UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Tesi di Laurea

UTILIZZO DI SOFT-BROWNIAN-OFFSET PER LA GENERAZIONE DI ATTACCHI AI FINI DELL'ADDESTRAMENTO DI RILEVATORI DI INTRUSIONI

APPLICATION OF SOFT-BROWNIAN-OFFSET TO GENERATE CYBER-ATTACKS TO TRAIN INTRUSION DETECTORS

GUGLIELMO BARTELLONI

Relatore: *Relatore*
Correlatore: *Correlatore*

Anno Accademico 2022-2023

INDICE

ELENCO DELLE FIGURE

"Inserire citazione"
— Inserire autore citazione

1

INTRODUZIONE

Un sistema di rilevamento di intrusioni, anche detto intrusion detection system (IDS), è un'applicazione o un dispositivo volta a monitorare continuamente la rete per identificare attività malevole.

In particolare un IDS controlla il traffico di rete oppure i log di sistema, alla ricerca di possibili anomalie che potrebbero indicare la presenza di attacchi.

Per migliorare il rilevamento delle intrusioni, nell'ultimo decennio, si è iniziato ad utilizzare algoritmi di Machine Learning e Deep Learning attingendo informazioni dai Big Data. [4]

Questo però ha portato a nuove problematiche, come ad esempio il fatto che i modelli di Machine Learning sono molto sensibili ai dati di addestramento.

Un dataset contenente pacchetti di rete sarà composto per la maggior parte da pacchetti "normali", cioè pacchetti di traffico abituale e, per la restante parte, da pacchetti di attacchi. I vari dataset sul traffico di rete presenti oggi però, non hanno una quantità sufficiente di attacchi per poter addestrare al meglio i modelli.

Un modello addestrato su un dataset con una percentuale di attacchi troppo bassa, non sarà in grado di rilevare bene gli attacchi [5].

Un metodo per sopperire a questa mancanza è quello di generare nuovi dati a partire da quelli già esistenti (Data Augmentation). I dati generati devono essere però sufficientemente differenti di modo da rappresentare meglio le anomalie che si hanno nella rete durante un attacco.

In questa tesi esploreremo un algoritmo di Data Augmentation chiamato Soft-Brownian-Offset (SBO) [6] e lo utilizzeremo per generare nuovi dati per addestrare un IDS.

In particolare Soft-Brownian-Offset permette di generare campioni così detti out-of-distribution (OOD), cioè campioni che non fanno parte della distribuzione dei dati di partenza. Questo è coerente col fatto che di solito i pacchetti di attacchi sono fuori dalla distribuzione dei pacchetti comuni.

2

FONDAMENTI

2.1 INTRUSION DETECTION SYSTEM

Un intrusione può essere definita come un evento che compromette integrità, confidenzialità e la disponibilità di un sistema informatico [7].

Gli Intrusion Detection System (IDS) sono delle soluzioni hardware o software che, posti all'interno di una rete o di un sistema, rilevano eventuali intrusioni.

Questi strumenti sono essenziali per tenere al sicuro le persone da attacchi informatici [1].

[8] Le principali funzioni degli IDS sono:

- Monitorare ed analizzare sia le attività utente che di sistema
- Tracciare le violazioni delle policy utente
- Analizzare le configurazioni e le vulnerabilità del sistema
- Rilevare tipici attacchi di rete
- Analizzare di attività anomale

[9] Solitamente gli IDS vengono classificati in base al tipo di analisi che effettuano e come questi rilevano le minacce. Ne esistono di tre tipi principali:

- Signature-Based Detection (SD)
- Anomaly-based Detection (AD)
- Stateful Protocol Analysis (SPA)

2.1.1 *Signature-Based Detection*

Questo tipo di rilevamento utilizza la firma di un attacco per poterlo rilevare. Quindi conoscendo questa firma, gli IDS la comparano agli eventi catturati della rete. Dato che questi attacchi hanno bisogno di una conoscenza pregressa sono anche chiamati Knowledge-based.

2.1.2 *Anomaly-based Detection*

Gli Anomaly-based Intrusion Detection Systems (AIDS) sono stati introdotti per sopperire alle mancanze del Signature-Based Detection. Questo tipo di rilevamento utilizza un modello che rappresenta il normale comportamento della rete. Se viene rilevato un evento che non è coerente con il modello di riferimento, allora viene segnalata un'anomalia. Questo tipo di rilevamento è chiamato anche Behavior-Based.

