2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume

DOI: 10.1109/DSN-S58398.2023

2833-292X/23/$31.00 ©2023 IEEE

Machine Learning on Public Intrusion Datasets: Academic Hype or Concrete Advances in NIDS?

Marta Catillo, Antonio Pecchia, Umberto Villano
Dipartimento di Ingegneria
Università degli Studi del Sannio
Benevento, Italy
{marta.catillo, antonio.pecchia, villano}@unisannio.it

***Abstract*—The number of papers on network intrusion detection based on machine and deep learning is growing at an unprecedented rate. Most of these papers follow a well-consolidated pattern: (i) proposal of an intrusion detection system based on machine (deep) learning, (ii) learning-testing with one (more) public intrusion dataset(s), (iii) achievement of outstanding detection performance. Is the intrusion detection problem solved? Unfortunately, no. This paper shares a deep reflection on the major limitations of public intrusion datasets and related machine learning experiments, which greatly diminish the findings documented by the literature. At the end of the day, in spite of the academic hype and the increasingly-complex machine and deep learning exercises around, the role of public datasets in advancing intrusion detection of real-world networks remains questionable. The way existing intrusion datasets are collected, released and used by the community should be approached with extreme caution. This paper provides concrete hints for the construction of future intrusion detection datasets and more rigorous machine learning experiments.**

***Index Terms*—deep learning, intrusion detection, NIDS, public datasets, cybersecurity**

I. Introduction

The body of scientific literature on machine learning (ML) and deep learning (DL) applied to network intrusion detection systems (NIDS) is growing at an unprecedented rate. Any simple Google scholar query for the titles including "intrusion detection" and one of "machine" or "deep" returns thousands hits. This ever-growing literature is pushed by several factors, which include, but are not limited to (i) availability of commercial and open source products (e.g., Netflow, Tranalyzer, CICFlowMeter) to transform computer network traffic into ready-to-use *flow records* suited to ML and DL, (ii) increasing number of public, flow-based, intrusion detection datasets (e.g., UNSW-NB15, UGR'16, CICIDS2017) and (iii) specialized hardware and deep learning frameworks (e.g., Keras, TensorFlow and PyTorch). A wide community of academics and practitioners – partially prompted by *publish-or-perish* pressure – is conducting measurement studies at the intersection of machine (deep) learning and NIDS.

The plethora of NIDS that keeps spreading every day follows the typical pattern in Fig. 1: (i) proposal of an intrusion detection system based on machine (deep) learning, (ii) learning-testing with one (more) public intrusion dataset(s), (iii) achievement of outstanding detection performance, e.g., recall-precision close to 1 (almost "perfect" detection). At the time being, intrusion detection might seem a perfectly-solved problem with no room for further improvements. Unfortunately, successful attacks reported every day by the news suggest that the intrusion detection problem is still there. The adoption of ML and DL for intrusion detection in real-world production networks remains challenging [1]; for example, deep models may misclassify many activities [2]. It is a fact that scholars need data to develop novel NIDS proposals and to test their validity. However, the rationale of using synthetic intrusion datasets as a "surrogate" of actual network traffic and potential intrusions – expecting that the outstanding detection figures obtained on the datasets will transfer to real-world environments – is just a hope.

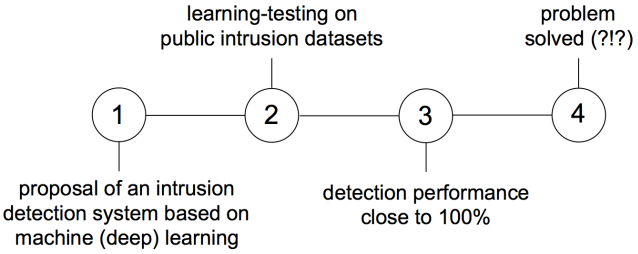

Fig. 1. Typical pattern of a NIDS paper based on machine (deep) learning.

