Journal of Universal Computer Science, vol. 30, no. 9 (2024), 1244-1264
submitted: 27/9/2023, accepted: 30/5/2024, appeared: 14/9/2024

Using CVSS scores can make more informed and more adapted Intrusion Detection Systems

Robin Duraz
(Chaire of Naval Cyberdefense, Lab-STICC, Brest, France
https://orcid.org/0000-0001-8364-2989, robin.duraz@ecole-navale.fr)

David Espes
(UBO, Lab-STICC, Brest, France
https://orcid.org/0000-0003-3445-947X, david.espes@univ-brest.fr)

Julien Francq
(Naval Group (Naval Cyber Laboratory, NCL), Ollioules, France
https://orcid.org/0000-0002-4604-4522, julien.francq@naval-group.com)

Sandrine Vaton
(IMT Atlantique, Lab-STICC, Brest, France
https://orcid.org/0000-0001-8940-6004, sandrine.vaton@imt-atlantique.fr)

Abstract: Intrusion Detection Systems (IDSs) are essential cybersecurity components. Previous cyberattack detection methods relied more on signatures and rules to detect cyberattacks, although there has been a change in paradigm in the last decade, with Machine Learning (ML) enabling more efficient and flexible statistical methods. However, ML often suffers from the lack of, and proper use of, cybersecurity information, be they for proper evaluation or even improving performance. This paper shows that using a *de facto* standard in cybersecurity: the Common Vulnerability Scoring System (CVSS), can improve IDSs at different levels, from helping in training an IDS, to more properly evaluating its performance, even taking into account systems with different protection requirements. This paper introduces Cyber Informedness, a new metric considering cybersecurity information to give a more informed representation of performance, influenced by the severity of the attacks encountered. Consequently, this metric is also able to differentiate performance of IDSs when security requirements, Confidentiality, Integrity and Availability, are defined using CVSS' environmental parameters. Finally, sub-parts of this metric can be integrated into the training phase's loss of Neural Networks (NNs)-based IDSs to build IDSs that better detect more severe attacks.

Keywords: Cybersecurity, Metrics, Machine Learning, Intrusion Detection Systems, CVSS
Categories: I.2.1, I.2.6
DOI: 10.3897/jucs.131659

1 Introduction

The world is increasingly digitalized, which brings a plethora of cybersecurity threats. While it is essential to make systems more secure by design [Ashibani and Mahmoud 2017], nothing is ever perfectly secure, so alternative solutions are needed. IDSs are used to monitor and analyze traffic and system logs to detect anomalies and potential attacks.

Traditional IDSs are signature-based and rather successful in detecting known attacks, with a very low probability of giving false alarms with sufficiently well-crafted rules, but they can easily miss zero-day or polymorphic threats [Hindy et al. 2020].

In the last decade, however, one of the most extensive research directions concerns ML algorithms. Many works [Sarker et al. 2020, Ferrag et al. 2020, Xin et al. 2018, Khraisat et al. 2019, Lakshminarayana et al. 2019] have expanded upon ML and Deep Learning (DL) performance relative to intrusion detection on popular datasets, and show that those approaches perform well. Finally, other approaches such as Artificial Immune Systems [Aickelin et al. 2013], Genetic Programming [Abadeh et al. 2011] or approaches based on Fuzzy Logic [Masdari and Khezri 2020] can either replace or complete classical ML approaches. While there has been much progress in devising different methods to perform intrusion detection using cybersecurity data, cybersecurity knowledge has rarely been used to improve upon these methods.

As far as metrics are concerned, intrusion detection is addressed from the ML viewpoint with generic metrics such as Accuracy, Precision, Recall or F1-Score. While these metrics are extensively used and tested, they suffer from significant flaws, particularly when performing multi-class classification. Indeed, imbalance in the data heavily influences results, and poor results on underrepresented classes are generally hidden. Furthermore, those metrics treat all data equally and are unable to cater for differences in the cost of mistakes (attacks missed or false alarms). Failing to detect port scans will obviously be less penalizing than failing to detect the exfiltration of confidential data.

While integrating cybersecurity knowledge into ML-based IDSs might prove fruitful, it poses a number of challenges. The typical way to provide information to ML methods is by using training data. Therefore, integrating cybersecurity knowledge will require to transform the knowledge to supplement this data. In cybersecurity, attacks are generally defined by their Technics, Tactics and Procedures (TTPs), the impact they have on the attacked systems and therefore often have a score related to their criticity. While integrating the MITRE ATT&CK framework[1] to improve ML-based IDSs might be the most beneficial because of its completeness, it would require a significant adaptation of their training mechanisms. On the contrary, adding a numerical value representing the severity of attacks is a much easier task. This score is generally computed using the CVSS[2], which takes into account parameters relative to the difficulty of performing an attack, as well as its impact, to compute a score representative of an attack's severity.

In a previous work [Duraz et al. 2023], authors introduced three new metrics based on CVSS, Miss Cost (MC), False Alarm Cost (FAC) and Cyber Informedness (CI) to provide a more informed evaluation of the actual performance of ML-based IDSs. These metrics can benefit both the cybersecurity expert that will use and integrate this IDS and the ML expert that builds and validates an IDS based on ML algorithms by showing results that both sides have more confidence in.

This work supplements previous research by extending the use of CVSS scores with respect to ML-based IDSs. As such, the contribution of this research is twofold:

- Show that CVSS scores' environmental parameters can be used to help selecting IDSs that are more adapted to specific system configurations with their own security requirements.
- Introduce a new loss formulation for NN-based IDSs that builds upon the Cross-Entropy loss and two of the newly defined metrics, Miss Cost (MC) and False

[1] https://attack.mitre.org/
[2] https://www.first.org/cvss/v3.1/specification-document

Alarm Cost (FAC) to train NN-based IDSs. Furthermore, the newly formulated loss is shown to help NN-based IDSs in detecting more severe attacks, while at the same time helping in reducing the amount of missed attacks, and is validated on DAPT2020, a dataset better representing current attack methodologies.

Consequently, this work closes the loop started in [Duraz et al. 2023] by integrating cybersecurity knowledge into both phases (training and evaluation) of an IDS's design. From simply evaluating performance in a more informed way, the new methodology proposed also enables a better adaptation to protecting systems with specific requirements, as well as the possibility to influence training to obtain IDSs that better detect more severe attacks.

The rest of the paper is organized as follows: Section 2 presents related works. Section 3 describes the proposed approach, while Section 4 presents the experimental setup. Section 5 presents and analyzes the results. Finally, Section 6 concludes the paper and discusses future avenues of research.

2 Related work

For ML-based IDSs, cybersecurity datasets are required. Unfortunately, it is difficult to obtain realistic data, i.e., with at least a diversity of up-to-date attacks, a complete environment, as well as traffic representative of the real world (imbalanced, with errors, etc.). Using real world data is an obvious choice, but is often impossible to obtain because of confidentiality or security reasons. Another solution is to create a synthetic dataset. While it eliminates the previous problems, it is much more difficult to make it realistic. It is important to follow a thorough methodology, such as highlighted by [Bhuyan et al. 2015, Sharafaldin et al. 2017], to ensure quality of the data created.

