Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5933

Contextual CVSS Scoring Accounting for Vulnerability Batches

Lucas Guimarães Miranda [**Universidade Federal do Rio de Janeiro** | ***lucas.gm@ufrj.br***]
Lucas Senos Coutinho [**Universidade Federal do Rio de Janeiro** | ***lscoutinho@cos.ufrj.br***]
Daniel Sadoc Menasché [**Universidade Federal do Rio de Janeiro** | ***sadoc@ic.ufrj.br***]
Gaurav Kumar Srivastava [**Siemens Corporation** | ***srivastava.gaurav@siemens.com***]
Anton Kocheturov [**Siemens Corporation** | ***anton.kocheturov@siemens.com***]
Enrico Lovat [**Siemens Corporation** | ***enrico.lovat@siemens.com***]
Abhishek Ramchandran [**Siemens Corporation** | ***abhishek.ramchandran@siemens.com***]
Tobias Limmer [**Siemens Corporation** | ***tobias.limmer@siemens.com***]

✉ *Institute of Computing, Universidade Federal do Rio de Janeiro, Av. Athos da Silveira Ramos, 274 - Cidade Universitária, Rio de Janeiro, RJ, 21941-916, Brazil.*

Received: 15 April 2025 • **Accepted:** 06 October 2025 • **Published:** 16 December 2025

Abstract Software vulnerabilities are intrinsically related to product characteristics. The properties of a vulnerability, along with its severity, must be assessed in the context of the product wherein the vulnerability is located. In this paper, our goal is to determine how context impacts severity. To this aim, we pose the following questions: 1) How do different sources statistically differ in the way they parametrize severity? 2) Are there latent patterns that can be learned to determine how context impacts severity? 3) How do vulnerability batches shape scoring practices across sources? To answer these questions, we leverage public data from the National Vulnerability Database (NVD). By comparing CVSS ratings reported by different sources, we provide insights into how scores are parametrized considering contextual factors. For the first question, we show that Industrial Control System (ICS) products tend to have higher attack complexity and more restrictive attack vectors than their general counterparts. For the second, we show that a Large Language Model, CVSS-BERT, can learn context-specific CVSS scores from vulnerability descriptions, achieving F1 scores above 90% and enabling knowledge transfer across sources. For the third, we show that while NVD often assigns uniform scores within a batch, CNAs introduce context-specific variations. These findings highlight the importance of context in assessing severity and suggest the feasibility of semi-automated, batch-aware vulnerability assessments.

Keywords: Vulnerability severity, CVE, CVSS, BERT, Data mining

1 Introduction

The severity of vulnerabilities is intrinsically related to context. Context, in turn, is a broad concept, encompassing how systems interface with each other, their exposure, and maturity level Maidl *et al.* [2021, 2019].

The Common Vulnerability Scoring System (CVSS) is the *de facto* standard for software vulnerability severity assessment. Each vulnerability is rated with a CVSS score, between 0 and 10. The level of agreement among experts and non-experts concerning CVSS score assessments is typically high Massacci [2024]; Allodi *et al.* [2020]; Allodi and Massacci [2014], and the National Vulnerability Database (NVD) provides CVSS scores using its own security experts to assess vulnerability severity. Such assessment, however, cannot accommodate all context-specific factors.[1]

Each vulnerability at NVD is associated with a Common Vulnerabilities and Exposures (CVE) identifier. Each CVE, in turn, receives one or more CVSS scores, which are issued by CVE Numbering Authorities (CNAs). CNAs include vendors, free software developers, security organizations and governmental agencies. Each CNA may issue context-specific CVSS scores, given its knowledge about the context of a vulnerability affecting a product of interest.

Challenge. CVSS ratings should indicate the severity of a vulnerability embedded in a given context. Given a product containing a vulnerability, the vulnerability context matches the whole product and the resulting CVSS rating should reflect as closely as possible the actual severity for a user of the product. Users, however, may take a product as a black-box and may not be able to assess the internals of its operation. Product developers, in contrast, have all the necessary information about how a vulnerability is behaving inside the considered product, but may have no information about alternative products wherein the vulnerability is found.

Goals and insight. In this paper, our goal is to determine how context impacts vulnerability severity. To this aim, we pose the following three research questions:

- **RQ1:** How do different sources (CNAs) *statistically* differ in the way they parametrize severity?

[1] According to FIRST, CVSS is a severity measure, not a risk measure, because it cannot accommodate the value of assets. In this paper, we follow this terminology. `https://www.first.org/cvss/v3.1/user-guide`

- **RQ2:** Are there latent patterns that can be *learned* to determine how context impacts severity?
- **RQ3:** How do CNAs adapt CVSS subscores within and across vulnerability batches, and to what extent does context-specific factors influence scoring decisions at both levels?

Our insight consists of using public data from NVD to capture context-specific information. Our key assumption is that different CNAs capture aspects specific to how the vulnerability is embedded into products. By comparing how CNAs rate vulnerabilities against how NVD issues those rates in its public portal, we aim to provide statistical insights on how context impacts severity, therefore addressing our first question. By focusing on information explicitly provided by NVD, we simplify the data gathering process and guarantee reproducibility.[2]

To address our second research question, we leverage a Large Language Model (LLM), CVSS-BERT Shahid and Debar [2021], for the task of learning context-specific CVSS 3.x scores for vulnerabilities from their descriptions. CVSS-BERT achieves F1 scores above 90% for that task, suggesting that there are indeed latent patterns that determine how context impacts severity. Such patterns, in turn, can be partially interpreted by analyzing keywords that impact the outcomes produced by CVSS-BERT.

Prior art. There has been previous work on discrepancies in vulnerability ratings, both in white papers[3] and research works Le and Babar [2022]; Croft *et al.* [2022, 2023]; Massacci [2024]; Wunder *et al.* [2023]; Human Factors in Security and Privacy Group [2024]. Nonetheless, none of those works considered a broad perspective towards divergences between CNAs and NVD. In addition, there have been previous efforts to account for environmental aspects while assessing CVSS scores Maidl *et al.* [2021, 2019], but none of those efforts leveraged large public datasets for this matter.

This paper extends our previous work presented in Coutinho *et al.* [2024]. The key novelty lies in organizing vulnerabilities into batches, usually grouped by CNA, which enables a more systematic analysis of divergences between CNAs and the NVD. This batch-based approach allows us to investigate discrepancies both within individual batches and across different batches. It also provides a structured framework for identifying consistent patterns and contextual influences on CVSS assessments, specially when accounting for the impact of CNAs.

Contributions. We (partially) answer our research questions by reporting key takeways in Sections 4, 5 and 6, addressing **RQ1**, **RQ2** and **RQ3**, respectively. In summary, our contribution is threefold:

Measurement of context-specific CVSS assessment. We conduct an analysis of how CVSS scores vary across CNAs. In particular, we show that Industrial Control Systems (ICS) products tend to have higher attack complexity and restrictive attack vectors when compared against their general counterparts (Section 4.3). We also indicate that CVSS score differences are increasing over time (Section 4.4) and that when CNAs report alternative scores, those scores usually tend to deviate from NVD in a similar fashion (Section 4.5).

Context-specific learning of CVSS scores. We leverage a LLM, CVSS-BERT, to learn context-specific CVSS scores for vulnerabilities from their descriptions. The feasibility of such learning, with F1 scores above 90%, indicates that CNAs follow consistent patterns, and that there is potential for knowledge transfer across different sources within NVD (Section 5).

Impact of batches. We introduce a batch-level perspective on vulnerability disclosure and CVSS scoring. By grouping CVEs into batches—often corresponding to coordinated disclosures by CNAs—we uncover how context-specific elements manifests collectively. Our analysis shows that batches are a structural feature of modern disclosure, with their prevalence increasing over time (Section 6.2). We demonstrate that batch formation enables a more explainable analysis of CVSS discrepancies: while NVD tends to assign uniform scores across a batch, CNAs often introduce variation reflecting context-specific nuances (Section 6.4). This contribution provides novel evidence that CVSS divergences are not random but emerge from systematic CNA practices within and across batches.

Outline. The next section introduces basic background, and Section 3 describes the considered dataset. Section 4 aims at answering our first question, reporting how CNAs statistically differ in the way they parametrize vulnerability severity. Section 5 focuses on our second question, showcasing the use of LLM to leverage latent patterns in the textual description of vulnerabilities, for the purpose of learning context-specific CVSS subscores. Section 6 reports batch analysis. Section 7 presents related work, and Section 8 concludes.

2 Background and Terminology

We begin this section by introducing the relevant terminology, followed by concrete examples of vulnerabilities that received varying severity scores in different contexts. In Section 2.3, we report the prevalence of context-specific CVSS scores. Section 2.4 extends this discussion by exploring the relevance of context-specific scores, while Section 2.5 delves into the hierarchical nature of context.

2.1 Terminology

In this section, we introduce basic terminology FIRST [2024].

CVE ID is the Common Vulnerabilities and Exposures (CVE) identifier, a unique ID used to refer to a vulnerability.

CVSS is the Common Vulnerability Scoring System and provides a way to capture the principal characteristics of a vulnerability, producing a numerical score, between 0 and 10, reflecting its severity. The numerical score can then be translated into ranges (low, medium, high, and critical) to help organizations properly assess and prioritize their vulnerability management processes. It consists of three scores: Base Score, Temporal Score, and Environmental Score. In this work, we focus on the Base Score.