Il principale vantaggio di questo tipo di IDS è la possibilità di rilevare gli attacchi zero-day [10], in quanto questi sistemi, non si basano sulla firma dei dati o su regole rigide. Tra gli altri vantaggi si ha che risulta essere difficile, per un eventuale criminale, capire quale sia il comportamento normale di un utente senza produrre un segnale da parte del sistema [1].

[1] Gli AIDS possono essere classificati in base al metodo utilizzato per la loro implementazione:

- Basati sulla Statistica (Statistical-Based)
- Basati sulla Conoscenza (Knowledge-Based)
- Basati sull'apprendimento automatico (Machine Learning-Based)

Gli AIDS Statistical-Based dopo aver registrato i dati di una porzione di elementi, derivano il modello statistico di un utente nella rete.

I Knowledge-based invece, utilizzano regole predefinite per generare il modello di riferimento. Dall'altra parte troviamo gli AIDS Machine Learning-Based, che utilizzano un algoritmo di apprendimento automatico per generare il modello di riferimento.

La figura 1 mostra più in dettaglio questo tipo di classificazione e i metodi di implementazione associati.

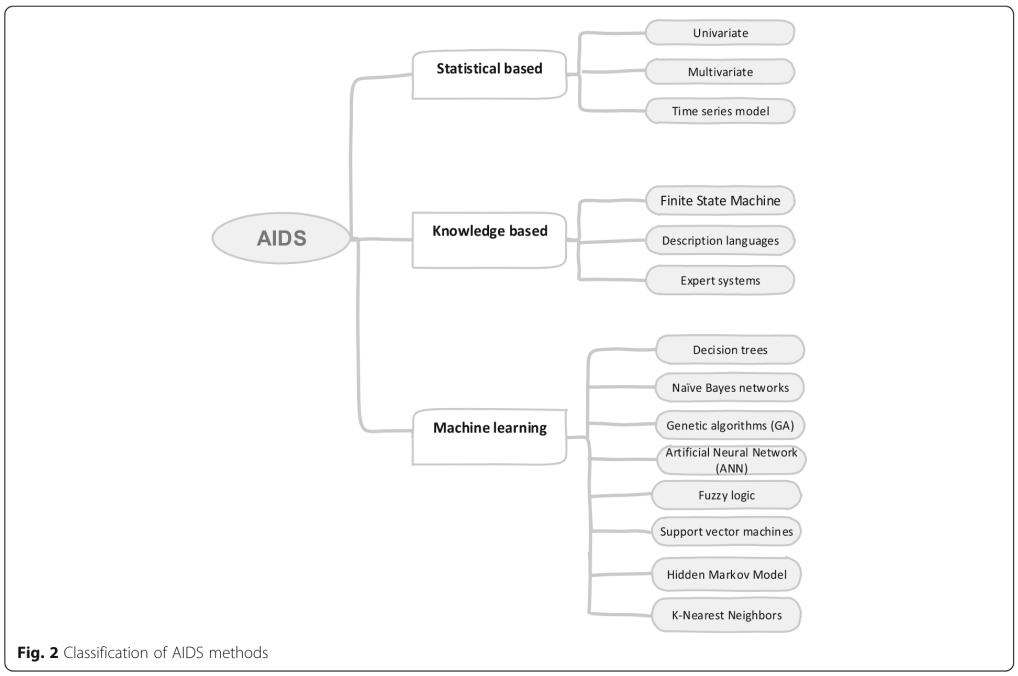

Fig. 2 Classification of AIDS methods

Figura 1: Schema che riassume i vari metodi di implementazione degli AIDS, da [1]. I modelli sono divisi nelle tre categorie descritte precedentemente, ma sono presenti anche i vari tipi di algoritmi utilizzati.

2.1.3 *Stateful-Protocol-Analysis*

In questo caso gli IDS conoscono lo stato e le specifiche del protocollo utilizzato. Vengono quindi rilevati degli eventi che non rispettano gli standard del protocollo, generalmente quelli da specifica e.g. IEEE.

Potrebbe sembrare che gli AD e gli SPA siano simili, in realtà i primi, conoscono il comportamento di una specifica rete, mentre i secondi, conoscono solo gli standard dei protocolli.