2.1 Datasets

KDD'99 [KDD Cup 99 Data 1999] and NSL-KDD [Tavallaee et al. 2009] are the two most used datasets [Hindy et al. 2020]. However, these datasets, and particularly the former, are heavily criticized because of their age and various other problems such as redundancy [Creech and Hu 2013, Siddique et al. 2019, Tobi and Duncan 2018].

The UNSW-NB15 [Moustafa and Slay 2015] and CIC-IDS2017 [Sharafaldin et al. 2018] datasets are more recent and based on quite complete environments. CIC-IDS2017 follows the methodology defined in [Sharafaldin et al. 2017] and criteria defined in [Gharib et al. 2016] to ensure quality of the created dataset. Both being more recent datasets, it also ensures that the environment and simulated traffic are more representative of nowadays' real-world traffic. The more recent DAPT dataset [Myneni et al. 2020] appears to better represent current attack methodologies. While it is not possible to attribute CVSS scores on this dataset's four stages (see Table 1), the different activities pertaining to each stage can be scored following the same methodology as CIC-IDS2017. Consequently, the UNSW-NB15, CIC-IDS2017 and DAPT2020 datasets are retained for this research. Details about each dataset are presented in Table 1.

2.2 Metrics

In order to evaluate performance of different ML-based IDSs, various metrics are generally used. The most complete representation of an IDS's performance and basis for

Dataset	Number of instances per class	Total
UNSW-NB15	Normal: 2218761, Generic: 215481, Exploits: 44525, Fuzzers: 24246, DoS: 16353, Reconnaissance: 13987, Analysis: 2677, Backdoor: 2329, Shellcode: 1511, Worms: 174	2540047
CIC-IDS2017	Benign: 2273097, DoS Hulk: 231073, Portscan: 158930, DDoS: 128027, DoS GoldenEye: 10293, FTP-Patator: 7938, SSH-Patator: 5897, DoS Slowloris: 5796, DoS Slowhttptest: 5499, Botnet: 1966, Web Attack Brute Force: 1507, Web Attack XSS: 652, Infiltration: 36, Web Attack SQL Injection: 21, Heartbleed: 11	2830743
DAPT2020	**Normal**: 63712 **Reconnaissance*** – Network Scan: 7614, Account Discovery: 124, Directory BruteForce: 1503, Web Vulnerability Scan: 2574, Account BruteForce: 94 **Establish Foothold*** – SQL Injection: 55, Directory Bruteforce: 8467, Account Bruteforce: 47, Account Discovery: 12, Malware Download: 2, Network Scan: 2, CSRF: 7, Command Injection: 12 **Lateral Movement*** – Network Scan: 117, Backdoor: 20, Account Discovery: 2272, SQL Injection: 29, Privilege Escalation: 13 **Data Exfiltration*** – Network Scan: 9, Data Exfiltration: 6	86691

* These represent DAPT2020's four attack stages. Not in bold are each stage's activities.

Table 1: Datasets details

most metrics, e.g., Accuracy, Precision, Recall, F1-Score, is the full confusion matrix. While the full confusion matrix remains one the best representation of performance, it can quickly become difficult to use as the number of classes increases. For research on intrusion detection, the metrics mentioned above that are generally used share two major drawbacks. Firstly, they are unable to treat differently different attack classes, and this is problematic because attacks are not equally dangerous, and remediation mechanisms are different. Secondly, they are mostly not resistant to imbalance.

Imbalance in the data is a problem already highlighted in the literature. It has been described in details in [Jeni et al. 2013, Gu et al. 2009, Sokolova et al. 2006], showing that many metrics might be ill-defined in case of heavily imbalanced datasets. In intrusion detection, normal traffic generally represents a part of the data significant enough for IDSs to show a high performance while potentially missing all attacks, e.g., classifying all traffic as normal (thus missing all attacks) in UNSW-NB15 still achieves more than 87% Accuracy. It thus highlights the need to find better metrics, or simply account for the skewness of class distributions.

To solve the imbalance problem, [Chicco 2017] has suggested the use of the Matthews Correlation Coefficient (MCC) that is probably the most complete metric with regard to summarizing the confusion matrix since it captures all the information contained therein, i.e., both True and False Positives and Negatives. While it is originally defined for binary classification, it can also be extended to the multi-class setting.

However, authors in [Zhu 2020] offer a strong rebuttal to the use of MCC in case of imbalanced datasets and suggest using metrics that are more stable with regard to imbalance, such as the geometric mean of TPR (True Positive Rate) and TNR (True Negative Rate) and Bookmaker Informedness (BI) equal to $TPR + TNR - 1$.

According to [Zhu 2020], BI accounts for imbalance and can reflect a less biased view of performance. Its formula is simple, yet appears to offer what most other metrics cannot, i.e., it allows to capture performance on both positive and negative instances with equal importance, irrespective of the imbalance. It is something that Accuracy or MCC are not capable of doing. However, resistance to imbalance of the MCC and BI metrics is still relatively unclear in the multi-class setting, since both [Chicco 2017] and [Zhu 2020] limited their analysis to the binary setting. Furthermore, although some metrics might appear more suitable than others, it is advised in [Sokolova et al. 2006] to rely on multiple metrics to correctly compare two algorithms.

Finally, MCC and BI appear to offer part of the solution to the imbalance problem. However, there is currently no metric that offers to solve the problem of attack classes that are inherently not equally important for the monitored system. For example, breaches in servers holding classified data need to be prioritized much more than simple brute-forcing attempts. Therefore, new metrics based on CVSS presented in this paper can fill in this gap. Following the advice in [Sokolova et al. 2006], multiple metrics will be retained to evaluate performance of ML-based IDSs: Accuracy, Precision (PPV), Recall (TPR), F1-Score, MCC, BI, and the three new metrics based on CVSS. Furthermore, new metrics based on CVSS will be generalizations of other metrics, for more credibility and better comparison between results.

2.3 CVSS for IDSs

CVSS is one of the most extensively used frameworks in cybersecurity, and allows attribution of numerical scores to vulnerabilities. Since vulnerabilities are exploited by attacks, these attacks can also, by association, be attributed CVSS scores. Therefore, CVSS scores can be used to get a numerical representation of a system's security by accounting for attacks it is susceptible to. Research on the usage of CVSS scores in the context of cybersecurity has mainly focused on evaluating the security of systems and few has been done to evaluate IDSs. While the idea of leveraging CVSS in Intrusion Detection is not recent, as in [Aussibal and Gallon 2008] where it has been used to evaluate severity of alerts raised by probes, its actual use has not progressed much since then.

In [Gao et al. 2018, Frigault et al. 2017], CVSS scores have been used in coordination with attack graphs and Bayesian networks to evaluate or estimate the security of networks, thus extending the use of CVSS scores to also consider attack paths instead of a single vulnerability. In [Boudermine 2023], CVSS scores are used with dynamic attack graphs to evaluate the overall security of a system.

Although CVSS scores are originally defined for regular IT networks, [Ur-Rehman et al. 2020] have focused on extending the framework to also encompass Industrial Control Systems, showing the interest in such framework to evaluate security of a system. Finally, recent work [Bolivar et al. 2019] suggests that CVSS can be used to prioritize what is more severe. While research to extend CVSS scores to more use cases exists, it focuses on evaluating the security of a system and can be researched further to improve IDSs. This work focuses on improving design and evaluation of IDSs by using CVSS to both train and evaluate IDSs for a more informed representation of the security it provides. While the means are improving both the design and evaluation of ML-based IDSs, the goal, motivation and end results remain to reduce the cybersecurity risk by training and using ML-based IDSs that properly take into account both cybersecurity information and requirements.