[2] Source code to reproduce results presented in this work are available at `https://tinyurl.com/cvsscnadata`

[3] `https://www.darkreading.com/application-security/discrepancies-discovered-in-vulnerability-severity-ratings`

CVSS subscores comprise the CVSS score. They are divided into two categories: impact and exploitability. **Impact metrics** assess the potential consequences of a successfully exploited vulnerability on Confidentiality, Integrity, and Availability. **Exploitability metrics** include Attack Vector, Attack Complexity, Privileges Required, User Interaction, and Scope. The subscores are subsequently combined to calculate the final CVSS score using the CVSS calculator, which applies an equation derived from regression analysis and expert knowledge. This calculator is publicly available at FIRST [2024].

CVE Numbering Authority (CNA) is an authorized entity with a specific scope and responsibility to regularly assign CVE IDs and publish corresponding CVE Records.

Root CVE Numbering Authority (Root CNA) is a CNA responsible for coordinating information among multiple CNAs.

CVE Numbering Authority Last Resort (CNA-LR)[4] is a CNA authorized by a Root to assign CVE IDs and to publish corresponding CVE Records within that Root's scope for vulnerabilities not covered by the scope of another CNA. When a CVE is out of the scope of existing CNAs, and requires an alternative CVSS score to contemplate a context-specific feature, this alternative CVSS score is marked at NVD as being issued by a CNA-LR. In this work, we consider a single notable CNA-LR, namely, MITRE.[5]

Vendor: some CNAs are software vendors. Others are public institutions or security-oriented organizations. In the remainder of this paper, as all the entities that issue CVSS scores are either CNAs or the CNA-LR (Mitre), we refer to CVSS sources collectively as CNAs, not distinguishing between vendors and non-vendors.

CVSS Environmental score is an extension of the CVSS Base score. It serves to contemplate environment-specific issues related to vulnerabilities. It refers to the environment as perceived by users, as opposed to the context as perceived by users and product developers. Environmental metrics enable the customization of CVSS scores, in terms of complementary/alternative security controls in place. The metrics are the modified equivalent of base metrics.

There is no public data on environmental scores. Indeed, the environment around a vulnerability, as defined above, is usually a sensitive information, that is accessible only to users. In this work, we refer to product-specific features as contextual features. Environmental CVSS modifiers are out of the scope of this paper.

2.2 Concrete examples

2.2.1 CVE-2021-44228: Log4shell and Siemens TIA portal

We divide software modules into layers. Siemens TIA portal is an example of a product that has multiple layers of modules stacked into it. The assessment of a vulnerability at a given layer, say, at the network layer, can leverage information about the layers around it to determine its severity. The CodeMeter Keyring for TIA Portal, for instance, had its password manager affected by log4shell. Given that the attacker needs a valid client certificate to leverage the vulnerability, in this context the CVSS of CVE-2021-44228 is assessed as 8.6 (high). In the same advisory issued by Wibu, CVE-2021-44228 is assessed as 9.5 (critical) when considered in the context of CodeMeter Cloud Lite.[6] This is because in the latter context, the attacker does not need any credentials to leverage the vulnerability. More generally, NVD assesses the same vulnerability with a CVSS score of 10, given that there are a number of contexts wherein the severity of log4shell is maximum. In the absence of contextual information, NVD issues the vulnerability a severity under its worst case context.

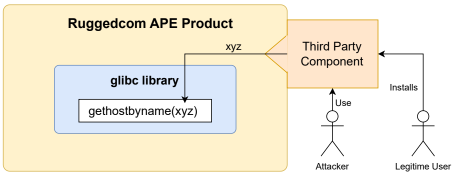


Figure 1. Ghost vulnerability affecting Siemens Ruggedcom: diagram inspired by a misuse case under the SQUARE methodology Mead and Stehney [2005].

2.2.2 CVE-2015-0235: Ghost

As another example, Siemens Ruggedcom APE is vulnerable to the Ghost vulnerability (CVE-2015-0235). Siemens scored it as medium severity, with CVSS of 5.5,[7] whereas NVD scored it as critical, with CVSS 10.0. The major differences rely on the Access Vector (AV), which is set as local by Siemens and network by NVD, and confidentiality and integrity impacts, which are set as high by NVD but none by Siemens. The vulnerability leverages a bug in the `gethostbyname` function, offered by glibc. Ruggedcom does not have an explicit call to that function, but it allows the installation of third party modules. Therefore, if a legitimate user has installed components that utilize the vulnerable functions, and that are accessible to the attacker, the system would be vulnerable (Fig. 1). Alternatively, the attacker would need to have local access to Ruggedcom in order to be able to install third party components, justifying the discrepancy between attack vectors between Siemens and NVD.

2.2.3 CVE-2022-37434: Zlib

A Zlib vulnerability can influence Apache HTTP Server, which may be embedded inside a device that produces an advisory. The NVD score for the Zlib vulnerability may differ from the device's advisory, but both reference CVE-2022-37434. As another example, OpenSSL also uses Zlib, but it does not use the Zlib library for gzip compression and decompression, which is one of the necessary conditions for exploiting the CVE-2022-37434 vulnerability. Alternatively, a Linux distribution, such as Suse, is also impacted by the same CVE, but assigns it a score of 8.1 as opposed to 9.8 from NVD, due to Attack Complexity (AC) which is rated as

[4] `https://www.cve.org/ResourcesSupport/Glossary?activeTerm=glossaryCNALR`

[5] `https://www.cve.org/ProgramOrganization/Structure`

[6] `https://cdn.wibu.com/fileadmin/wibu_downloads/security_advisories/Advisory_WIBU-211213-01.pdf`

[7] SSA: `https://cert-portal.siemens.com/productcert/pdf/ssa-994726.pdf`

Table 1. Distribution of CNAs and their vulnerabilities at NVD and CVSS vector divergences between NVD and CNAs.

year	CNAs	CVEs	CVSS Vector divergences	
			number	**fraction**
2015	1	1	1	1.0000
2016	1	4	3	0.7500
2017	7	103	64	0.6214
2018	20	1094	680	0.6216
2019	39	786	482	0.6132
2020	54	2846	1467	0.5155
2021	82	5417	2977	0.5496
2022	141	7156	4673	0.6530
2023	218	13982	10001	0.7153
2024	134	3167	2298	0.7256

high by Suse but low by NVD.[8]

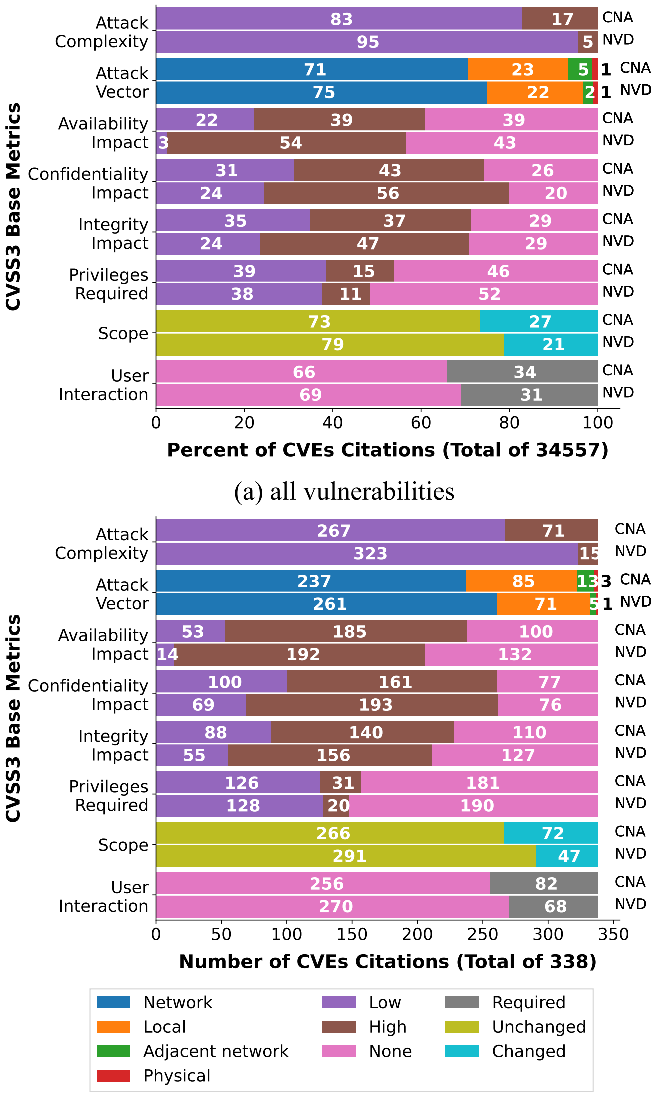
2.2.4 Remarks

In the above examples we considered how the stacking of distinct software modules impacts severity. Although the problems faced in the analysis of deployments, e.g., factory versus public spaces, are similar in spirit to the ones presented above, data about deployments is very sensitive, hence scarce. Therefore, our data-driven analysis of context-specific CVSS scores focus primarily on stacked software modules assessed by different CNAs.

2.3 Prevalence and relevance

The prevalence of CNAs and alternative CVSS scores, as reported by NVD, has been increasing over the years. Table 1 indicates that the number of CNAs that report CVSS scores at NVD increased from 1 to 218 between 2016 and 2023, and the percentage of CVSS scores that diverge between NVD and CNAs remained greater than 50%, reaching 70% in 2023.[9] Indeed, the number of vulnerabilities whose CVSS vectors were reported by multiple CNAs increased from 4 to 13,261. Among the CVSS scores reported in 2023, 9,435 diverged in at least one subscore.