2.2 MACHINE LEARNING PER RILEVAMENTO DI INTRUSIONI

Il Machine Learning (ML) è un sottoinsieme dell'intelligenza artificiale e permette ai sistemi di imparare dai dati e di migliorare le loro prestazioni nel tempo senza essere esplicitamente programmati. Nel caso degli Intrusion Detection Systems, gli algoritmi di ML permettono di rilevare in maniera precisa e rapida gli attacchi per grandi quantità di dati in poco tempo [11].

Solitamente questi algoritmi sono divisi in:

- Supervised
- Unsupervised

Inoltre gli algoritmi di ML possono essere usati per la classificazione o per la predizione.

Il processo di classificazione si compone di vari passi tra cui:

- Preprocessamento dei dati
- Definizione dei dati di test
- Scelta dell'algoritmo
- Impostazione dei parametri
- Addestramento del modello
- Test del modello

come mostrato in figura 2.

Il modello viene addestrato su un dataset con classi conosciute e cerca di trovare una relazione tra le caratteristiche dei dati (features) e le classi stesse. In particolare un classificatore mappa una funzione:

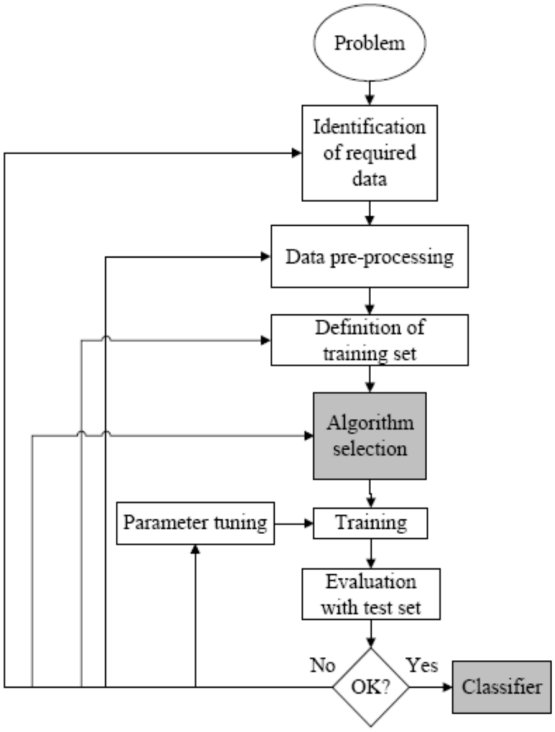

Figura 2: Il processo del Supervised Machine Learning, da [2]

$$f(x_1, \dots x_S) = \hat{y}$$

che assegna uno scalare facente parte di un insieme C di elementi disgiunti (le classi), ad un insieme di S elementi vettoriali (le caratteristiche). [12]

Nella regressione invece, si cerca di approssimare la funzione:

$$f(x_1, \dots x_S) = y$$

a partire dalle caratteristiche. La funzione f è approssimata utilizzando varie tecniche di interpolazione, estrapolazione, analisi di regressione e curve fitting [12].

Nel nostro caso utilizzeremo un algoritmo di classificazione.

Supervised Machine Learning

Il Supervised Machine Learning utilizza un dataset con classi completamente etichettate cercando di trovare una relazione tra gli elementi del dataset e le classi di questi dati. I dataset utilizzati per l'addestramento, necessitano di essere catalogati precedentemente. Ci sono vari algoritmi di questo tipo e.g. Support Vector Machine (SVM), Naïve Bayes, Reti Neurali, Alberi di Decisione, Random Forest.

Unsupervised Machine Learning

In questo caso, gli algoritmi di ML non hanno un dataset con classi etichettate. Questi algoritmi traovano delle relazioni tra i dati, cercando di raggrupparli in base a delle caratteristiche comuni.

L'unsupervised machine learning mostra però basse prestazioni per quanto riguarda il rilevamento quindi non possono essere utilizzati come principale strumento per gli IDS. Il motivo di questo scarso rendimento è dovuto al fatto che è molto probabile generare dei Falsi Positivi (vengono rilevati attacchi quando in realtà non cene sono) oppure generare dei Falsi Negativi (non vengono rilevati attacchi quando in realtà ce ne sono) questo va a degradare le performance generali del modello.

D'altra parte però questi modelli si sono dimostrati opinabilmente migliori per rilveare gli attacchi zero-day. [10].