3 Integrating CVSS scores into IDSs

CVSS scores can be integrated into all phases pertaining to building an IDS, as shown in Figure 1.

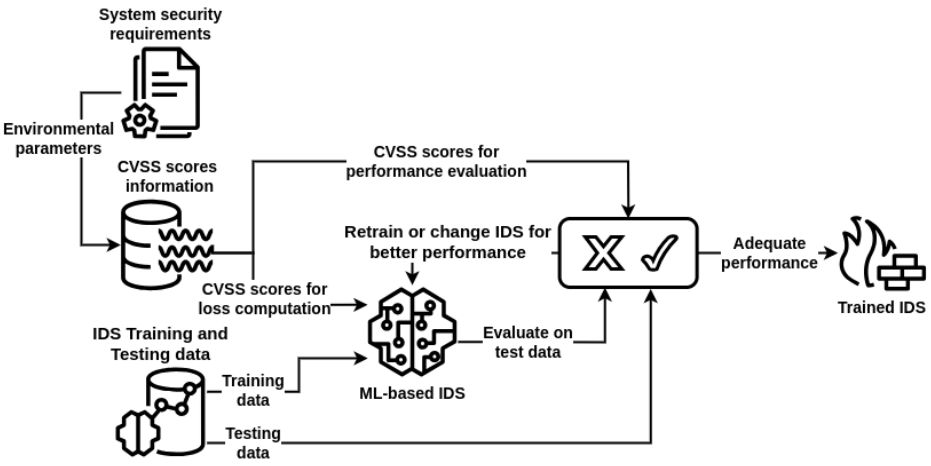

Figure 1: CVSS integration into IDSs

First, CVSS scores and their usage can be influenced by the definition of the system to be protected. Taking into account Confidentiality, Integrity and Availability (CIA) requirements of the system can change CVSS scores via their environmental parameters. In this way, CVSS scores will be adapted to the given system, opening the way for building specifically designed IDSs. Secondly, CVSS scores can be integrated into the training phase of ML-based IDSs. While this might require substantial work for most ML-based IDSs, CVSS scores can more easily be integrated in the loss formulation of NNs-based IDSs. Last but not least, CVSS scores can be integrated into evaluation metrics that can take into account the severity of attacks, thus making more informed IDSs. Because the evaluation metrics integrating CVSS are the basis for all three, this is developed first.

3.1 CVSS in evaluation metrics

Besides being an evaluation of performance, a given metric (or set of metrics) in cyber-security should be able to provide objective information about the actual cost of being mistaken, particularly in critical situations. In this paper, three new metrics using CVSS scores and accounting for missed attacks and false alarms are thus created.

3.1.1 False Alarm Cost and Miss Cost

Let c be a class, with $c \in \{0, 1, ..., C\}$ and 0 being the normal class. For every instance i, let G_i be the ground truth value and D_i be the decision for this instance. CVSS_i is the CVSS score corresponding to instance i.

1 stands for the indicator function and $\bar{\bullet}$ is the averaging operator. $\overline{\text{CVSS}_c}$ thus corresponds to the mean of CVSS scores for instances belonging to class c, which is necessary when instances of the same class do not have the same CVSS score (as is the case in the UNSW-NB15 dataset).

For each attack class c ($c \neq 0$), and with N the total number of instances, we define the False Alarm Cost (FAC, Equation 1) and the Miss Cost (MC, Equation 2) as follows:

$$\text{FAC}_c \overset{def}{=} \frac{\sum_{i=1}^{N} \mathbf{1}_{D_i=c} \cdot \mathbf{1}_{G_i \neq D_i}}{10 \sum_{i=1}^{N} \mathbf{1}_{D_i=c}} \cdot \overline{\text{CVSS}_c} \tag{1}$$

$$\text{MC}_c \overset{def}{=} \frac{\sum_{i=1}^{N} \mathbf{1}_{D_i \neq c} \cdot \mathbf{1}_{G_i=c} \cdot \text{CVSS}_i}{10 \sum_{i=1}^{N} \mathbf{1}_{G_i=c}} \tag{2}$$

In both formulae, the number 10 in the denominator represents the maximum possible value of a CVSS score, thus acting as a normalizing constant (bounding results between 0 and 1) while also highlighting the importance of attacks having a higher score.

As such, both formulae are generalizations of ML metrics. FAC is the generalization of the False Discovery Rate, the proportion of mistakes by predicting a specific class. Intuitively, it represents the frequency of false alarms, weighted by the CVSS score of these alarms. MC is the generalization of the False Negative Rate, the proportion of class instances that are incorrectly classified. Intuitively, it represents the frequency of missed attacks, weighted by their individual CVSS scores. These newly defined metrics are equal to their ML metrics counterparts when all CVSS scores are equal to 10 for classes different from normal traffic.

3.1.2 Cyber Informedness

Both metrics mentioned above can be combined into a single metric taking into account both False Positives and False Negatives that is defined analogously to BI. Therefore, it is assumed it would similarly exhibit nice properties regarding class imbalance.

For each class c ($c \neq 0$), the Cyber Informedness (CI) metric that contains both FAC and MC is given by (3).

$$CI_c \overset{def}{=} 1 - \text{FAC}_c - \text{MC}_c \tag{3}$$

This metric aims to give a cybersecurity-informed idea about the performance of an IDS, aggregating both FAC and MC, with 1 being the best possible score. It also represents the success of an IDS to correctly identify a specific attack, with less penalties for failing to recognize less critical attacks.

3.2 CVSS in the loss computation

In much the same way as in the previously defined metrics, CVSS scores can be integrated into a loss used by a NN to train. Since in most multi-class classification problems, the

loss used is a Cross-Entropy loss (CE), CVSS scores have been integrated into a custom loss based on the CE.

Let c be a class, with $c \in \{0, 1, ..., C\}$ and 0 being the normal class. For every instance i, let x_i represents the output logits of the Neural Network, G_i be the ground truth value and D_i be the decision for this instance. CVSS_i is the CVSS score corresponding to instance i. Finally, let V be the set of indices for which CVSS_i exists.