Figure 2(a) illustrates the discrepancies between CVSS subscores, comparing NVD CVSS scores against CNAs alternatives. We distinguish between general vulnerabilities, affecting products across multiple CNAs, and ICS related vulnerabilities, affecting ICS vendors. Figure 2(a) It indicates, for instance, that the CNA attack complexity is higher than NVD for 13% of the instances ($18\% - 5\%$) while accounting for products specificities. Figure 2(b) further indicates that for ICS vulnerabilities the attack complexity as reported by CNAs is assumed to be even higher, as for 51 CVEs ($62 - 11$) affecting ICS products, corresponding to 25% of the ICS vulnerabilities considered in our analysis ($51/203$) the complexity increases from low to high.



(a) all vulnerabilities

(b) ICS vulnerabilities

Figure 2. Discrepancies among CVSS subscores between NVD and multiple CNAs, accounting for all CVEs that have at least two CVSS scores

2.4 Beyond one-size-fits-all

In this section, we discuss some of the reasons why context-specific CVSS scores are key for a robust, decentralized and economically viable cybersecurity strategy. In particular, we highlight the necessity for CNAs to assume greater responsibility, moving away from a centralized model where NVD is the key actor into a more scalable ecosystem (see HackRead [2024]).

As indicated by the NVD, there is a need to cope with the challenge of dealing with a growing backlog of vulnerabilities submitted to the NVD.[10] One solution is to establish a consortium of stakeholders that can collaborate on research to improve the NVD.[11] Each stakeholder becomes responsible for a set of products and corresponding vulnerability contexts,

[8] `https://www.suse.com/security/cve/CVE-2022-37434.html`

[9] NVD reports two CVSS scores for the same vulnerability whenever a source motivates so. CNAs have been individually reporting CVSS scores for a long time, and those alternative scores are now being included in NVD more often. As an example, Siemens started issuing their own CVSS scores since 2011, but NVD only embraced alternative scores much later. To obtain alternative CVSS scores from old Siemens CVEs, one needs to resort to Siemens Security Advisories (SSAs). The same applies to other CNAs.

[10] Announcement on April, 2024: `https://nvd.nist.gov/general/news/nvd-program-transition-announcement`

[11] `https://www.bitsight.com/blog/evaluating-dependence-on-nvd`

and resulting context-specific CVSS scores.

In what follows, we indicate some of the advantages of this approach, which naturally aligns with context-specific CVSS scores.

Economic efficiency: Centralizing vulnerability reporting and scoring in the NVD poses significant economic and timing challenges, given the sheer volume of vulnerabilities and the limited resources available to assess them. A decentralized approach, where CNAs contribute by reporting and scoring vulnerabilities for their products, distributes the workload, reducing bottlenecks and ensuring more timely updates.

Responsiveness: The exponential growth in the number and complexity of cyber threats necessitates a scalable cybersecurity infrastructure. By adopting a federated model where CNAs take on the responsibility of assessing and reporting vulnerabilities specific to their products, the ecosystem can grow and scale more effectively. This approach alleviates the burden on a single centralized authority like the NVD and ensures a more agile and responsive security posture across the industry.[12]

Elements involved in vulnerability detection and assessment: Vulnerability monitoring and assessment involve different tools at different levels. At the product level, Software Bill of Materials (SBOM) contains the details of all components that make up a software application. Given those components, Vulnerability Exploitability eXchange (VEX) can be used to specify corresponding vulnerabilities. Given the vulnerabilities, Cyber Resilience Act (CRA) determines the associated risks, which are then framed under regulations which determine which actions must be taken.

Coverage (breadth) and insight (depth): The current centralized model captures only the "tip of the iceberg" when it comes to vulnerabilities, with many going unreported or underreported. A federated approach encourages a more comprehensive reporting landscape, where CNAs, being closer to their products, can offer deeper insights into specific vulnerabilities. This ensures a more thorough understanding of the cybersecurity threats landscape, paving the way for more timely, robust and effective defense mechanisms.

2.5 Hierarchical nature of context

Context is hierarchical, and product components also follow a hierarchy. We distinguish the following contextual levels, varying from more to less restrictive:

1. **software modules:** context of software modules shipped together with the vulnerability, and that are installed in a product that contains the vulnerability[13]
2. **local operating system:** context of operating system
3. **networked systems:** other systems around the system being considered, including firewalls
4. **solution and deployment (in operation):** devices usually deployed in a factory or deployed in public spaces correspond to different contexts Maidl *et al.* [2021].

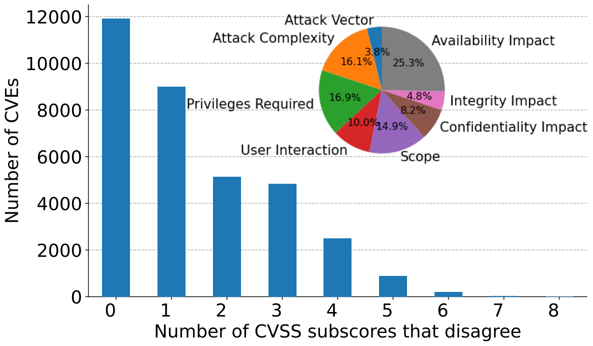


Figure 3. CVSS subscore differences. The pie chart shows the prevalence of subscore disagreements, given that there is a single CVSS subscore disagreement between NVD and CNA.

As pointed out in Section 2.2.4, our data-driven analysis of context-specific CVSS scores focus primarily on stacked software modules assessed by different CNAs, as discussed in the sequel.

2.6 Who issues CVSS scores?

In theory, any entity can issue a CVSS score towards a CVE. However, NVD publishes at most two CVSS scores per CVE. One of the scores is issued by its team of experts. The other score can be issued by a CNA or CNA-LR. The scores may differ for multiple reasons, including lack of information or context-specific characteristics.[14]

One of the most prevalent sources of alternative CVSS scores is MITRE. This is because MITRE is a CNA-LR, being a last resort, for those who are not CNAs, to publish alternative CVSS vectors (see Table 2).

Github is another important source, and counts with two CNAs. One of the CNAs discusses Github Enterprise Server issues only.[15] The other Github CNA covers CVEs requested by software maintainers using the Github Security Advisories feature. In the rest of this paper, Github refers to the latter.

3 Dataset

We collect data from NVD, on July 28, 2024. Our data comprises CVEs published from 2015 to 2024. We focus only on vulnerabilities for which there are two CVSS scores.

We analyze CVSS subscores (see Section 2.1), together with the CNAs reporting those scores. We found a total of 34,557 CVEs with two CVSS scores, regardless of whether they are equal or not. Out of those, 22,646 have at least one CVSS subscore that differs between the two sources. Except otherwise noted, we consider CVSS 3.x, as the fraction of vulnerabilities with CVSS 2.0 or 4.0 is negligible.

We distinguish results for all vulnerabilities and ICS-related vulnerabilities. When discussing ICS products, we focus on the following CNAs: ABB, Hitachi, Honey Well and Rockwell Automation. Siemens is separately considered in Section 4.5.

[12] `https://www.linkedin.com/pulse/security-industry-depends-nvd-patrick-garrity--jwq9c`

[13] We may have multiple layers of modules stacked in software products.

[14] See `https://nvd.nist.gov/general/FAQ-Sections/CVE-FAQs`

[15] `https://www.cve.org/PartnerInformation/ListofPartners/partner/GitHub_P`

Table 2. Number of vulnerabilities per CNA, along with counts of vulnerabilities where the CVSS scores assigned by the CNA were equal to, higher than, or lower than those in the NVD.

CNA	CNA Contact	Number of Vulnerabilities	CNA = NVD	CNA > NVD	CNA < NVD
Github	security-advisories@github.com	4481	1510	1114	1857
Vuldb	cna@vuldb.com	2659	89	41	2529
Patchstack	audit@patchstack.com	2135	227	1100	808
Cisco	ykramarz@cisco.com	1488	829	241	418
Wordfence	security@wordfence.com	1056	481	437	138
IBM	psirt@us.ibm.com	1002	445	157	400
MITRE	cve@mitre.org	969	283	320	366
DHS	ics-cert@hq.dhs.gov	937	438	210	289
EMC	security_alert@emc.com	741	244	183	314
RedHat	secalert@redhat.com	735	289	108	338
Siemens (all SSAs)	productcert@siemens.com	1846	1268	169	330
Siemens (from NVD)	productcert@siemens.com	308	177	51	80

In Section 4.5, we consider the analysis of three distinct CVSS scores towards each CVE. This extension is beyond the scope of NVD, as NVD reports only up to two CVSS vectors per CVE, and one of the vectors is always provided by NVD experts. To accommodate this extension, which requires specialized analysis of security advisories, we consider Siemens Security Advisories (SSAs). We parsed SSAs and collected CVSS scores directly from the SSAs. In Section 4.5 we report our results, involving the comparison of NVD against an SSA against another CNA.

4 The Landscape of CVSS Divergences

In this section, we indicate how different factors impact CVSS scores across CNAs (Section 4.1), and report statistics on CVSS score differences (Section 4.2), accounting for ICS products (Section 4.3), longitudinal analysis (Section 4.4) and conclude with an analysis beyond pairwise comparisons (Section 4.5).