This paper aims to share a deep reflection on the use of public intrusion datasets and the findings inferred on the top of them by means of machine (deep) learning. Notwithstanding the academic hype around this topic, concrete advances in NIDS clash with a tremendous number of flaws and limitations that will be discussed in the following. In spite of the brilliant ML and DL exercises around, the role of public datasets in advancing the state-of-the-practice in NIDS is questionable. The reflections presented in this paper are backed by well-argued scientific insights based on concrete facts inferred from existing intrusion datasets and related literature, and our direct experience on the topic. The way existing intrusion datasets are collected, released and used by the community should be approached with extreme caution. This paper provides concrete hints for the construction of future intrusion detection

datasets and more rigorous machine learning experiments.

Non-goals of this paper. This paper (i) is not a critique to existing literature, (ii) is not an exhaustive list of problems and best-practices, (iii) is not a data collection or experimentation guideline. The problem is huge and fraught by many open challenges; most of the items presented below are independent areas meant to be investigated on their own. Rather, this paper aims to start a constructive discussion on the topic and to make both prospective authors and reviewers aware of the issues pertaining to the use of synthetic data in ML(DL)-based NIDS. The rest of the paper is organized as follows. Sect. II provides some basic notions. Sect. III and IV address the problem both from the datasets and ML (DL) perspective. Sect. V concludes with our reflections on advancing knowledge on NIDS.

II. Basic notions on public datasets

Public intrusion detection datasets [3] are typically made of traffic data captured by tools, such as tcpdump and Wireshark, in a controlled environment that attempts to mimic the behavior of a network in normal traffic conditions subject to realistic attacks. Public datasets may be available in raw format (e.g., *pcap* packet data files) or – more commonly for ML and DL purposes – labeled comma-separated values (*csv*) records of features. Records are linked to the notion of *network flow*, which is a common abstraction in NIDS. A **network flow** is an aggregation of packets exchanged between a source computer and a destination across a network; a flow of packets is logically equivalent to a "call" or a "connection" according to the RFC 2722 [4]. A network flow is represented by a **flow record** – often informally called network flow – which holds the values of categorical and numeric features that provide context data and summary statistics computed from the headers of the packets pertaining to a flow. A recent survey discusses the trade-off between *packet*- and *flow*-based features and the implications in adversarial ML for NIDS [5].

There are different tools to generate flow records; however, some of them (e.g., Netflow, Tranalyzer) are essentially oriented to produce information useful for the computer networks community and are not particularly suited to intrusion detection purposes. This is the reason for the widespread use of **CICFlowMeter** [6] by the Canadian Institute for Cybersecurity (CIC), whose characteristic is to complement the customary information of each record (e.g., source and destination IP, protocol and ports) with statistical timing information (e.g., mean and variance of forward/backward packet inter-arrival times and of packets transferred per second). It is worth noting that the timing information was originally intended to recognize the type of encrypted flows [7], and that it turned out to be useful for intrusion detection only later on.

In the last few years, the most commonly used flow-based dataset for NIDS research has undoubtedly been **CICIDS2017**. Its reference paper [8] currently (April 2023) scores more than 2,300 Google scholar citations, which makes it among the top-used datasets. In the NIDS "research market", ready-to-use pre-processed data – straightforwardly usable by ML and DL practitioners – is a means to attract citations. This is perfectly acceptable, provided that the data are extensively validated (and fixed) before they are released to the public. Unfortunately, CICIDS2017, as well as the younger brother CSE-CIC-IDS2018 (less used at the moment), are fraught by all sorts of problems. Even if other flow-based NIDS datasets are available (e.g, UNSW-NB15 [9] and UGR'16 [10]), in the following the focus will be mostly on CICIDS2017 (due to its massive use) with further insights on other datasets.

III. Typical issues of public intrusion datasets

Simplifications of the data collection environment.

Synthetic intrusion datasets can hardly summarize complexity and uncertainty of real-world networks. The fact that the datasets are inevitably constrained by the simplifications of the specific network environment made at the time of the capture (e.g., topology, network speed and traffic conditions) is a major – and often neglected – reason for the difficulty to port successful "lab-made" NIDS to real-world environments, which include heterogeneous and non-stationary workloads. To be readily used as a benchmark for intrusion detection techniques and tools, a dataset should contain normal traffic intertwined with correctly-implemented state-of-the-art attacks representative of real-world network conditions.

Contemporaneity and effectiveness of the attacks.