As a reminder, the basic CE is defined in Equation 4.

$$\text{CE} \overset{def}{=} \sum_{i=1}^{N} -log \frac{exp(x_{i,G_i})}{\sum_{c=1}^{C} exp(x_{i,c})} \tag{4}$$

To properly integrate CVSS scores into a custom loss, this custom loss has been divided in three different parts:

– Miss Cross-Entropy loss (MCE), a part accounting for missed attacks, defined analogously to MC. It is defined in Equation 5.

$$\text{MCE} = \sum_{i=1}^{N} -log \frac{exp(x_{i,G_i})}{\sum_{c=1}^{C} exp(x_{i,c})} \cdot \mathbf{1}_{G_i \neq 0, G_i \neq D_i, i \in \text{V}} \cdot \text{CVSS}_i \tag{5}$$

– False Alarm Cross-Entropy loss (FACE), a part accounting for false alarms, defined analogously to FAC. It is defined in Equation 6.

$$\text{FACE} = \sum_{i=1}^{N} -log \frac{exp(x_{i,G_i})}{\sum_{c=1}^{C} exp(x_{i,c})} \cdot \mathbf{1}_{G_i = 0, D_i \neq 0} \cdot \overline{\text{CVSS}_{D_i}} \tag{6}$$

– Remaining Cross-Entropy loss (RCE), a part accounting for missed attacks when CVSS score does not exist, which is required for the UNSW-NB15 dataset. It is defined in Equation 7.

$$\text{RCE} = \sum_{i=1}^{N} -log \frac{exp(x_{i,G_i})}{\sum_{c=1}^{C} exp(x_{i,c})} \cdot \mathbf{1}_{G_i \neq 0, G_i \neq D_i, i \notin \text{V}} \tag{7}$$

Finally, the complete loss CVSSCE is defined in Equation 8, simply being a sum of its three parts.

$$\text{CVSSCE} \overset{def}{=} \text{MCE} + \text{FACE} + \text{RCE} \tag{8}$$

Because it is composed of three different losses that are simply summed, it is relatively trivial to give more importance to one loss, e.g., give more importance to MCE by adding a weight to it if missing attacks is more critical than raising false alarms.

4 Experimental Setup

4.1 Choice of metrics

The finalized set of metrics chosen, both for comparison purposes and validation of the newly introduced metrics, is:

- Common ML metrics: Accuracy, F1-score, TPR (Recall), PPV (Precision).
- Metrics potentially resistant to imbalance: MCC and BI. Both range between -1 and 1.
- Cyber-informed metrics: MC, FAC and CI. The former two range between 0 and 1 while the latter ranges between -1 and 1.

All metrics, except Accuracy and MCC, were computed on a per-class basis. The averaging method retained is macro-averaging, which averages irrespective of the class imbalance to reduce its influence.

4.2 Dataset Pre-processing

All datasets were split using a stratified scheme into 70% train (60% and 10% validation for DNNs) and 30% test sets.

For the UNSW-NB15 dataset, features such as IP addresses, timestamps, *attack_cat* were removed, while categorical features or features having a small number of unique values, were one-hot encoded. The resulting dataset has 229 features.

For the CIC-IDS2017 dataset, two features and 5792 instances were removed because of problematic or missing values. A further eight features were removed because they only had one value. The resulting dataset has 70 features.

For the DAPT2020 dataset, Flow ID, IP addresses and timestamps were removed. Labels used are a combination of both Stage and Activity, e.g., Network Scan Reconnaissance or Malware Download Establish Foothold, for a total of 21 classes. In this way, attack classes represent a single behavior.

4.3 CVSS Scores for Cyber-related Metrics

Ideally, datasets would be constituted with CVSS scores or CVE IDs that represent the exploited vulnerabilities, as it is the case for UNSW-NB15. Fortunately, tools used to generate attacks generally give the CVE IDs of the exploited vulnerabilities, so integrating CVSS scores or CVE IDs in the datasets is often relatively easy.

For the CIC-IDS2017 dataset, DAPT2020, and many other publicly available datasets, there is no such information. In these cases, although the information is missing, it is often possible to manually score attack classes, given they are sufficiently detailed and classes are homogeneous enough, i.e., a class represents very similar attacks. For CIC-IDS2017 and DAPT2020, the attacks were described in the original papers [Sharafaldin et al. 2018, Myneni et al. 2020] and are sufficiently detailed to score attack classes with the CVSS calculator[3]. The vectors used for computation are visible in Table 2. While Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks can have a relatively similar impact, DoS attacks tend to reduce performance of the target until an eventual shutdown, whereas DDoS can quickly make the targeted resource unavailable, thus the difference in impact. Attack classes have been grouped according to their name for better clarity and are visible in Table 2.

[3] https://www.first.org/cvss/calculator/3.1

Attacks	AV	AC	PR	UI	S	C	I	A	CVSS Scores		
									Basic	Env. 2*	Env. 3*
DoS attacks	Network	Low	None	None	Unchanged	None	None	Low	5.3	4.6	6.1
Scan, Patator and Brute Force attacks	Network	Low	None	None	Unchanged	Low	None	None	5.3	6.1	4.6
Web Attack XSS and CSRF	Network	Low	None	None	Unchanged	None	Low	None	5.3	5.3	5.3
Infiltration	Local	High	None	Required	Changed	High	None	None	5.5	6.5	2.9
SQL Injection attacks	Network	Low	None	None	Unchanged	Low	Low	Low	7.3	7.4	7.4
Malware Download, Backdoor	Network	Low	None	None	Unchanged	None	Low	None	5.3	5.3	5.3
DDoS	Network	Low	None	None	Unchanged	None	None	High	7.5	5.7	9.3
Heartbleed, Data Exfiltration	Network	Low	None	None	Unchanged	High	None	None	7.5	9.3	5.7
Botnet, Privilege Escalation	Network	Low	None	None	Unchanged	High	High	High	9.8	9.8	9.8

AV: Attack Vector, AC: Attack Complexity, PR: Privileges Required, UI: User Interaction, S: Scope, C: Confidentiality, I: Integrity, A: Availability.
* Env. 2 and Env. 3 scores are only used for CIC-IDS2017 in subsection 5.2
Env. 2 corresponds to High Confidentiality, Medium Integrity, Low Availability requirements.
Env. 3 corresponds to Low Confidentiality, Medium Integrity, High Availability requirements.
Details about possible values for each category, as well as their signification, can be found at https://www.first.org/cvss/v3.1/specification-document.

Table 2: CVSS scores for CIC-IDS2017 and DAPT2020

To evaluate changes in performance with CVSS-related metrics with CVSS environmental scores, three different ”environments” were considered:

- The basic environment, without any modification.
- A high Confidentiality, medium Integrity and low Availability (Environment 2), e.g., a marketing company’s client database with other backups for data redundancy.
- Low Confidentiality, medium Integrity and high Availability (Environment 3), e.g.,

a video streaming service.

Environment 2 (Env. 2) and Environment 3 (Env. 3) in Table 2 are representative of two different systems with different CIA requirements. Therefore, CVSS scores changes are reported. These changes in CVSS scores will in turn impact the relative performance reflected by CVSS-related metrics to help in selecting the IDSs that are most adapted to a particular environment.

4.4 UNSW-NB15 data subset for CVSSCE

To more effectively evaluate the impact and effectiveness of integrating CVSS scores into the loss formulation, only a subset of the UNSW-NB15 dataset has been retained. This particular choice has been made to focus on the influence of the loss integrating CVSS scores while reducing the impact of other variables. This version of the dataset will thus be used in subsection 5.3. As shown in Table 3, only browser exploits and normal traffic were retained. Browser exploits were then separated into different classes according to their CVSS score, e.g., browser exploits with a CVSS score of 10 belong to class exploits-Browser-10, because they are exploits targeting different vulnerabilities and with potentially different attack mechanisms. By selecting a single attack class separated in multiple sub-classes, traffic should overall be much more homogeneous between attack classes, and as stated previously, should reduce the influence of variables other than the loss using CVSS scores.