4.1 Availability impact is more sensitive to context than exploitability factors

Recall from Section 2 that CVSS subscores comprise impact and exploitability metrics. In this section, we discuss which metrics are found to be more sensitive to context.

Figure 3 shows a substantial agreement between CVSS subscores as assessed by NVD and by CNAs. Indeed, 11,911 out of 34,557 CVEs have CVSS scores fully agreeing between CNAs and NVD. When CVSS scores disagree, the disagreement is usually about one subscore. As shown in the pie chart of Figure 3, availability impact is the most common source of disagreement. Indeed, depending on the purpose of the product being considered, availability impact may differ. Attack vector, in contrast, which is an exploitability metric, shows more agreement. It differs for 3.9% of CVEs when accounting for CVEs for which there is a single CVSS subscore disagreement (see pie chart in Figure 3) and for 7% of the considered CVEs, when accounting for all CVEs (see Figure 2(a)).

Figures 4 and 5 further support this claim, showing the normalized mutual information (NMI) and Pearson correlation between CVSS subscores as assessed by NVD and by CNAs. The figures indicate that the NMI tends to be higher for exploitability metrics, and lower for impact metrics. Note that whereas NMI is insensitive to the way in which CVSS values are coded, correlation is sensitive to coding. Therefore, to produce Figure 5, for each subscore we coded each of its metric values as a constant, following CVSS 3.x specification (e.g., for Attack Vector, the alternatives Network, Adjacent, Local and Physical correspond to constants 0.85, 0.62, 0.55 and 0.2, respectively).[16]

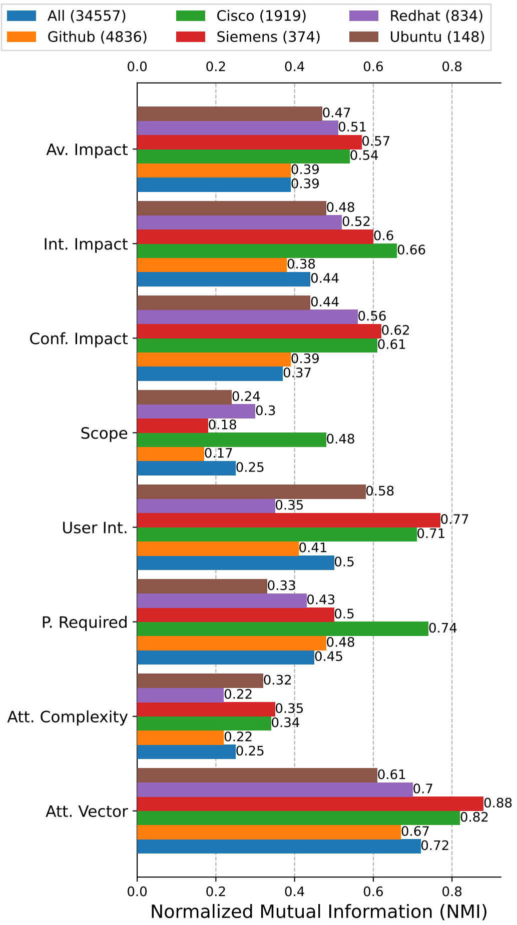


Figure 4. Normalized mutual information value between NVD and CNAs

Interestingly, despite the fact that Attack Complexity

[16] https://www.first.org/cvss/v3.1/specification-document

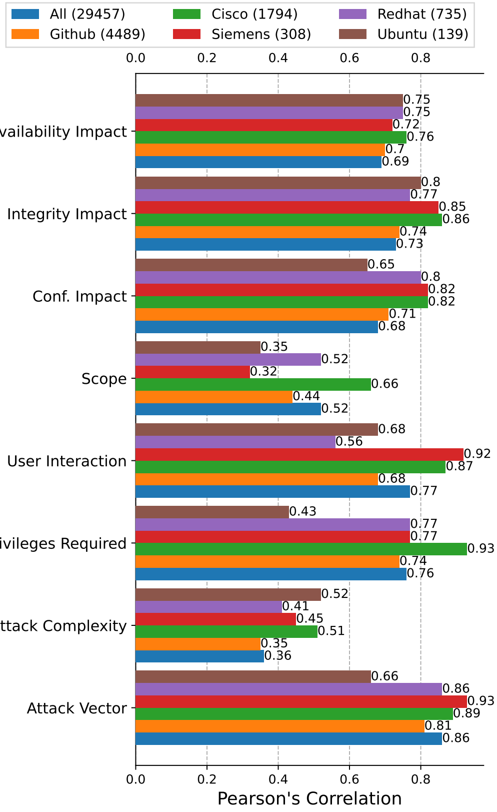


Figure 5. Pearson correlation value between NVD and CNAs

showed a high agreement score between NVD and CNAs, the NMI and correlation coefficients were relatively small (0.24 and 0.36, respectively). This occurred because 80% of the vulnerabilities have a low Attack Complexity, implying that the entropy of Attack Complexity is low, which in turn produces low NMI and correlation values. For all other CVSS subscores, correlation between NVD and CNAs is above 0.69 when accounting for all vulnerabilities (blue bars in Figure 5).

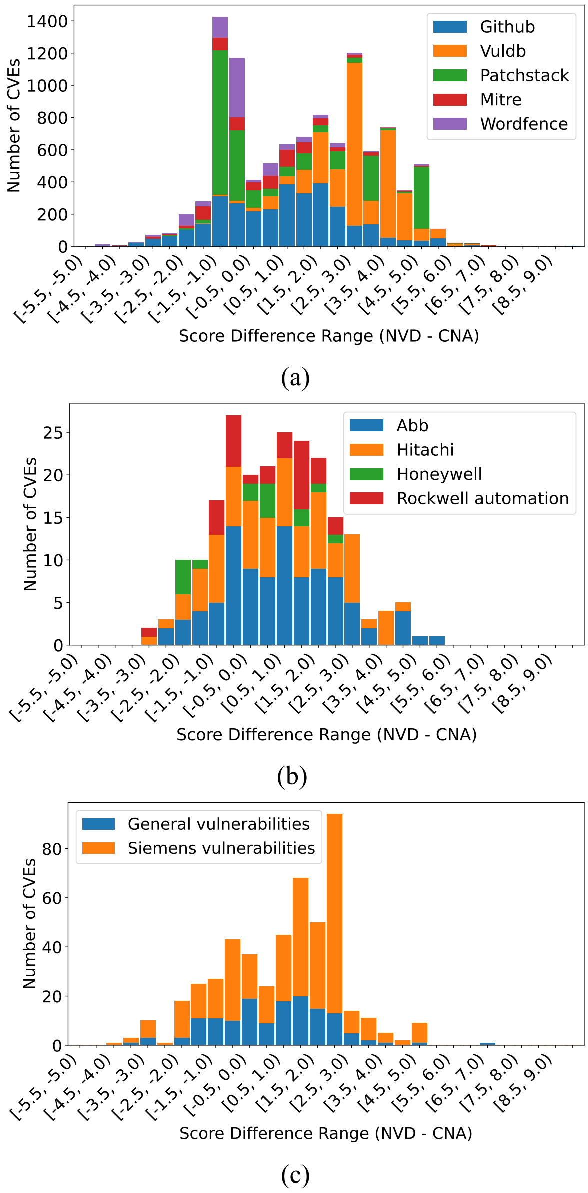
The high correlations between NVD and CNA assessments, as observed in Figure 5, are in agreement with the fact that Large Language Models (LLMs) can be used to transfer knowledge between NVD and CNAs. We will further delve into those aspects in Section 5, where we leverage LLMs to automatically determine subscores using vulnerability descriptions.

Key Takeaway. *Availability impact shows greater variability and context sensitivity compared to exploitability metrics, resulting in higher disagreement between NVD and CNA assessments.*

4.2 What are the statistics of contextual CVSS score differences?

In this section, we consider the statistics of the differences between CVSS scores as reported by NVD and CNAs. To this aim, we take the difference between base CVSS scores as reported by NVD minus the score as reported by a CNA (NVD−CNA).

In Figure 6(a) we apply the analysis to the top 5 CNAs with the highest number of vulnerabilities featuring a CVSS vector inconsistent with the NVD (a total of 8,877 vulnerabilities). As depicted in Figure 6(a), the differences between scores for some CNAs roughly follow a bell-shaped distribution. This reflects the fact that those differences are affected by many complex factors, and the chances that all factors favor an increase or decrease in CVSS scores are small compared to the chance that they (partially) cancel out each other. This is particularly evident for Github.



(a)

(b)

(c)

Figure 6. Differences between scores among (a) top 5 CNAs across all CNAs, (b) ICS CNAs, (c) differences between scores (NVD − Siemens), accounting for vulnerabilities appearing in Siemens Security Advisories (SSAs) where there is discrepancy between Siemens and NVD.