2.3 XGBOOST

XGBoost (eXtreme Gradient Boosting) è un algoritmo di Machine Learning che utilizza alberi di decisione per la classificazione e la regressione.

Nel mondo reale è utilizzato anche la predizione dei click per le pubblicità [13] e come algoritmo per le competizioni su Kaggle [14].

Il suo punto di forza principale è la grande scalabilità in tutti gli scenari. XGBoost ha prestazioni superiori di dieci volte rispetto alle soluzioni esistenti. Inoltre riesce a sfruttare al meglio la potenza di calcolo della macchina su cui viene eseguito [14]. È stato scelto XGBoost come algoritmo in questa tesi per i motivi sopra citati.

2.3.1 *Alberi di Decisione*

Nella sua parte fondamentale XGBoost utilizza gli alberi di decisione.

Un albero di decisione è uno strumento relativamente semplice per classificare i dati dove vengono poste delle domande che riguardano le features. Ogni domanda è contenuta in un nodo dell'albero, ed oguno di questi punta ad un figlio che rappresenta una possibile risposta. In questo modo le domande formano una gerarchia (Figura 3), codificata come un albero [3].

L'algoritmo più conosciuto che utilizza alberi di decisione è il C4.5 [15].

2.4 DATASET

I dataset sono un insieme di dati utilizzati per l'addestramento e il test dei modelli di ML. In questa tesi sono stati utilizzati tre dataset: ADFA-NET, CIDDS e CIC-IDS18.

2.4.1 *ADFA-NET*

ADFA è un dataset creato da un gruppo di ricerca dell'Università di New South Wales (Australia) nel 2013 e include dieci tipi di attacchi. Contiene in particolare attacchi di Brute Force, Java Based Meterpreter, Add new Superuser, Linux Meterpreter payload e attacchi alla C100 Webshell. [16]

Questo è il più semplice dataset su cui sono stati effettuati i test.

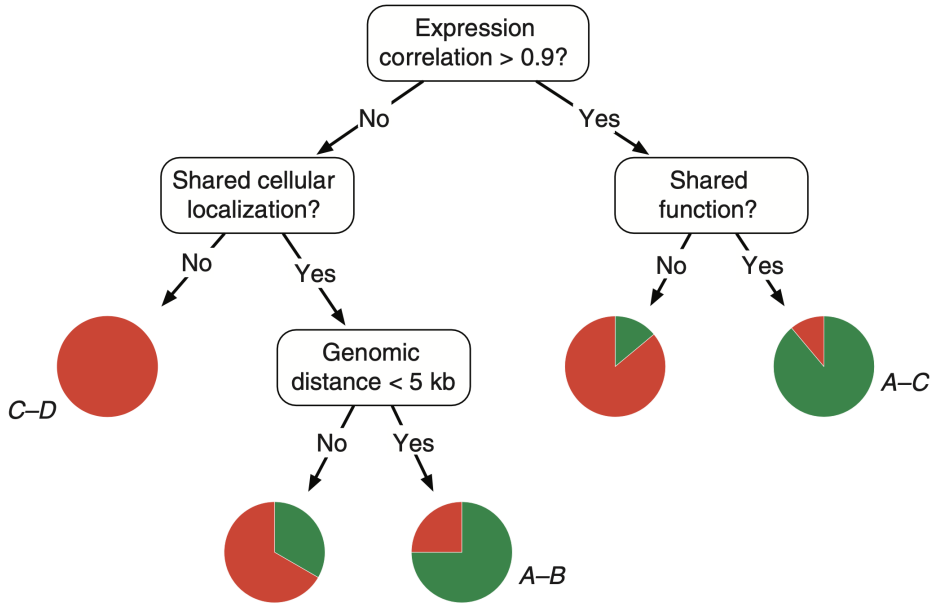

Figura 3: L'immagine mostra un esempio di albero di decisione di Carl Kingsford e Steven L Salzberg [3]

2.4.2 *CIDDS*

2.4.3 *CIC-IDS18*

Solitamnete i dataset non rappresentano al meglio la realtà, in quanto per motivi di privacy questi vengono offuscati e anonimizzati perdendo molta della diversificazione utile per l'addestramento di un modello. Per sopperire a queste mancanze è stato creato CIC-IDS cercando di mantenere il più possibile caratteristiche di uno scenario reale. Esso contiene una grande quantità di dati ricoprendo le principali tipologie di attacchi che si possono trovare in una rete, tra questi abbiamo DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltrazione, Port scan e Botnet [16].