Dataset	Number of instances per class	Total
UNSW-NB15	Normal: 2218761, exploits-Browser-10.0: 537, exploits-Browser-9.3: 13988, exploits-Browser-8.5: 232, exploits-Browser-7.6: 233, exploits-Browser-7.5: 1149, exploits-Browser-7.1: 94, exploits-Browser-6.8: 511, exploits-Browser-5.1: 1589, exploits-Browser-5.0: 274, exploits-Browser-4.3: 443	2237811

Table 3: Datasets details

This task is highly difficult for two reasons: classes are even more imbalanced than in the original UNSW-NB15 dataset as normal traffic represents more than 99% of the data, and the different attack classes are very similar, making it much harder to differentiate them. Moreover, one attack class is also much more present than the eight other classes.

4.5 ML algorithms

In order to evaluate the proposed set of metrics and understand the differences brought by the introduction of cybersecurity-based metrics, experiments were run with a wide range of algorithms, trying various hyper-parameter combinations to find the best performing IDS on the two datasets considered. The retained algorithms are:

– A dummy classifier, classifying every instance as of the most frequent class (normal traffic in both datasets) to serve as a baseline.

– Relatively simple algorithms that should give an idea about the complexity of the classification task: Gaussian Naïve Bayes (GNB), Linear Support Vector Classification (LSVC), Decision Trees (DTs).

– More complex algorithms that should reflect the expected performance of IDSs relying on ML: Random Forests (RFs), Multi-Layer Perceptron (MLP), Deep Neural Networks (DNNs).

All algorithms are from the *scikit-learn*[4] library except DNNs that were programmed using the *PyTorch*[5] and *PyTorch Lightning*[6] libraries.

5 Results

In order to evaluate the usefulness of the newly defined metrics, IDSs based on algorithms presented in subsection 4.5 were trained and tested on both the UNSW-NB15 and CIC-IDS2017 datasets. Results for both datasets are presented in Table 4. For each category of ML algorithm, a coarse grid-search scheme was used to pick hyper-parameter values and the IDS obtaining the best results was kept. For those IDSs, results are shown for the retained metrics. Considering only some of the metrics, particularly Accuracy and MCC, it is difficult to see which IDS performs better than others. The most significant differences on both datasets can be seen with PPV and the newly defined metrics.

5.1 Zoom comparison of two IDS' performances

A more significant difference can often be seen with the newly defined metrics. The following example compares results presented in Table 4 for the LSVC (Linear Support Vector Classification) and MLP (Multi-Layer Perceptron) on the UNSW-NB15 dataset.

When looking at the results, the Accuracy of both IDSs is very close, whereas results are very different according to FAC and CI. For the Accuracy, this is understandable because results on most classes are very close. Both IDSs have relatively similar performance (under a 5% difference) on all classes, except Exploits and DoS. LSVC outperforms MLP detecting DoS instances (69% versus 37%). On the other hand, MLP significantly outperforms LSVC for detecting Exploits (74% versus 46%).

In the UNSW-NB15 dataset, for attacks that do have CVE IDs and thus an assigned CVSS score, Exploits is the class with the highest average CVSS score because most instances have a high CVSS score (9.3 or 10). DoS attacks, on the contrary, generally have CVSS scores between 5 and 8. Exploits attacks are generally more dangerous, i.e., have a higher CVSS score, which is directly translated into those two metrics. Indeed, the MLP that performs better on Exploits has results that are more than two times better for FAC and close to 60% better on the CI metric. Furthermore, the relative difference in results of both models is higher for metrics using CVSS than their counterparts (CI vs BI, FAC vs PPV, MC vs TPR), which shows that the MLP-based IDS is globally better at detecting attacks with a higher CVSS.

Operationally, it means the MLP-based IDS will more often detect attacks that are critical and might endanger the system. When using such an IDS, automated mitigation

[4] https://scikit-learn.org/stable/index.html
[5] https://pytorch.org/
[6] https://www.pytorchlightning.ai/

Dataset	Algorithm	Acc.	F1	TPR	PPV	MCC	BI	MC	FAC	CI
UNSW-NB15	**Dummy**	0.873	0.093	0.1	0.087	0	−0.102	0.654	0	0.346
	GNB	0.490	0.130	0.296	0.264	−0.039	−0.253	0.329	0.494	0.175
	LSVC	0.972	0.445	0.436	0.480	0.879	0.394	0.264	0.344	0.391
	DT	0.979	**0.586**	**0.565**	0.666	0.910	**0.535**	**0.186**	0.228	0.584
	RF	**0.981**	0.571	0.549	0.738	**0.919**	0.521	0.205	0.180	0.614
	MLP	0.980	0.520	0.518	**0.772**	0.912	0.488	0.220	**0.153**	**0.625**
	DNN	0.978	0.505	0.511	0.627	0.908	0.480	0.206	0.257	0.535
CIC-IDS2017	**Dummy**	0.803	0.059	0.066	0.053	0	−0.172	0.603	0	0.397
	GNB	0.723	0.499	**0.848**	0.469	0.572	0.579	**0.069**	0.321	0.609
	LSVC	0.986	0.546	0.589	0.602	0.960	0.574	0.256	0.253	0.490
	DT	0.998	0.839	0.843	0.836	0.995	**0.842**	0.101	0.104	0.794
	RF	**0.998**	**0.850**	0.836	0.870	**0.995**	0.834	0.106	0.085	**0.808**
	MLP	0.996	0.725	0.721	0.835	0.989	0.718	0.177	0.103	0.719
	DNN	0.997	0.757	0.739	**0.896**	0.991	0.736	0.168	**0.067**	0.764

Values were truncated to the third decimal. Best results for a given metric and dataset are in **bold**.

Table 4: Performances on the UNSW-NB15 and CIC-IDS2017 datasets

strategies can be used with more confidence, and human operators will be able to divert their energy in investigating other more relevant alarms that might represent previously undetected attacks.

5.2 CVSS' Environmental score

The CVSS Environmental Score allows to define requirements in Confidentiality, Integrity and Availability, and to modify base vector values. In doing so, scores returned by the CVSS scoring system differ from what would be returned without any requirements.

The main advantage of using the Environmental Score is being able to find and differentiate IDSs that might be more adapted to protect a system given its security requirements, when their performance according to common metrics are relatively similar. In Table 5, two very similar DTs were compared using three environments as defined in subsection 4.3. Both are `DecisionTreeClassifier` from the scikit-learn library, and the only difference between both is the criterion used. The first, DT 1, was trained with a Gini criterion while the second, DT 2, was trained with an entropy criterion with

otherwise equal parameters. Both IDSs being very similar, performance is also expected to be very similar.

Environment	Model	TPR	PPV	BI	MC	FAC	CI
Basic Environment	**DT 1**	0.830	0.880	0.828	0.117	0.080	0.802
	DT 2	0.824	0.882	0.822	0.114	0.069	0.815
Env. 2 (High C, Medium I, Low A)	**DT 1**	0.830*	0.880*	0.828*	0.130	0.085	0.783
	DT 2	0.824*	0.882*	0.822*	0.132	0.080	0.786
Env. 3 (Low C, Medium I, High A)	**DT 1**	0.830*	0.880*	0.828*	0.104	0.076	0.818
	DT 2	0.824*	0.882*	0.822*	0.096	0.059	0.843

Values were truncated to the third decimal. C: Confidentiality, I: Integrity, A: Availability.
* TPR, PPV and BI values are equal for all environments.