We repeated the above analysis for four ICS CNAs: Abb, Hitachi, Honeywell, and Rockwell Automation. Siemens is treated separately, as indicated below. Figure 6(b) reports our results. The previously observed bell-shaped tendency is now more evident, both in the collective analysis and when each CNA is considered independently. We conducted statistical hypothesis tests to assess normality, with the null hypothesis being that the considered distribution is Gaussian. Results accounting for all vulnerabilities failed to reject the null hypothesis, suggesting a plausible adherence to a normal distribution. The Shapiro-Wilk test, for instance, considers as null-hypothesis that the data follows a normal distribution,

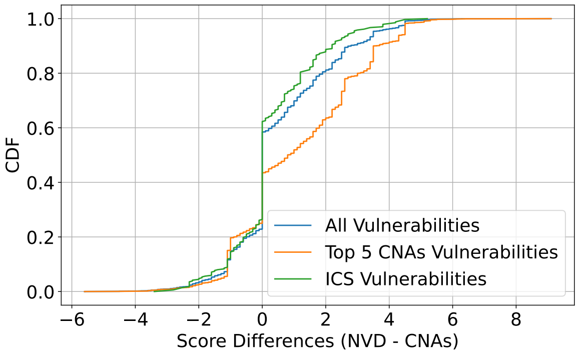


Figure 7. CDF of score differences, accounting for ICS against other CNAs.

and produces p-value of 0.24, which exceeds the significance threshold of 0.05, indicating the lack of evidence to refute the null hypothesis. However, this pattern does not hold for all sources; Honeywell and Rockwell Automation, for instance, significantly deviate from this behavior when examined in isolation. Whereas the former is slightly skewed towards higher CVSS scores reported by the CNA, the latter is skewed towards NVD reporting higher CVSS scores.

We also repeat the above analysis for Siemens Security Advisories (SSAs). Figure 6(c) reports our results. In this figure, we distinguish between general vulnerabilities, that affect Siemens products as well as products from other vendors, and Siemens only vulnerabilities, that contain only Siemens products in their lists of affected products at NVD.

Note that whereas Figures 6(a) and 6(b) are obtained exclusively with data from NVD, Figure 6(c) is obtained using data from the Siemens Portal (details in Section 3). Most CVSS scores as reported in SSAs agree with NVD. In addition, comparing Figure 6(c) against Figure 6(b) we note that there are many more CVEs in the former; this is because at SSAs Siemens can report CVSS values for as many CVEs as needed, whereas to report CVSS scores at NVD there must be a formal exchange with NVD.

Key Takeaway. *The differences between CVSS scores reported by NVD and various CNAs often follow a bell-shaped distribution. The distribution of differences can be leveraged for filtering outliers and anomaly detection, and for hypothesis tests, e.g., to determine which CNAs report CVSS values that statistically differ from NVD.*

Figure 8 provides further insight into the behavior of CVSS score differences. Figures 8(a) and 8(b) show boxplots of CVSS differences accounting for the top 5 CNAs and ICS CNAs, respectively. It further supports the claims presented above. First, we note that after removing outliers the distribution of score differences is roughly symmetric for most of the CNAs, with PatchStack, VulDB and Rockwell Automation being notable exceptions. Second, we note that score differences are typically centered around positive values, meaning that NVD assigns higher scores compared to CNAs.

4.3 How do ICS products compare against general products?

Figure 7 reports the cumulative distribution function (CDF) of the differences in CVSS scores (CVSS from NVD minus CNA). The blue line in the middle corresponds to all vulnerabilities. The orange line below corresponds to the top 5 CNAs, whereas the green line above corresponds to ICS CNAs. All medians are positive and in agreement with the above discussion. We observe that the median of the difference between CVSS scores, for ICS products, is smaller than that of general products.

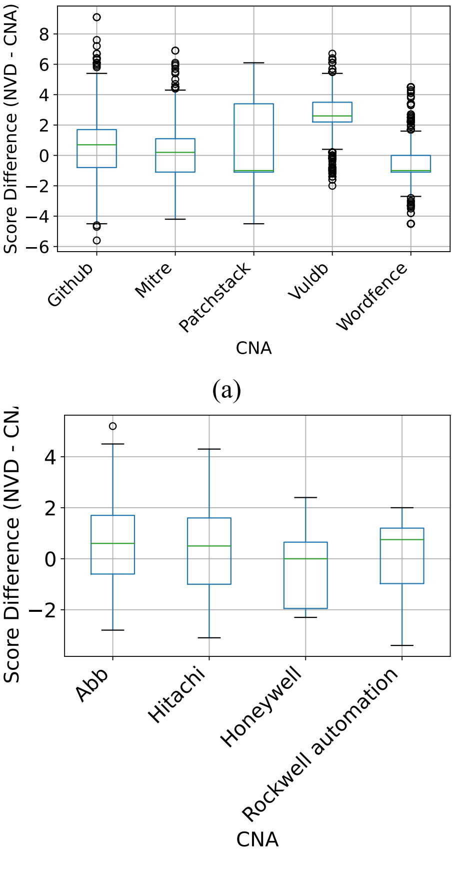


Figure 8. Box plots of CVSS score differences

Note that some vulnerabilities affecting ICS products are exclusive to those products. In the case of Siemens, for instance, out of the 1,846 vulnerabilities appearing in Siemens Security Advisories (SSAs), 1,014 are exclusive to Siemens products. For 1,268 out of the 1,846, there is agreement between Siemens and NVD. Out of those agreements, 620 refer to vulnerabilities affecting only Siemens products. Now focusing on discrepancies, Figure 6(c) accounts for CVEs for which there is discrepancy, and indicates that NVD tends to overrate vulnerabilities affecting exclusively Siemens products, whereas Siemens typically adjusts scores for general vulnerabilities in such a ways that scores increase or decrease by values in the range between -3 and +3.

When we compare the CVSS subscores of ICS against general CNAs, we observe that attack complexity is usually higher and attack vector is more restricted for ICS products (Fig. 2). Rigorous testing and adherence to industry standards may explain this observation, in addition to contextual differences in how the vulnerabilities are embedded into the product.

Key Takeaway. *NVD generally assigns higher CVSS scores than CNAs, with a notable tendency for NVD to overrate vulnerabilities exclusive to ICS products, highlighting domain-specific differences in contextual sensitivity between*

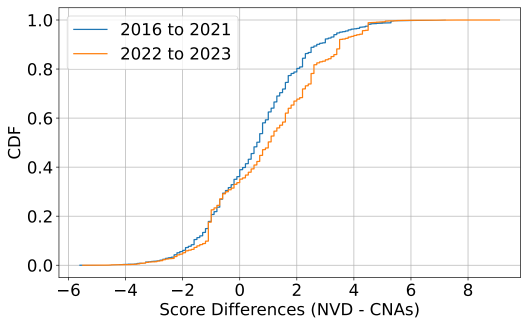


Figure 9. CDF of score differences split by year.

NVD and CNA evaluations.[17]

4.4 How do CVSS score differences evolve?

Figure 9 suggests that over the last two years the difference between NVD and CNAs scores has been higher than over the previous years. The median value remained positive throughout all the considered periods. This indicates that, on average, the NVD consistently assigns higher scores than the CNAs, and this trend has become more prominent in the recent two-year timeframe.

Recall that Table 1 presents the annual distribution of CNAs and their vulnerabilities, along with discrepancies in CVSS vectors between NVD and CNAs. Interestingly, while the absolute number of CVSS vector divergences rose over the years, the fraction of divergences fluctuated, peaking at 0.75. Together with the observations from the previous paragraph, this suggests a varying degree of alignment between NVD and CNA assessments, emphasizing the increasing need for context-specific CVSS scores.

Key Takeaway. *The difference between NVD and CNA CVSS scores has increased, with NVD consistently assigning higher scores; this trend, along with fluctuating divergence rates, highlights the growing need for context-specific CVSS assessments.*

4.5 Accounting for triplets

In this section, we consider vulnerabilities for which we find three CVSS scores. Two of those scores are reported by NVD. The third is obtained from Siemens Security Advisories (SSAs). Table 3 shows the covariance matrix of the scores. As expected, there is a strong positive correlation between all scores. In particular, the correlation is stronger between Siemens scores and other CNA scores, followed by the correlation between Siemens and NVD. The fact that there is stronger correlation between Siemens and other CNAs may suggest that when CNAs report alternative scores, those scores usually tend to deviate from NVD in a similar fashion. Indeed, the CVSS deviation is similar across devices whenever a given component is similarly embedded into those different devices.

Table 3. Covariance matrix regarding base CVSS scores

	Siemens	NVD	other CNA
Siemens	5.027852	3.382297	4.406357
NVD	3.382297	3.051727	3.084234
other CNA	4.406357	3.084234	5.305958

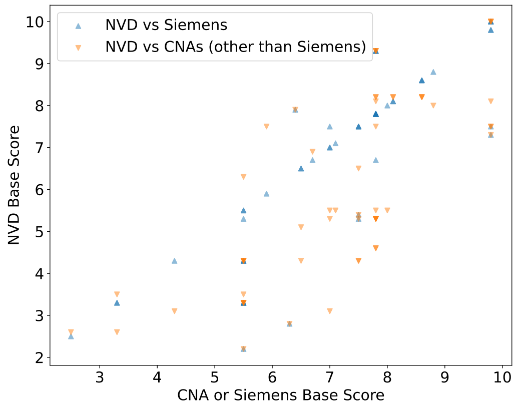


Figure 10. Scatter plot of CVSS scores for NVD against CNAs

Then, Figure 10 shows a scatter plot of the vulnerabilities for which we found three CVSS scores. Each data point represents a pair of CVSS scores pertaining to the same vulnerability. Pairs (CVSS Siemens, CVSS NVD) are depicted in blue, and (CVSS CNA different from Siemens, CVSS NVD) are shown in orange, noting that we account for all CNAs. The line $y = x$, representing points where the CVSS base score from NVD matches the score from the CNA, contains a significant number of blue points. This indicates that, for numerous vulnerabilities, Siemens scores align with those from NVD. When there is a disparity between CNAs and NVD, it is often observed that the CNA scores tend to be smaller than the corresponding NVD scores.