CICIDS2017 is by now about seven years old, and – though passed off in many recent papers as a "modern" dataset – the attacks performed therein have little in common with the ever-evolving sophistication of current network attacks. It is well-known that benchmark datasets in NIDS remain valid over a short time period and they quickly become obsolete [5], [11]. Even more important, it has been demonstrated that Denial of Service (DoS) attacks of both CICIDS2017 and other public traffic captures (ISCXIDS2012, NDSec-1 2016, MILCOM 2016 and SUEE 2017) are mostly ineffective against suitably-configured and hardened web servers [12]. This leads to the paradox in which many researchers claim they are able to detect about 100% of attacks that have no impact or just a sort of minor disturbance on real-world services.

Representativeness of the normal baselines.

CICIDS2017 contains normal traffic that simulates the behavior of 25 users and a few protocols on an internal network; there are examples of datasets that rely on simple traffic generators, such as httperf. The representativeness of the normal baselines is crucial in ML and DL methods that aim to learn the characteristics of the normal traffic in order to pinpoint anomalies as an attack (such as semi-supervised learning methods). The detection results depend on the baseline used; however, this problem is typically neglected.

Bugs of the feature extractor and incorrect flow records.

The extraction of flow records from packet captures is undoubtedly a difficult task. Only after a lustrum of "blind" use of CICIDS2017 by NIDS researchers, a number of studies have noted several major bugs and errors affecting the ubiquitous CICFlowMeter, which led to incorrect flow

records of both CICIDS2017 and CSE-CIC-IDS2018 [13]–[16]. Since its inception, CICFlowMeter has been around in the public domain in many different versions due to supporting libraries, correction of bugs, addition/deletion of features. This is exacerbated by the lack of detailed documentation on the tool configuration, even when it would be really necessary (e.g., knowledge of the timeout after which flows are cut and thus producing fragments of flows). At the time being, we are aware of several, fixed, alternatives of CICIDS2017 and CICFlowMeter [17]–[19], which makes it hard to keep experimentation up with these continuous changes.

Data Labeling.

Assembling ready-to-be-used datasets requires a significant labeling effort, i.e., marking each record as either "normal" or "attack" (and possibly the type of the attack). Due to the huge volume of records, the task turns out to be difficult, whether carried out manually or automatically. A common solution is resorting to time-based labeling: the traffic between the attacker and the victim during the interval of time of the attack is marked as "attack". This practice is dangerous, due to the concrete chance to mark background normal traffic as malicious, thus contaminating the dataset. The challenges of labeling network traffic have been known since the early days of NIDS; however, labeled-unlabelled data still represent an open problem. For example, recent work has focused on traffic labeling methods [11] and the utility of unlabelled data [20].

Class imbalance.

The management of the environment used to collect network traffic for a dataset is a complex matter. For example, it is hard to produce the same number of flow records for each attack performed; some attacks, such as DoS, are inherently much more voluminous than other types. Just to provide few concrete examples, the original CICIDS2017 contains 230,124 DoS Hulk, 652 Web Attack XSS and 11 Heartbleed records [13]; attack records of the more recent WUSTL-IIOT-2021 dataset for the industrial Internet of Things (IoT) – to mention an adjacent domain suffering from similar issues – accounts about 90% DoS data and less than 0.5% for other categories (Backdoor, Command Injection) [21]. Class imbalance is a long-standing debate in the community: while it mimics the cardinalities of the classes in real-world settings, it is a major threat to the evaluation metrics as described below.

IV. Implications from the ML (DL) perspective

Attack-revealing features and ease of detection.

Flow records typically consist of many features (e.g., 83 in the ubiquitous CICFlowMeter). Some features may trivially expose the attacks to the ML (DL) model: this is the case when there is a single attacker in the capture environment, the attacks pass through the same frontier router or there is a single victim: a trivial check of the IP addresses allows to discriminate normal from attack flows. Unfortunately, it is not rare to find manuscripts – submitted to peer review – that propose ML(DL)-based detectors where all the features are used *as a whole* with no critical reasoning at all. Even when the "attack-revealing" features are removed, sometimes the simplicity and regularity of normal baselines and attack patterns allow one to detect the attack flows with simple analysis techniques. This is discussed in [22], which claims that a simple OneR classifier performs fairly well on many academic datasets, and so that the use of DL is overkill.