Table 5: Model performances of two DTs on the CIC-IDS2017 dataset for different environments

Both models results are almost equal on all classes, except for Infiltration (High impact on Confidentiality, DT 1 has 60% Acc. while DT 2 has a 20% Acc.) and Web Attack SQL Injection (Low impact on Confidentiality, Integrity and Availability, DT 1 has 33% Acc. while DT 2 has 67% Acc.). Using common ML metrics, it is uncertain which IDS is the best.

The difference in CVSS scores for both Infiltration (Basic Env.: 5.5, Env. 2: 6.5, Env. 3: 2.9) and Web Attack SQL Injection (Basic Env.: 7.3, Env. 2: 7.4, Env. 3: 7.4) can be read in Table 2. When choosing which IDS to use on a specific system, it is important that its performance in detecting specific attacks aligns with the requirements of the system. For example, when choosing an IDS for a system corresponding to Env. 3, DT 2 seems more suitable since it is better at detecting Web Attack SQL Injection that impacts availability.

In order to fully benefit from the usage of CVSS scores, and create IDSs perfectly adapted to a given system, ML-based IDSs should also be trained by taking into account CVSS scores. More experiments are performed in subsection 5.3 to complete the approach.

5.3 Using CVSS to train Neural Networks

While integrating CVSS scores into evaluation metrics might help in choosing a more adequate IDS, it does not make them better. This is where integrating CVSS score into the training phase of IDSs counts, to build IDSs that will inherently detect more often severe attacks. In order to test the advantages of using a loss integrating CVSS scores, two NNs (Neural Networks) were trained and tested on the dataset reported in Table 3, one with a classic CE (Cross-Entropy loss) as in Equation 4, and the other with the loss using

Training loss	TPR	PPV	BI	MC	FAC	CI
Basic CE	0.258	0.246	0.233	0.579	0.589	−0.169
CVSSCE	0.288	0.360	0.263	0.547	0.488	−0.035

Values were truncated to the third decimal.

Table 6: Performance of NNs trained with a basic CE or with CVSSCE

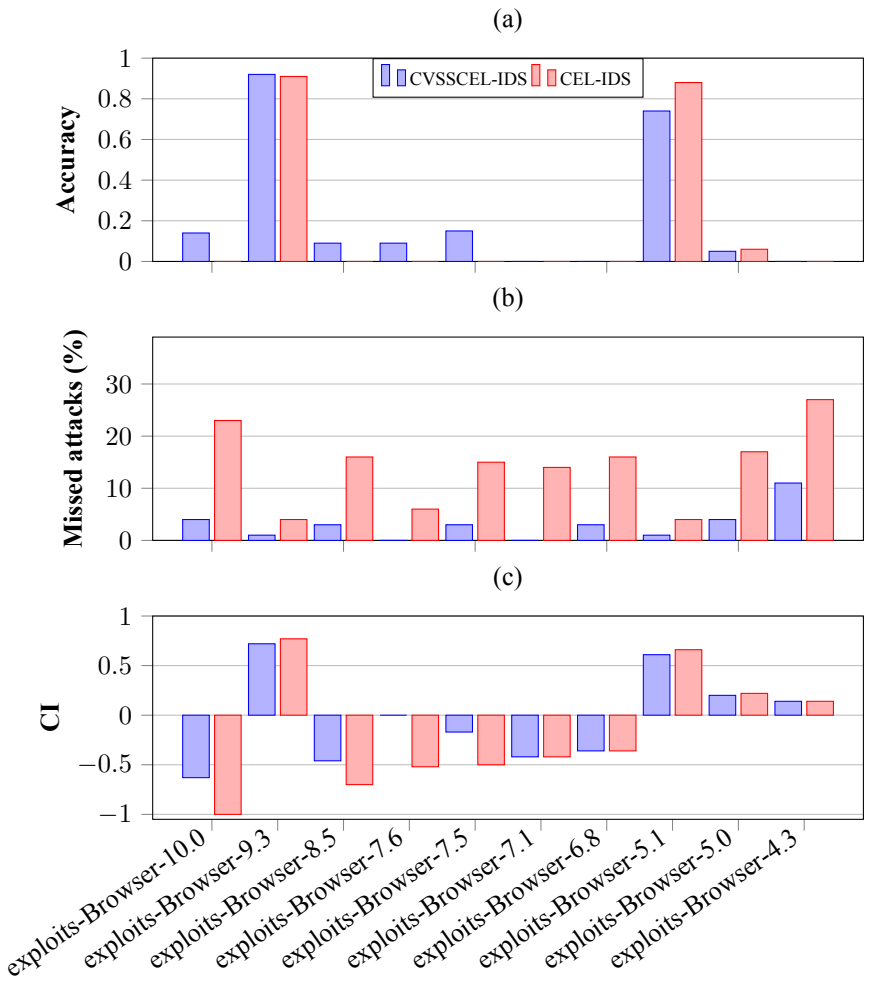

Figure 2: Comparison of two IDSs trained with a CVSSCE (CVSSCE-IDS) and with a basic CE (CE-IDS).

CVSS scores as formulated in Equation 8. The NNs used the exact same architecture as those reported in Table 4, because more complex architectures did not seem to improve performance and instead made the training more unstable.

Results of IDSs trained with either loss are reported in Table 6. Interestingly, only taking classic ML metrics into account, the IDS trained using the loss with CVSS already exhibits a close to 50% higher PPV (Precision), which means the IDS is much more often right when predicting different attacks.

Taking into account CVSS-related metrics, the IDS trained using the loss with CVSS is also clearly more effective in detecting attacks, and possibly more severe attacks.

In order to properly investigate whether training with a loss using CVSS scores do help in detecting more severe attacks, it is required to go over performance of the IDSs on different classes. In Figure 2, results of the two IDSs are shown with regard to Accuracy (a), percentage of completely missed attacks (instances classified as normal for each class) (b) and CI scores (c). For convenience, the IDS trained without CVSS scores will be called CE-IDS, and the one trained with will be called CVSSCE-IDS.

Because exploits-Browser-9.3 and exploits-Browser-5.1 are more frequent than other attack classes, their Accuracy is expectedly higher for both IDSs. However, CVSSCE-IDS has an Accuracy around 10% higher for the four other most severe attacks, which is quite significant considering these attacks are completely undetected by CE-IDS.

Furthermore, almost all attacks were completely missed 10 to 20% less by CVSSCE-IDS, and were often misclassified as exploits-Browser-9.3 if not correctly classified. A possible explanation is that using CVSS scores in the loss formulation ends up penalizing missed attacks (particularly severe attacks) much more, thus forcing the IDS to refine its representation of attack classes more than without using CVSS scores. Results using the CI metric for each class also tend to validate the higher performance of the IDS trained using CVSS scores.

5.3.1 Validation for a more recent attack methodology

To ascertain that the proposed approach is applicable to more recent attack methodologies, experiments performed in subsection 5.3 have been reproduced on the DAPT2020 dataset. NNs having the same architecture were trained using both CVSSCE and CE losses.

Training loss	TPR	PPV	BI	MC	FAC	CI
Basic CE	0.391	0.483	0.362	0.338	0.348	0.314
CVSSCE	0.491	0.317	0.419	0.097	0.453	0.449

Values were truncated to the third decimal.