Key Takeaway. *The covariance matrix and scatter plot analysis reveal consistent patterns across different CNAs when compared to NVD, enabling learning as indicated in the upcoming sections.*

5 Knowledge Transfer using CVSS-BERT

In this section, we indicate how Large Language Models (LLMs) can be leveraged to transfer knowledge across CNAs. We consider the task of inferring context-specific CVSS subscores automatically from the textual description of CVEs. To carry out this task, we chose CVSS-BERT Shahid and Debar [2021], a fine-tuned version of BERT for CVSS assessment. By providing the vulnerability description as input, and producing CVSS subscores as output, the LLM automates the process currently performed by experts, while also providing insights on keywords that are relevant for such classification. Training is conducted separately on each of the eight subscores of the CVSS vector (see Figure 2).

Our goals are to 1) show the feasibility of learning context-specific CVSS scores in an automated fashion; 2) indicate the extent at which general data from NVD helps in the scoring of context-specific CVSS scores and 3) quantify the loss of accuracy when predicting heterogeneous CVSS scores from

[17] This discrepancy can be partially explained by the NVD's lack of context and the use of different assessment methodologies by expert groups Wunder *et al.* [2023]; Human Factors in Security and Privacy Group [2024].

Table 4. Comparison of results obtained with CNAs and NVD: classification performance when using only CNAs; only NVD

	accuracy	**precision**	**recall**	**F1-score**
Attack Vector	0.82; 0.85	0.81; 0.84	0.82; 0.85	0.80; 0.83
Attack Complexity	0.80; 0.96	0.72; 0.95	0.80; 0.96	0.72; 0.95
Privileges Required	0.68; 0.72	0.68; 0.72	0.68; 0.72	0.68; 0.71
User Interaction	0.78; 0.88	0.79; 0.89	0.78; 0.88	0.78; 0.88
Scope	0.77; 0.93	0.75; 0.92	0.77; 0.93	0.76; 0.92
Confidentiality Impact	0.70; 0.79	0.71; 0.80	0.70; 0.79	0.69; 0.79
Integrity Impact	0.70; 0.83	0.70; 0.83	0.70; 0.83	0.69; 0.83
Availability Impact	0.71; 0.86	0.69; 0.84	0.71; 0.86	0.69; 0.84
Average	0.74; 0.86	0.73; 0.86	0.74; 0.86	0.73; 0.85

multiple CNAs as opposed to CVSS scores issued by a central authority such as NVD.

Dataset. Our CVSS-BERT analysis leverages a dataset comprising 34,557 CVEs and 69,114 CVSS scores – all vulnerabilities published by the NVD with an additional CVSS score provided by a CNA (see Section 3). Each CVE includes a description, CVSS vector provided by the NVD, and CVSS vector provided by a CNA. The input to the BERT model is the vulnerability description and the output is a CVSS subscore. For each of our analysis, we split the dataset, with 50% of the data for training and 50% for testing as in Shahid and Debar [2021].

To account for context-specific CVSS subscores, we inserted at the beginning of the description string of each vulnerability the name of the CNA that published that CVSS followed by the string 'cna'. One of our intentions was to assess how much the CNA influenced CVSS subscores, as further detailed next.

For each of the considered settings, we report the accuracy, precision, recall and F1-score for the prediction of each CVSS subscore. Additionally, the top tokens that had the most influence on the decisions of each metric value are shown in Table 9.

5.1 How NVD and context-specific inferences behave in an isolated fashion?

We analyze the inference of CVSS subscores separately using data from CNAs and NVD. Table 4 shows the classification performance of models trained and tested using context-specific CVSS vectors from CNAs and NVD. We use a total of 14,729 CVSS vectors for training, and the remaining vectors for testing. The results show the accuracy, precision, recall, and F1-score for each subscore, comparing the performance when using only CNAs versus only NVD.

Upon analyzing the models in Table 4, we observe a significant 10% difference in average accuracy across subscores when comparing NVD against CNAs. Although the amount of data is the same, the key distinction is that one model used vectors exclusively from CNAs, while the other used vectors only from the NVD. This accuracy difference is primarily due to the diversity of CNAs (over 200), each employing its own approach to classify vector subscores. This variability makes it more difficult for the model to identify consistent patterns, thus reducing its overall efficiency. Furthermore, CNAs often have deeper insights into the software and its architecture, utilizing non-public information that is not reflected in the vulnerability description. For the NVD, in contrast, the subscore classification follows a consistent pattern across all vectors. This uniformity makes it easier for the model to classify vectors and generate accurate inferences.

Table 5. NVD and CNAs: differing in at least one subscore

	accuracy	**precision**	**recall**	**F1-score**
Attack Vector	0.90	0.90	0.90	0.89
Attack Complexity	0.88	0.87	0.88	0.87
Privileges Required	0.75	0.75	0.75	0.75
User Interaction	0.88	0.88	0.88	0.87
Scope	0.87	0.87	0.87	0.86
Confidentiality Impact	0.80	0.80	0.80	0.80
Integrity Impact	0.80	0.81	0.80	0.80
Availability Impact	0.79	0.80	0.79	0.79
Average	0.84	0.84	0.84	0.83

Key Takeaway. *NVD assessments are more consistent than CNAs because 1) it counts with a single security team, that 2) uses public information to classify the vulnerabilities. CNAs in contrast are 1) heterogeneous, with certain teams being more conservative than others, and 2) use internal information for their assessment.*

5.2 How NVD and context-specific inferences behave in a fully integrated fashion?

In Table 5, we mixed CVSS vectors from CNAs and NVD. We accounted for CVEs with at least one discrepancy between the subscores provided by the two available CVSS vectors. Since the descriptions are the same for both vectors of a vulnerability, the descriptions are duplicated here, one referring to the NVD vector and the other to the CNA vector, noting that the CNA name was added to the vulnerability description as described in Section 5. Data from the CNA vectors and NVD are mixed in both the training and testing sets. Table 6 is similar to Table 5, except that all data is considered, and not only those entries with differences from the NVD to the CNA. In this case, not only do the descriptions (input) repeat, but also some CVSS vectors (output) corresponding to the same input. Tables 5 and Table 6 are produced using 22,646 and 34,557 CVSS vectors for training, respectively, with an equal number of vectors for testing.

Next, we compare Table 5 against Table 4. Despite the slightly larger dataset for training, the average efficiency experiences a minor decrease compared to Table 4. This occurs because the gains due to additional instances do not compensate the fact that some of the patterns learned from NVD do not apply to the CNAs.

Table 6 presents the results of a broader dataset, encompassing all instances from both NVD and CNAs (34,557 instances). Here, we witness an improvement in efficiency, primarily attributable to the inclusion of vendor data that aligns with the NVD. These instances reinforce existing classification patterns, enriching the model's learning process and leading to higher overall performance.

Key Takeaway. *Mixing CVSS vectors from CNAs and NVD shows that including more data improves efficiency as aligned patterns reinforce learning (see Table 6, to be contrasted against Table 4).*

5.3 How knowledge is transferred from NVD to CNAs and vice versa?

In this section, we consider knowledge transfer across domains, as reported in Tables 7 and 8. In Table 7, CNA vectors

Table 6. Bundle of NVD and CNAs: all instances

	accuracy	precision	recall	F1-score
Attack Vector	0.91	0.91	0.91	0.91
Attack Complexity	0.91	0.89	0.91	0.89
Privileges Required	0.78	0.80	0.78	0.78
User Interaction	0.91	0.91	0.91	0.91
Scope	0.90	0.90	0.90	0.90
Confidentiality Impact	0.82	0.82	0.82	0.82
Integrity Impact	0.83	0.83	0.83	0.83
Availability Impact	0.83	0.83	0.83	0.82
Average	0.87	0.87	0.87	0.86

Table 7. CNAs as train, NVD as test

	accuracy	precision	recall	F1-score
Attack Vector	0.90	0.92	0.90	0.91
Attack Complexity	0.90	0.94	0.90	0.92
Privileges Required	0.74	0.75	0.74	0.74
User Interaction	0.92	0.92	0.92	0.92
Scope	0.87	0.87	0.87	0.87
Confidentiality Impact	0.73	0.76	0.73	0.74
Integrity Impact	0.79	0.81	0.79	0.79
Availability Impact	0.75	0.84	0.75	0.78
Average	0.84	0.85	0.84	0.84

are used for training and NVD vectors for testing. In Table 8, the opposite occurs, with NVD used for training and CNA for testing.

Table 7 indicates promising outcomes by training the model with CNA data and subsequently testing it with NVD data. This observation suggests that training with diverse data and applying it to a specific source like the NVD yields favorable results. Conversely, as illustrated in Table 8, training with a specific source and then testing with a variety of sources led to lower performance.

In summary, the challenge to generalize across different datasets when training on a single source is more stringent, resulting in poorer performance, when compared to the training with data from diverse sources and testing with NVD.

