (2022) 194-210.
5. A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, S. Nõmm, MedbIoT: Gener-

ación de un conjunto de datos de botnet IoT en una red IoT de tamaño medio, en: ICISSP, 2020, pp. 207-218.

1. H. Kang, D.H. Ahn, G.M. Lee, J.D. Yoo, K.H. Park, H.K. Kim, IoT network intrusion dataset, 2019.
2. P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutor. 21 (1) (2018) 686-728.
3. Y. Dong, T. Pang, H. Su, J. Zhu, Evasión de defensas ante ejemplos adversariales transferibles mediante ataques invariantes de traducción, en: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4312-4321.
4. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, en: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265-283.
5. A. Gulli, S. Pal, Deep Learning with Keras, Packt Publishing Ltd, 2017.
6. A.C. Müller, S. Guido, Introducción al aprendizaje automático con Python: A Guide for Data Scientists, ''O'Reilly Media, Inc.'', 2016.
7. M.-I. Nicolae, M. Sinn, T.N. Minh, A. Rawat, M. Wistuba, V. Zantedeschi, I.M. Molloy, B. Edwards, Adversarial robustness toolbox v0.2.2, 2018, CoRR arXiv:1807.01069, URL http://arxiv.org/abs/1807.01069.
8. F. Pierazzi, F. Pendlebury, J. Cortellazzi, L. Cavallaro, Intriguing properties of adversarial ml attacks in the problem space, en: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 1332-1349.
9. R. Rohith, M. Moharir, G. Shobha, et al., Scapy-a powerful interactive packet ma-nipulation program, in: 2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS), IEEE, 2018, pp. 1-5.
10. O.A. Adeleke, N. Bastin, D. Gurkan, Network traffic generation: A survey and methodology, ACM Comput. Surv. 55 (2) (2022) 1-23.
11. N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, T. Hou, Manda: On adversarial ex-ample detection for network intrusion detection system, IEEE Trans. Dependable Secure Comput. (2022).
12. R. Sheatsley, N. Papernot, M.J. Weisman, G. Verma, P. McDaniel, Adversarial examples for network intrusion detection systems, J. Comput. Secur. (Preprint) (2022) 1-26.
13. M.A. Merzouk, F. Cuppens, N. Boulahia-Cuppens, R. Yaich, Investigating the practicality of adversarial evasion attacks on network intrusion detection, Ann. Telecommun. (2022) 1-13.

**Houda Jmila** se licenció en Ingeniería Informática en Telecom Sudparis, Instituto Politécnico de París (Francia) en 2011 y se doctoró en Telecomunicaciones e Informática en 2015 en la misma universidad. Actualmente es investigadora posdoctoral en Telecom Sud-Paris. Sus intereses de investigación incluyen la seguridad de redes y la gestión automatizada de recursos en redes virtuales, redes 5G y el ecosistema IoT, así como las aplicaciones de aprendizaje automático a estos dominios.



**Mohamed Ibn Khedher obtuvo** su título de ingeniero en 2007, y su máster en 2009, en Informática por la Escuela Nacional de Informática de Túnez. Obtuvo su doctorado en Informática en Telecom SudParis con la colaboración de la Universidad de Evry, Francia. Durante su doctorado, desarrolló un software para la reidentificación de personas a partir de secuencias de vídeo. Desde 2015, ha trabajado como ingeniero de software en un centro de investigación especializado en sistemas avanzados de asistencia al conductor. Actualmente, es ingeniero de investigación sénior en IRT systemX, Francia. Trabajó en el programa de investigación francés Grand defi "Trusted AI" como experto sénior en robustez de IA. Sus principales intereses son la verificación de redes neuronales, la robustez de la IA frente a adversarios, el aprendizaje automático, los estándares de codificación de vídeo, la videovigilancia, la biometría y el análisis de escritura.



14