Table 7: Performance of NNs trained with a basic CE or with CVSSCE on DAPT2020

As shown in Table 7, results obtained validate that NNs trained using CVSS information tend to completely miss attacks much less often, and thus also exhibit a much higher performance with CI. The approach of using CVSS scores inside the loss formulation seems to be even more effective on this dataset that better represents current attack

methodologies. This shows promising prospects with regard to application to real-world scenarios.

Finally, while using a loss based on CVSS might slow down convergence or require a bit more computations, experiments performed on DAPT2020 tend to show a training time overhead of around 10%. On bigger datasets, a similar overhead can be expected. Additionally, since the loss is only used at training time, there is no overhead for inference.

6 Conclusion and Future Work

Using CVSS scores with IDSs seems to be able to fulfill multiple purposes, be it for training to help IDSs detect more severe attacks, or to be integrated in evaluation metrics to find IDSs that are better at detecting severe attacks or more adapted to a particular system. When models show very similar performance on the newly defined metrics, it generally means that their performance is similar on all attacks or that they differ for attacks having similar CVSS scores. The high performance on critical attacks is thus adequately highlighted. Interestingly, some IDSs, although exhibiting relatively poor performance in general, can have an unexpectedly good performance in some aspects, e.g., the GNB-based IDS for CIC-IDS2017 which is the best attack detector at the cost of more false alarms. Thus, using those IDSs could be interesting when implementing ensemble methods for intrusion detection.

Using CVSS scores with environmental parameters also seems to enable building IDSs being specialized for protecting specific systems with different requirements. However, CVSS base vectors of the encountered attacks need to be available to be able to compute modified CVSS scores, thus requiring more information on the attacks.

Integrating CVSS scores into a loss formulation to train NN-based IDSs seems highly effective and does not seem to really suffer from evident drawbacks. A possible shortcoming of using such methods is that integrating CVSS scores into the training of an IDS does not seem as simple with ML methods other than NNs.

CVSS scores are the most often used to describe the impact of exploiting vulnerabilites with regard to the CIA triad. Furthermore, this is a metric used by the cybersecurity community to score every new CVE, which makes it one of the best options in leveraging cybersecurity knowledge to create better IDSs. The only times it might prove less suitable is when there is no immediate impact on the system, e.g., privilege escalation. In these cases, attacks could be attributed a CVSS score depending on what it can enable. For example, privilege escalation can give full access and could therefore be given a high impact on each component of the CIA triad. This reflects the fact that privilege escalation should really be detected, and thus its CVSS score should be high. Although these new metrics are based on a *de facto* cybersecurity standard score, using such standard requires a more consistent effort in data collection to take advantage of CVSS scores to train and find better and more adapted IDSs.

Finally, in a real world scenario, alerts raised by signature-based methods can often be matched to CVE IDs either automatically or by a human expert, which could help create a local training dataset containing CVSS information. In the few cases where this is not possible, e.g., attacks with various capabilities, such as computer viruses, they could nevertheless be split into parts with a single capability as has been done for DAPT2020 or be represented in attack graphs to be properly scored.

In future work, the integration of CVSS score could be adapted to use the version 4.0 of the framework. The usage could also be extended to ICS datasets, which would require a new computation of CVSS scores. It could also be interesting to see if using

CVSS scores greatly impacts IDSs based on ensemble models. Finally, the methodology presented in this paper could be extended to take into account attack graphs, to use CVSS scores in a more comprehensive way.

Acknowledgements

This work is supported by the Chair of Naval Cyber Defence and its partners Ecole Navale, ENSTA-Bretagne, IMT-Atlantique, Naval Group and Thales.

References

[Abadeh et al. 2011] Abadeh, M. S., Mohamadi, H., Habibi, J.: ”Design and Analysis of Genetic Fuzzy Systems for Intrusion Detection in Computer Networks”; Expert Systems with Applications, 38, 6 (2011), 7067-7075.

[Ahmad et al. 2021] Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., Khan, M. S.: ”Intrusion Detection in Internet of Things Using Supervised Machine Learning Based on Application and Transport Layer Features Using Unsw-Nb15 Data-Set”; EURASIP Journal on Wireless Communications and Networking, 10 (2021).

[Aickelin et al. 2013] Aickelin, U., Greensmith, J., Twycross, J.: ”Immune System Approaches To Intrusion Detection - a Review (ICARIS)”, CoRR (2013) http://arxiv.org/abs/1305.7144v1.

[Ashibani and Mahmoud 2017] Ashibani, Y., Mahmoud, Q. H.: ”Cyber Physical Systems Security: Analysis, Challenges and Solution”; Computers and Security, 68, (2017), 81-97.

[Aussibal and Gallon 2008] Aussibal, J., Gallon, L.: ”A New Distributed IDS Based on CVSS Framework”; Proc. IEEE International Conference on Signal Image Technology and Internet Based Systems (2008).

[Axelsson 2000] Axelsson, S.: ”The Base-Rate Fallacy and the Difficulty of Intrusion Detection”; ACM Transactions on Information and System Security, 3, 3 (2000), 186-205.

[Bhuyan et al. 2015] Bhuyan, M. H., Bhattacharyya, D. K., Kalita, J. K.: ”Towards Generating Real-Life Datasets for Network Intrusion Detection”; International Journal of Network Security, 17, (2015), 683-701.

[Bolivar et al. 2019] Bolivar, H., Jaimes Parada, H. D., Roa, O., Velandia, J.: ”Multi-criteria Decision Making Model for Vulnerabilities Assessment in Cloud Computing regarding Common Vulnerability Scoring System”; Proc. Congreso Internacional de Innovación y Tendencias en Ingenieria, Bogota (2019), 1-6.

[Boudermine 2023] Boudermine, A.: ”A dynamic attack graphs based approach for impact assessment of vulnerabilities in complex computer systems”; Institut Polytechnique de Paris, 2022.

[Chicco 2017] Chicco, D.: ”Ten Quick Tips for Machine Learning in Computational Biology”; BioData Mining, 10, 1 (2017).

[Creech and Hu 2013] Creech, G., Hu, J.: ”Generation of a new IDS test dataset: Time to retire the KDD collection”; Proc. IEEE Wireless Communications and Networking Conference (2013), 4492-4497.

[Duraz et al. 2023] Duraz, R., Espes, D., Francq, J., Vaton, S.: ”Cyber Informedness: A New Metric using CVSS to Increase Trust in Intrusion Detection Systems”; Proc. European Interdisciplinary Cybersecurity Conference (2023), 53-58.

[Durumeric et al. 2014] Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., Halderman, J. A.: ”The matter of Heartbleed”; Proc. Conference on Internet Measurement Conference, Vancouver (2014), ACM Publishing, 475-488.

[Faker and Dogdu 2019] Faker, O., Dogdu, E.: ”Intrusion Detection Using Big Data and Deep Learning Techniques”; Proc. ACM Southeast Conference (2019), 86-93.

[Ferrag et al. 2020] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., Janicke, H.: ”Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study”; 50, (2020).

[Frigault et al. 2017] Frigault, M., Wang, L., Jajodia, S., Singhal, A.: ”Measuring the Overall Network Security by Combining CVSS Scores Based on Attack Graphs and Bayesian Networks”; Springer, International Publishing (2017), Network Security Metrics, 1-23.

[Gamage and Samarabandu 2020] Gamage, S., Samarabandu, J.: ”Deep Learning Methods in Network Intrusion Detection: a Survey and an Objective Comparison”; Journal of Network and Computer Applications, 169, (2020), 1416-1426.