Key Takeaway. *Training models with diverse CNA data and testing with NVD yields better results compared to training with NVD and testing with diverse CNA data, highlighting the importance of varied training data for generalization across different datasets.*

Table 8. NVD as train, CNAs as test

	accuracy	precision	recall	F1-score
Attack Vector	0.89	0.89	0.89	0.88
Attack Complexity	0.84	0.82	0.84	0.78
Privileges Required	0.68	0.70	0.68	0.67
User Interaction	0.86	0.86	0.86	0.86
Scope	0.80	0.79	0.80	0.79
Confidentiality Impact	0.66	0.67	0.66	0.65
Integrity Impact	0.73	0.76	0.73	0.72
Availability Impact	0.66	0.70	0.66	0.58
Average	0.76	0.76	0.76	0.73

5.4 Language and tokens

Table 9 displays some of the top tokens associated with each subscore category in the context of vulnerability analysis. Such tokens are terms that frequently appear within the descriptions or attributes of vulnerabilities, and provide insight into the characteristics of different types of vulnerabilities as seen by CVSS-BERT. Indeed, this table is readily available from CVSS-BERT.

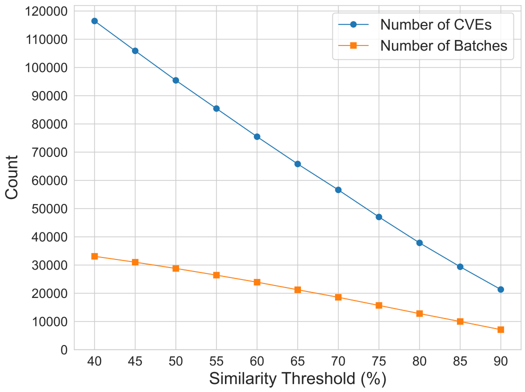


Figure 11. Number of CVEs and batches captured per similarity threshold

Under the category Attack Vector, in addition to the tokens that appear in Table 9 we also observed additional relevant tokens such as ‘site’, ‘remote’, ‘script’, ‘web’, and ‘local’. This suggests that these tokens frequently appear in descriptions related to how an attacker gains access to a system or network. In the Attack Complexity category, tokens like ‘attacker’ and ‘user’ are prominent. These tokens indicate aspects related to the complexity to leverage vulnerabilities for an attack.

Overall, Table 9 provides insight into the language and terminology commonly used to describe different aspects of vulnerabilities, offering valuable information for vulnerability assessment and classification, among other downstream tasks.

All columns in Table 9 correspond to words that appear in the CVE descriptions, as made available at NVD. The two notable exceptions are the first two columns, CNA and NVD, which correspond to tokens that were introduced to build our dataset, contrasting CNAs and by NVD. Recall from Section 5 that an instance in the considered learning problem comprises as its input one of the two keywords, CNA or NVD, followed by the description of the vulnerability as provided by NVD. The corresponding output is the CVSS subscore, assessed by a CNA or by NVD, respectively.

Knowing who assessed the CVSS is relevant. As indicated by the first column in Table 9, when a CVSS score is issued by a CNA, the knowledge about the source is relevant to infer all subscore values. In addition, when the source is NVD, the knowledge of this piece of information is also relevant for most of the subscores.

Some keywords are more relevant to assess exploitability metrics and other for impact metrics. Table 9 indicates that keywords such as ‘attacker’ are relevant to determine exploitability metrics (first five lines in the table), whereas keywords such as ‘cross’ and ‘arbitrary’ are more relevant for impact metrics (last three lines in the table).

Key Takeaway *Knowing the source of the CVSS assessment (CNA or NVD) is crucial for accurately inferring subscore values, and certain keywords are more relevant for assessing exploitability metrics while others are more pertinent for impact metrics.*

6 Vulnerabilities Disclosed in Batches

Many vulnerabilities are disclosed in batches. In this section, we begin by considering the characterization of batches (Section 6.1 and Figure 11). Then, we analyze the prevalence

Table 9. Relevant tokens to determine each subscore

	CNA	NVD	attacker	versions	file	user	allows	authentic	cross	arbitrary	access
Attack Vector	×		×	×	×	×					
Attack Complexity	×	×	×	×		×					
Privileges Required	×					×	×	×			
User Interaction	×		×		×	×			×		
Scope	×	×	×						×		
Confidentiality	×	×							×	×	
Integrity Impact	×				×				×	×	
Availability Impact	×	×								×	×

of batches (Section 6.2 and Figures 12 and 13). Batches are often associated with coordinated disclosure strategies from CNAs or from the NVD itself. In particular, the distribution of CVE batch sizes (Figure 12) reveals the prevalence of small-to-medium-sized batches, while the growth trend over the years (Figure 13) demonstrates the increasing reliance on batch disclosures. In Section 6.3 we indicate how context-specific factors is captured through batches, and in Section 6.4 we return to the main theme of this work, namely CVSS discrepancies, now accounting for batches.

6.1 Methodology for Identifying Batches of CVEs

To identify batches of related vulnerabilities, we leverage the textual description of each CVE, as published by the NVD. Our approach is based on the hypothesis that context-specific vulnerabilities often share semantically similar descriptions and are disclosed in temporally contiguous CVE IDs.

We proceed by vectorizing the textual descriptions using pre-trained language embeddings and computing the cosine similarity between each CVE and its contiguous neighbor (i.e., the CVE with the next ID). For each similarity threshold τ, we form a batch if the cosine similarity between contiguous CVEs exceeds τ. This process builds batches dynamically as similarity relationships are discovered along the sequence of CVE identifiers.

Figure 11 illustrates the impact of the similarity threshold τ on the number of batches and number of CVEs identified as part of a batch. The x-axis reports the cosine similarity threshold τ, and the y-axis shows the count of batches and the number of CVEs grouped by this method. As expected, lower values of τ result in larger and more numerous batches, while higher thresholds produce fewer and smaller clusters.

In this work, we adopt a threshold of $\tau = 0.55$, motivated by inspection of the largest batches produced at this threshold. Increasing τ beyond 0.55 would eliminate these validated groupings. Alternatively, thresholds below 0.5 tended to group unrelated vulnerabilities, creating noisy and overly broad batches. The threshold $\tau = 0.55$ provided the best balance, identifying validated groupings like the top batches shown in Table 10 while minimizing the inclusion of unrelated CVEs. This value ensures that our batch-level analysis is based on meaningful, contextually related vulnerability clusters.

Notably, the largest group, with 322 CVEs in a single batch disclosed in 2018, corresponds to vulnerabilities related to blockchain design, namely Ethereum smart contract flaws (see Table 10). The additional four case studies discussed in Section 6.4 further validate the practical relevance of the methodology and motivates our choice of τ for subsequent analysis.

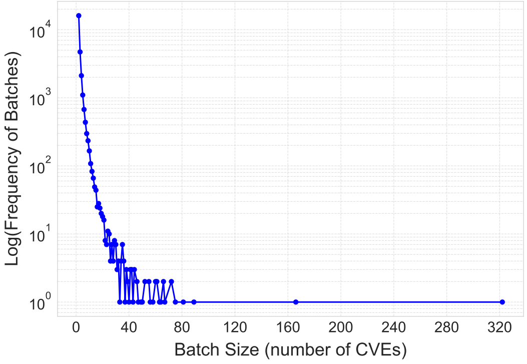


Figure 12. Distribution of CVE batch sizes (log scale)

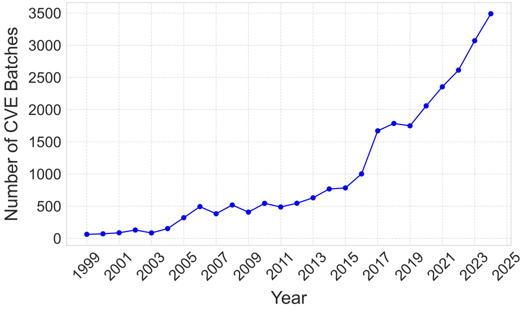


Figure 13. Amount of CVE batches per year

Table 10. Top 5 batches (largest size)

First CVE	Last CVE	Size	About
CVE-2018-13462	CVE-2018-13783	322	Ethereum smart contract flaws
CVE-2023-48442	CVE-2023-48607	166	Adobe XSS vulnerabilities
CVE-2017-16252	CVE-2017-16340	89	Insteon Hub buffer overflows
CVE-2021-1897	CVE-2021-1985	81	Qualcomm component issues
CVE-2021-29545	CVE-2021-29619	75	TensorFlow model bugs

6.2 Prevalence of Batches of CVEs

Figure 12 further illustrates the distribution of batch sizes. Most batches are small, with a long tail of larger batches, as made evident by the log scale on the y-axis. These larger batches often correspond to product families or libraries with a shared vulnerability pattern across multiple components.

The prevalence of batch disclosures has increased steadily over the years. As shown in Figure 13, the number of CVE batches has grown annually, reflecting a growing reliance on batch-based disclosure mechanisms by CNAs and the NVD.

This trend is indicative of systematic and possibly automated practices by certain vendors to report vulnerabilities in groups, often associated with coordinated disclosure campaigns or the release of product advisories.

We also observe that the mean and standard deviation of batch sizes per year (Figure 14) have remained relatively stable over time, with occasional spikes driven by large coordinated disclosures (e.g., the blockchain-related batch in 2018 containing 322 CVEs).

These results confirm that batch disclosures are not rare exceptions, but rather a structural characteristic of modern vulnerability reporting, especially for vendors with large product portfolios or security research organizations disclosing variants of the same flaw across multiple deployments.

It is important to consider the implications of batch size on our analysis. Large batches, such as the Ethereum example (322 CVEs), are often indicative of a systematic flaw affecting an entire product ecosystem or a widespread library. The analysis of score variations within these batches provides strong evidence of how a single underlying issue is contextualized across different implementations.