Data partitioning.

ML-practitioners rely on different data splits for training, validation and testing purposes. Random sampling with no replacement would seem perfectly reasonable to obtain disjoint data splits from a given dataset of records; however, it leads to process and classify *in an arbitrary order* records that indeed belong to a timestamped sequence (either normal or attack). An issue arises when the DL model devised is a neural network with memory, such as a recurrent neural network (RNN) or long short-term memory (LSTM). Another risk is to miss rare/unusual records from one (more) of the splits, thus altering the representativeness. This problem is dealt with in [23] with some possible solutions. Further challenges arise if the objective of the research is the analysis of data drift phenomena or advanced persistent threat (APT).

Unmotivated complexity.

Notwithstanding the large body of literature, the academic solutions proposed seem to find no or very limited adoption in actual production environments. Most – if not all – recent proposals are based on the use of deep networks. Among the highly-complex deep networks proposed so far for intrusion detection are cascades-ensembles of Autoencoders (AE), AE plus LSTM, Convolutional Neural Networks (CNN) plus LSTM [24]–[26]. These solutions are characterized by large footprint, high energy consumption, and fairly high response times. But the most serious problems are learning and tuning times. The learning and tuning of a deep network for a given dataset may require weeks, if not months, of try-and-retry experimentations. The time required grows with the complexity of the solution. It would be great to know for each solution proposed by the literature how long it takes learning/tuning on a different dataset and why the proposing authors did not opt for much more simple solutions instead of "blindly" applying increasingly complex methods.

Use of the evaluation metrics.

Class imbalance may cause overestimating the actual "strength" of NIDS. Let the confusion matrix in TABLE I, where the normal records (negatives) are much lower than attacks (positives), i.e., 100 *vs* 10,000, and half of the negatives (50) are misclassified. The numbers return satisfactory recall (R) and precision (P) of 0.99, but an unacceptable false positive rate (FPR) of 0.50. Moreover, a high R may mask the

TABLE I
DETECTION PERFORMANCE BIASED BY CLASS IMBALANCE.

	label	
pred.	**N**	**P**
N	50	100
P	50	9,900
total	100	10,000

R = 0.99
P = 0.99
FPR = 0.50

attack records		Det_X	Det_Y
"A"	9,900	9,900	9,800
"B"	50	0	50
"C"	50	0	50
total	10,000	9,900	9,900

fact that a detector, such as Det_X in TABLE I, might detect just the top-occurring attack class, i.e., "A", but *none* of "B" or "C": the much better detector Det_Y – detecting attack "A" fairly well and all of "B" and "C" – ends up with the same R of Det_X due to class imbalance. R and P alone (typically the only metrics available in many papers) can be misleading and lead to unfulfilled expectations: disclosing the FPR is necessary in NIDS. For unbalanced test sets, detection performance can be measured with P-R curves and AUPRC (area under the P-R curve) [27]. An interesting option is evaluating NIDS based on the ability to detect intrusions held-out from training [28].

Lack of transferability.

Most of the existing – and impressive – intrusion detection results hold just on the top of the individual datasets that were used to obtain the results themselves. In the context of NIDS, transfer learning [29] might be used to learn a predictive function for a "target" domain (real-world traffic) based on a "source" domain (general benchmark network datasets) given the same learning task (intrusion detection) in both domains. While [5] indicates the use of transfer learning as a future direction, recent work demonstrates that it is hard to transfer NIDS knowledge, even across similar attacks [30]–[33].

V. Advancing NIDS: What it can be done

Present-day success of artificial intelligence has shown that machines do learn. But machines can also learn silly and useless patterns, i.e., "garbage in – garbage out" (GIGO) principle. The availability of ready-to-use datasets, environments and libraries where a few lines of Python code are enough to build an ML model is a wonderful opportunity to experiment. Unfortunately, sometimes detection performance close to 100% is the outcome of toy examples. Advances of scientific knowledge on NIDS can be achieved by approaching dataset collection, release and usage with extreme caution.