[Gao et al. 2018] Gao, N., He, Y., Ling, B.: ”Exploring Attack Graphs for Security Risk Assessment: a Probabilistic Approach”; Wuhan University Journal of Natural Sciences, 23, 2 (2018), 171-177.

[Gharib et al. 2016] Gharib, A., Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A.: ”An Evaluation Framework for Intrusion Detection Dataset”; Proc. International Conference on Information Science and Security (2016), IEEE Publishing, 1-6.

[Gu et al. 2009] Gu, Q., Zhu, L., Cai, Z.: ”Evaluation Measures of the Classification Performance of Imbalanced Data Sets”; Springer, Berlin Heidelberg (2009), Communications in Computer and Information Science, 461-471.

[Hassan et al. 2020] Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., Fortino, G.: ”A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment”; Information Sciences, 513, (2020), 386-396.

[Hindy et al. 2020] Hindy, H., Brosset, D., Bayne, E., Seeam, A. K., Tachtatzis, C., Atkinson, R., Bellekens, X.: ”A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems”; IEEE Access, 8, (2020), 104650-104675.

[Holm 2014] Holm, H.: ”Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?”; Proc. 47th Hawaii International Conference on System Sciences (2014).

[Injadat et al. 2021] Injadat, M., Moubayed, A., Nassif, A. B., Shami, A.: ”Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection”; IEEE Transactions on Network and Service Management, 18, 2 (2021), 1803-1816.

[Jeni et al. 2013] Jeni, L. A., Cohn, J. F., De La Torre, F.: ”Facing Imbalanced Data Recommendations for the Use of Performance Metrics”; Proc. Humaine Association Conference on Affective Computing and Intelligent Interaction (2013), IEEE Publishing, 245-251.

[Kasongo and Sun 2020] Kasongo, S. M., Sun, Y.: ”Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the Unsw-Nb15 Dataset”; Journal of Big Data, 7, 1 (2020).

[KDD Cup 99 Data 1999] KDD Cup 99 Data (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[Khraisat et al. 2019] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: ”Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges”; Cybersecurity, 2, 1 (2019).

[Lakshminarayana et al. 2019] Lakshminarayana, D. H., Philips, J., Tabrizi, N.: ”A Survey of Intrusion Detection Techniques”; Proc. 18th IEEE International Conference On Machine Learning And Applications (2019).

[Lever et al. 2016] Lever, J., Krzywinski, M., Altman, N.: ”Classification Evaluation”; Nature Methods, 13, 8 (2016), 603-604.

[Masdari and Khezri 2020] Masdari, M., Khezri, H.: ”A Survey and Taxonomy of the Fuzzy Signature-Based Intrusion Detection Systems”; Applied Soft Computing, 92, (2020).

[Moualla et al. 2021] Moualla, S., Khorzom, K., Jafar, A.: ”Improving the Performance of Machine Learning-Based Network Intrusion Detection Systems on the Unsw-Nb15 Dataset”; Computational Intelligence and Neuroscience (2021), 1-13.

[Moustafa and Slay 2015] Moustafa, N., Slay, J.: ”UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)”; Military Communications and Information Systems Conference (MilCIS) (2015), 1-6.

[Myneni et al. 2020] Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., Kang, M.: ”DAPT 2020 - Constructing a Benchmark Dataset for Advanced Persistent Threats”; Springer, International Publishing (2020), Deployable Machine Learning for Security Defense, 138-163.

[Roesch et al. 1999] Roesch, M., Stanford Telecommunications: ”Snort - Lightweight Intrusion Detection for Networks”; Proc. LISA’99, 13th USENIX conference on System administration (1999), 229-238.

[Saito and Rehmsmeier 2015] Saito, T., Rehmsmeier, M.: ”The Precision-Recall Plot Is More Informative Than the Roc Plot When Evaluating Binary Classifiers on Imbalanced Datasets”; PLOS ONE, 10, 3 (2015), 1-21.

[Sarker et al. 2020] Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., Ng, A.: ”Cybersecurity Data Science: an Overview From Machine Learning Perspective”; Journal of Big Data, 7, 41 (2020).

[Sharafaldin et al. 2017] Sharafaldin, I., Gharib, A., Lashkari, A. H., Ghorbani, A. A.: ”Towards a Reliable Intrusion Detection Benchmark Dataset”; Software Networking, 1 (2017), 177-200.

[Sharafaldin et al. 2018] Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A.: ”Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”; Proc. 4th International Conference on Information Systems Security and Privacy (2018), 108-116.

[Shiravi et al. 2012] Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A. A.: ”Toward Developing a Systematic Approach To Generate Benchmark Datasets for Intrusion Detection”; Computers and Security, 31, 3 (2012), 357-374.

[Siddique et al. 2019] Siddique, K., Akhtar, Z., Khan, F. A., Kim, Y.: ”Kdd Cup 99 Data Sets: a Perspective on the Role of Data Sets in Network Intrusion Detection Research”; Computer, 52, 2 (2019), 41-51.

[Sokolova et al. 2006] Sokolova, M., Japkowicz, N., Szpakowicz, S.: ”Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation”; Springer, Berlin Heidelberg (2006), Lecture Notes in Computer Science, 1015-1021.

[Song et al. 2011] Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., Nakao, K.: ”Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation”; Proc. First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, ACM Publishing, Salzburg (2011), 29-36.

[Tama et al. 2020] Tama, B. A., Nkenyereye, L., Riazul Islam, S. M., Kwak, K.-S.: ”An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble”; IEEE Access, 8, (2020), 24120-24134.

[Tavallaee et al. 2009] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A.: ”A Detailed Analysis of the KDD CUP 99 Data Set”; Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications (2009).

[Tobi and Duncan 2018] Tobi, A. M., Duncan, I.: ”Kdd 1999 Generation Faults: a Review and Analysis”; Journal of Cyber Security Technology, 2, 3-4 (2018), 164-200.

[Ur-Rehman et al. 2020] Ur-Rehman, A., Gondal, I., Kamruzzaman, J., Jolfaei, A.: ”Vulnerability Modelling for Hybrid Industrial Control System Networks”; Journal of Grid Computing, 18, 4 (2020), 863-878.

[Vinayakumar et al. 2019] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: ”Deep Learning Approach for Intelligent Intrusion Detection System”; IEEE Access, 7, (2019), 41525-41550.

[Wang et al. 2021] Wang, Z., Li, Z., Wang, J., Li, D.: ”Network Intrusion Detection Model Based on Improved Byol Self-Supervised Learning”; Security and Communication Networks (2021), 1-23.

[Xin et al. 2018] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C.: ”Machine Learning and Deep Learning Methods for Cybersecurity”; IEEE Access, 6, (2018), 35365-35381.

[Zhang et al. 2021] Zhang, Z., Zhang, Y., Guo, D., Song, M.: ”A Scalable Network Intrusion Detection System Towards Detecting, Discovering, and Learning Unknown Attacks”; International Journal of Machine Learning and Cybernetics, 12, 6 (2021), 1649-1665.

[Zhu 2020] Zhu, Q.: ”On the Performance of Matthews Correlation Coefficient (MCC) for Imbalanced Dataset”; Pattern Recognition Letters, 136, (2020), 71-80.