Questo risulta di gran lunga il più complesso dataset utilizzato in questa tesi contenendo più di settanta colonne e quasi duecentomila righe.

3

SOFT BROWNIAN OFFSET

4

GENERAZIONE DI ATTACCHI USANDO SOFT BROWNIAN OFFSET

La procedura adottata per questa tesi è la seguente:

- Preparazione dei dataset
- Generazione dei pacchetti Out of Distribution (OOD)
- Analisi dei dati ottenuti
- Addestramento del modello
- Valutazione del modello

Nel capitolo 5 verranno mostrati i risultati ottenuti.

4.1 METODOLOGIE

Ogni dataset è stato analizzato, per capire quali fossero i dati più significativi per l'addestramento del modello.

È stato necessario filtrare alcune colonne che contenevano valori non numerici come, nel caso di CIC-IDS, la colonna "Timestamp" che includeva la data e l'ora del pacchetto.

Inoltre le righe contenenti valori come "Inf" o "NaN" sono state modificate perché, se lasciate intatte, avrebbero causato errori durante la generazione dei dati out-of-distribution.

Entrambi i dataset hanno una colonna "label" per indicare la tipologia del pacchetto, nello specifico, sono presenti le tipologie varie tipologie di attacchi e.g. "bruteForce", "dos", "pingScan", "portScan". Dato che, nel nostro caso, non ci interessa sapere nello specifico l'attacco del pacchetto, tutti gli attacchi, sono stati etichettati come "attack".

Dopo aver fatto queste operazione di preprocessamento, si è passati alla generazione dei sample OOD.

La prima prova effettuata è stata quella di utilizzare Soft Brownian Offset a partire dal dataset completo senza distinzioni di tipologia di pacchetto, per vedere se era possibile in questo modo, generare degli attacchi verosimili. Questa soluzione però genera dei pacchetti che sono troppo vicini, in termini di caratteristiche, a quelli normali perché i dataset hanno una quantità molto maggiore di dati normali rispetto a quelli di attacchi. Abbiamo quindi dovuto scartare questo metodo.

Un altro metodo possibile che non è stato approfondito è quello di generare i pacchetti come sopra, filtrando quelli troppo "vicini" (sempre in termini di caratteristiche) ai pacchetti normali. Si ottiene così un dataset che ha la giusta quantità di dati out-of-distribution e potrebbe migliorare l'apprendimento di un modello.

Il metodo che invece si è utilizzato e che ottiene buoni risultati è quello di utilizzare Soft Brownian Offset solo sui pacchetti di attacco, in questo modo si ottiene delle varianti dei pacchetti di attacco che riescono a migliorare il rilevamento da parte degli IDS.

Confronteremo inoltre i risultati con una generazione di pacchetti effettuata a partire da solo traffico normale evidenziando le differenze che si hanno con i due approcci.

Per il calcolo delle performance del modello XGBoost, a seguito dell'addestramento, si è utilizzato il coefficiente di Matthews.

5

RISULTATI

Di seguito verranno mostrati i più significativi riguardanti la tesi prodotti atraverso del codice Python utilizzando diverse librerie, tra cui:

- Pandas e Numpy, per la manipolazione dei dataset
- Plotly, per la visualizzazione dei dati
- UMAP, per la riduzione della dimensionalità dei dataset
- SBO, per la generazione dei dati fuori dalla distribuzione
- XGBoost, il modello addestrato
- Scikit-learn, per la valutazione del modello e per il preprocessamento dei dati

Tutto il codice utilizzato si trova su Github [17] .

5.1 PREPARAZIONE DEI DATI

Come detto nel capitolo 4, i dati i dati non possono essere utilizzati come sono ma hanno bisogno di un preprocessamento. In particolare, sono stati eliminati i dati non numerici attraverso il metodo di Pandas "pandas.DataFrame.replace" come segue:

```
input_data.replace([np.inf, -np.inf], -1, inplace=True)
input_data.replace(np.nan, -1, inplace=True)
```

Si estrae poi la colonna label dal dataset, che dovrà essere utilizzata successivamente per il modello, e si procede al one hot encoding dei dati attraverso il medoto "pandas.get_dummies":

```
input_data =pd.get_dummies(input_data)
```