Data Collection. In our opinion, traffic data collection is – and will – remain the biggest problem. The network environment is a moving target with a continuous shift in the technologies, devices, protocols, appliances, traffic and attacks (just to mention a few). Collecting realistic "lab-made" traffic is hard; at the other end of the spectrum, large-scale organizations and companies are constrained by obvious concerns because connection meta-data (implicitly) or traffic payloads (explicitly) carry confidential information. Simplifications are unavoidable; however, the construction of future intrusion detection datasets can advance along different directions. For example, the network used should not be too "basic" and the normal traffic should not be generated through simplistic and repetitive traffic generators. There exist frameworks to spot defects while generating datasets with synthetic attacks [34]. Scholars should strive to emulate state-of-the-art attacks, whose parameterization and timing must be varied in order to elicit different attack variants, possibly under different configurations and defenses of the victims. While doing so, a reasonable balancing of normal and attack traffic is also advised. Network traffic data should be accompanied by detailed performance measurements of the victim during the occurrence of both normative operations and attacks. This item is missed by the current dataset proposals.

Data Release. Assuming traffic data is collected correctly, any error during *preprocessing*- and *release*-related tasks can invalidate the collection efforts and ends up with major damage to the community. In this respect, there is a lot to learn from the (in)famous history of CICFLowMeter–CICIDS2017. As for the ubiquitous abstraction of flow record, feature extraction should be carried out by leveraging well-tested and documented tools. Another issue pertains to labeling, which cannot be performed only on the basis of start–end time of the attacks; in any case, the methodology used should be clearly indicated to allow replication and verification. Datasets should be accompanied by extensive documentation of network, OS, application configurations and workload specifications in order to allow third-party researchers to replicate the datasets by mimicking the originating setups. A further problem is the lack of a tool to perform the inverse operation of mapping the flow records back to the network flows, so as to inspect the corresponding packets from the originating captures. This is of fundamental importance to help understand the nature of a flow and to explain its classification. As far as we know, no proposal currently available in the literature follows all these hits; any attempt to move in the directions outlined above can represent a steady contribution for the NIDS community.

Data Usage. Doing "mindful" ML and DL is a hard task. Different from other domains, such as image/speech recognition and natural language processing, NIDS pose a number of unique challenges. In order to advance NIDS, ML developers must thoroughly understand all the technical information associated with the datasets, beforehand. Capitalizing on a few detection metrics, e.g., R and P, is not enough without deeper investigation. There are papers that achieve astonishing results just because the attacks-revealing features were not discarded, i.e., the well-known "Clever Hans" effect (the model produces the correct predictions based on the wrong features). Chaining increasingly-complex deep neural networks after networks must not be the rule, especially when simpler approaches are possible; if not, the authors should clearly provide information on the complexity and time required to perform a new learning procedure. Choosing the proper performance measures and analyzing them in-depth, especially if the dataset is unbalanced. A special focus should be on the reproducibility and transferability of the models, by communicating the precise hyperparameters and after testing on alternative datasets or, better, real-world data. In this respect, authors must avoid using wrong/outdated datasets. Unfortunately, it is very common to see IoT and, more recently, 5G NIDS papers assessed through general network datasets, such as CICIDS2017 (which surely it is not an IoT/5G dataset); one more issue is the significant use of NSL-KDD and KDD CUP'99, which are outdated.

As a side note, we believe the scientific community shares some "responsibilities" on this. While the authors are tempted to submit whatever comes to their minds, reviewers are responsible to check what (and how) datasets are used, correctness

and quality of ML experiments; this is intertwined with the need for profound changes in how NIDS papers are assessed.

REFERENCES

[1] 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, no. 1, p. e4150, 2021.

[2] L. Zhang, X. Lu, Z. Chen, T. Liu, Q. Chen, and Z. Li, "Adaptive deep learning for network intrusion detection by risk analysis," *Neurocomputing*, vol. 493, pp. 46–58, 2022.

[3] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computer & Security*, vol. 86, pp. 147–167, 2019.

[4] [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2722

[5] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[6] [Online]. Available: https://github.com/ahlashkari/CICFlowMeter

[7] G. Draper-Gil., A. H. Lashkari., M. S. I. Mamun., and A. A. Ghorbani., "Characterization of encrypted and vpn traffic using time-related features," in *Proc. International Conference on Information Systems Security and Privacy*. SciTePress, 2016, pp. 407–414.