Infine si cambia le etichette dei vari attacchi in "attack":

```
attacks_packets_types =['Bot', 'DDOS attack-HOIC', 'DDOS attack-LOIC-UDP',
                    'DoS attacks-Hulk', 'DoS attacks-SlowHTTPTest',
                                                    'FTP-BruteForce',
                    'Infilteration', 'SSH-Bruteforce']
attacks_packets.replace(attacks_packets_types, 'attack', inplace=True)
```

5.2 GENERAZIONE DEI PACCHETTI

Si è generato i nuovi pacchetti attraverso la libreria python SBO fornita da [6]. Il metodo "soft_brownian_offset" della libreria richiede:

- i dati da cui generare i nuovi pacchetti
- d_min (d^-)
- d_off (d^+)
- softness (σ)
- numero di pacchetti da generare

```
  data_ood =soft_brownian_offset(data_i, d_min_, d_off_,
                            softness=softness_,
                            n_samples=n_ood_samples)
```

5.3 ADDESTRAMENTO DEL MODELLO

5.4 CALCOLO DELLE METRICHE

6

CONCLUSIONI

BIBLIOGRAFIA

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, p. 20, July 2019. (Citato nelle pagine 3, 9, 10, and 11.)

[2] Babcock University, O. F.Y, A. J.E.T, A. O, H. J. O, O. O, and A. J, "Supervised Machine Learning Algorithms: Classification and Comparison," *International Journal of Computer Trends and Technology*, vol. 48, pp. 128–138, June 2017. (Citato nelle pagine 3 and 13.)

[3] C. Kingsford and S. L. Salzberg, "What are decision trees?," *Nature Biotechnology*, vol. 26, pp. 1011–1013, Sept. 2008. (Citato nelle pagine 3, 15, and 16.)

[4] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, Jan. 2021. (Citato a pagina 7.)

[5] S. S. Gopalan, D. Ravikumar, D. Linekar, A. Raza, and M. Hasib, "Balancing Approaches towards ML for IDS: A Survey for the CSE-CIC IDS Dataset," in *2020 International Conference on Communications, Signal Processing, and Their Applications (ICCSPA)*, (Sharjah, United Arab Emirates), pp. 1–6, IEEE, Mar. 2021. (Citato a pagina 7.)

[6] F. Moller, D. Botache, D. Huseljic, F. Heidecker, M. Bieshaar, and B. Sick, "Out-of-distribution Detection and Generation using Soft Brownian Offset Sampling and Autoencoders," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (Nashville, TN, USA), pp. 46–55, IEEE, June 2021. (Citato nelle pagine 7 and 22.)

[7] E. Biermann, E. Cloete, and L. Venter, "A comparison of Intrusion Detection systems," *Computers & Security*, vol. 20, pp. 676–683, Dec. 2001. (Citato a pagina 9.)

[8] A. S. Ashoor and S. Gore, "Importance of Intrusion Detection System," vol. 2, no. 1, 2010. (Citato a pagina 9.)

[9] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, pp. 16–24, Jan. 2013. (Citato a pagina 9.)

[10] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application," *IEEE Access*, vol. 9, pp. 90603–90615, 2021. (Citato nelle pagine 10 and 14.)

[11] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020. (Citato a pagina 12.)

[12] F. Hoffmann, T. Bertram, R. Mikut, M. Reischl, and O. Nelles, "Benchmarking in classification and regression," *WIREs Data Mining and Knowledge Discovery*, vol. 9, Sept. 2019. (Citato a pagina 14.)

[13] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. Candela, "Practical Lessons from Predicting Clicks on Ads at Facebook," in *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, (New York NY USA), pp. 1–9, ACM, Aug. 2014. (Citato a pagina 15.)

[14] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (San Francisco California USA), pp. 785–794, ACM, Aug. 2016. (Citato a pagina 15.)

[15] S. L. Salzberg, "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Machine Learning*, vol. 16, pp. 235–240, Sept. 1994. (Citato a pagina 15.)

[16] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, (Funchal, Madeira, Portugal), pp. 108–116, SCITEPRESS - Science and Technology Publications, 2018. (Citato nelle pagine 15 and 16.)

[17] G. Bartelloni, "APPLICATION OF SOFT-BROWNIAN-OFFSET TO GENERATE CYBER-ATTACKS TO TRAIN INTRUSION DETECTORS," Apr. 2023. (Citato a pagina 21.)