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani., "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. International Conference on Information Systems Security and Privacy*. SciTePress, 2018, pp. 108–116.

[9] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. International Conference Military Communications and Information Systems Conference*. IEEE, 2015, pp. 1–6.

[10] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Computer & Security*, vol. 73, pp. 411–424, 2017.

[11] J. L. Guerra, C. Catania, and E. Veas, "Datasets are not enough: Challenges in labeling network traffic," *Computers & Security*, vol. 120, p. 102810, 2022.

[12] M. Catillo, A. Pecchia, M. Rak, and U. Villano, "Demystifying the role of public intrusion datasets: a replication study of DoS network traffic data," *Computers & Security*, vol. 108, p. 102341, 2021.

[13] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study," in *Proc. Security and Privacy Workshops*. IEEE, 2021, pp. 7–12.

[14] A. Rosay, F. Carlier, E. Cheval, and P. Leroux, "From CIC-IDS2017 to LYCOS-IDS2017: A corrected dataset for better performance," in *Proc. International Conference on Web Intelligence*. ACM, 2021, pp. 570–575.

[15] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error Prevalence in NIDS datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *Proc. Conference on Communications and Network Security*. IEEE, 2022, pp. 254–262.

[16] M. Lanvin, P.-F. Gimenez, Y. Han, F. Majorczyk, L. Mé, and E. Totel, "Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes," in *Proc. International Conference on Risks and Security of Internet and Systems*, 2022.

[17] [Online]. Available: http://downloads.distrinet-research.be/WTMC2021

[18] [Online]. Available: https://lycos-ids.univ-lemans.fr/

[19] [Online]. Available: https://intrusion-detection.distrinet-research.be/CNS2022/index.html

[20] G. Apruzzese, P. Laskov, and A. Tastemirova, "SoK: The impact of unlabelled data in cyberthreat detection," in *Proc. European Symposium on Security and Privacy*. IEEE, 2022, pp. 20–42.

[21] [Online]. Available: https://www.cse.wustl.edu/~jain/iiot2/index.html

[22] L. D'hooge, M. Verkerken, T. Wauters, F. De Turck, and B. Volckaert, "Castles Built on Sand: Observations from Classifying Academic Cybersecurity Datasets with Minimalist Methods," in *Proc. International Conference on Internet of Things, Big Data and Security*. SciTePress, 2023.

[23] C. Catania, J. Guerra, J. M. Romero, G. Caffaratti, and M. Marchetta, "Beyond Random Split for Assessing Statistical Model Performance," 2022. [Online]. Available: http://arxiv.org/abs/2209.03346

[24] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. International Conference of Network and Distributed System Security Symposium*, 2018.

[25] Y. Zhong, W. Chen, Z. Wang, Y. Chen, K. Wang, Y. Li, X. Yin, X. Shi, J. Yang, and K. Li, "HELAD: A novel network anomaly detection model based on heterogeneous ensemble learning," *Computer Networks*, vol. 169, p. 107049, 2020.

[26] G. De La Torre Parra, P. Rad, K. R. Choo, and N. Beebe, "Detecting internet of things attacks using distributed deep learning," *Journal of Network and Computer Applications*, vol. 163, p. 102662, 2020.

[27] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. International conference on Machine learning*. ACM, 2006, pp. 233–240.

[28] T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, "Which algorithm can detect unknown attacks? comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection," *Computers & Security*, vol. 127, p. 103107, 2023.

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[30] M. Verkerken, L. D'Hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, p. 12, 2021.

[31] G. Apruzzese, L. Pajola, and M. Conti, "The cross-evaluation of machine learning-based network intrusion detection systems," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5152–5169, 2022.

[32] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "Transferability of machine learning models learned from public intrusion detection datasets: the CICIDS2017 case study," *Software Quality Journal*, vol. 30, pp. 955–981, 2022.

[33] G. de Carvalho Bertoli, L. Alves Pereira Junior, O. Saotome, and A. L. dos Santos, "Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach," *Computers & Security*, vol. 127, p. 103106, 2023.

[34] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 2, p. 8, 2021.