Smaller batches, while more numerous, often correspond to a handful of related vulnerabilities within a single product or a specific version release. For example, the pair of vulnerabilities CVE-2023-21555 and CVE-2023-21556 form a small batch affecting Windows Layer 2 Tunneling Protocol (L2TP). Although they target the same products, they leverage different weaknesses, captured by distinct Common Weakness Enumeration (CWE) identifiers. It also worth noting that there are explicit references between CVE-2023-21555 and CVE-2023-21556 in the vulnerability descriptions, further reinforcing that they are part of a batch. While the statistical power of such small batches is lower, they offer valuable, highly-specific insights into how vendors assess localized issues, providing a microscopic view of context-driven scoring.

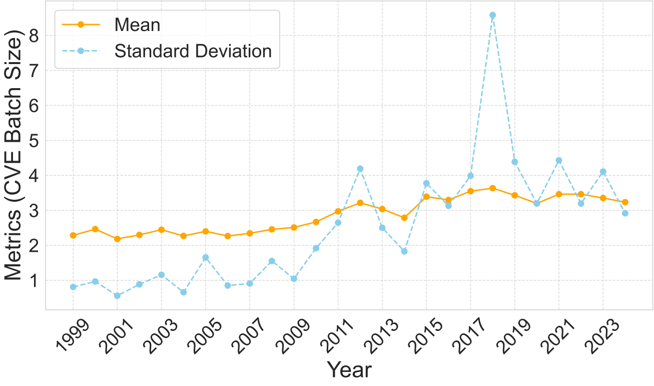


Figure 14. Mean and standard deviation of CVE batch size per year

6.3 Context Captured through Batches

To evaluate the coherence of vulnerability assessments within a batch, we analyze the variation of scores from different rating systems. In addition to CVSS, we consider the Exploit Prediction Scoring System (EPSS). EPSS is a data-driven initiative that provides a probability score (between 0 and 1) representing the likelihood that a vulnerability will be exploited in the wild within the next 30 days. Unlike CVSS, which measures severity, EPSS measures threat. A consistent EPSS score across a batch would suggest that the vulnerabilities are perceived as having a similar exploitation threat level.

Some batches correspond to groups of vulnerabilities related to a given product. These are cases in which CNAs, such as vendors, create multiple related CVEs to represent distinct contextual instantiations of a vulnerability across different components or deployment settings. This granularity simplifies the task of CVSS assignment, as it enables CNAs to encode context directly into the structure of the vulnerability database. The impact of such practice is evident when comparing score variations inside a batch, as shown in Figures 15 and 16. Notably, Figure 15 shows that for 62% of batches, all vulnerabilities receive the same EPSS score, suggesting coherent assessments. However, for the rest of the batches, the fact that multiple related vulnerabilities received different CVE scores allowed for fine tuning of EPSS scores. Similar comments apply to CVSS scores, as illustrated in Figure 15 and further discussed next.

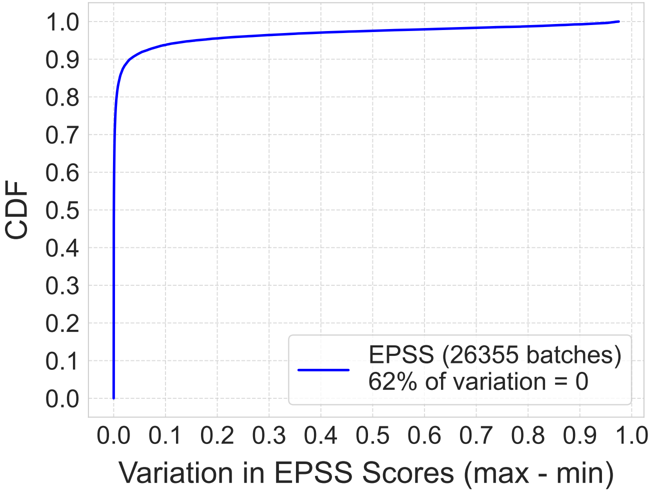


Figure 15. CDF of EPSS score variation in same batch

6.4 Explaining CVSS Discrepancies Through Batch Analysis

When CVSS changes inside a batch, the root cause of discrepancies can sometimes be identified. Unlike the rest of the paper, where CVSS differences between CNAs and NVD were analyzed mostly through statistical divergence, batch-level analysis allows us to investigate fine-grained contextual shifts. In particular, discrepancies in subscores across a batch can often be attributed to changes in the vulnerability description, affected product, or environment assumptions. These variations are reflected in Figure 16 and offer a more explainable lens into CVSS divergence—contrasting with previous sections where the only contextual clue was the CNA identity.

6.4.1 CVSS Discrepancies

To better understand how CVSS scores vary within grouped vulnerability disclosures, we analyzed the variation of base scores across different versions of CVSS for each identified

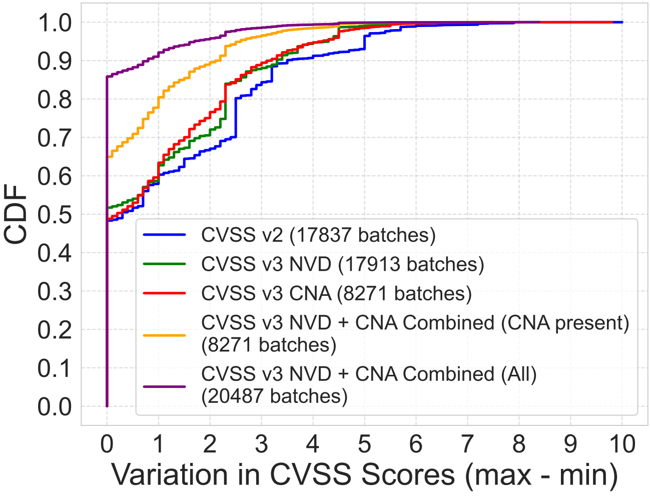


Figure 16. CDF of CVSS score variation in same batch

batch. For this purpose, we computed the difference between the maximum and minimum values of CVSS base scores assigned within each batch. We considered CVSS v2, CVSS v3 as assessed by NVD, and CVSS v3 as reported by CNAs.

The variations were computed as follows. For each batch, we grouped all associated CVEs and evaluated the maximum minus minimum score assigned to them (see Figure 16). This process was applied to the CVSS v2 base score (blue line), CVSS v3 base scores from NVD (green line) and CVSS v3 base scores from CNAs (red line). We also considered combined CVSS v3 score—defined as the maximum variation across the set comprising all NVD and CNA values within the batch. The "combined" variation captures disagreement both between and within sources. To account for cases where CNA scores are not available, we computed two versions of this combined metric: one restricted to batches where CNA scores were present (yellow line), and one including all batches regardless of CNA availability (purple line).

Figure 16 depicts the empirical cumulative distribution functions (CDFs) for each of these score variation types. The horizontal axis corresponds to the score variation (i.e., max – min within each batch), and the vertical axis represents the cumulative fraction of batches with variation less than or equal to the corresponding value. The CDFs allow us to compare the internal consistency of scoring across different CVSS versions and sources.

We observe that CVSS v2 exhibits less variation within batches compared to CVSS v3, possibly due to its coarser scoring granularity. For CVSS v3, discrepancies are more pronounced, especially when CNA assessments are considered. The curve labeled "CVSS v3 Combined (NVD + CNA) (All)" aggregates score differences from both sources and highlights divergence not only across vulnerabilities but also across assessors. In particular, the CDF for batches with both CNA and NVD scores available shows the highest variability, reflecting a broader spectrum of context-sensitive evaluations.

This analysis reinforces the hypothesis that CNAs often introduce additional context into CVSS scoring, which leads to greater variation within batches. Conversely, NVD assessments, which aim for generality and worst-case assumptions, tend to be more homogeneous. These discrepancies, observed at the batch level, provide further evidence of the contextual nature of CVSS assessments and the need for context-specific considerations when interpreting vulnerability severity.

6.4.2 Example 1: Blockchains

The vulnerabilities between CVE-2018-13462 and CVE-2018-13783 (first line in Table 10) form a large batch of 322 CVEs related to a recurring design flaw in Ethereum smart contracts, particularly involving integer overflows in the `mintToken` function. These vulnerabilities allow malicious actors—typically the contract owner—to mint tokens arbitrarily, compromising the integrity of the token supply mechanism. Despite the scale and similarity of these issues, we were unable to find significant variation in the assigned *CVSS v3* base scores: at NVD, all received a score of 7.5 (*High*) or remain unscored. However, the structure of the disclosure facilitates contextual re-evaluation: since each vulnerability has a unique CVE identifier corresponding to a specific smart contract or blockchain deployment, it is possible to fine-tune severity scores to the relevant context without requiring changes to the batch as a whole.

6.4.3 Example 2: Adobe XSS Vulnerabilities

The batch ranging from CVE-2023-48442 to CVE-2023-48607 includes 166 CVEs affecting various Adobe products, primarily related to Cross-Site Scripting (XSS) vulnerabilities. These issues often stem from insufficient input sanitization or improper output encoding in web-accessible components of Adobe applications. While the underlying flaw type—XSS—is consistent across the batch, the context in which each vulnerability appears may significantly affect its exploitability and impact (e.g., within a browser plugin versus a document viewer).

Similar to the blockchain case, the CVEs in this batch received uniform CVSS v3 scores of 5.4 from the NVD, with a notable exception: CVE-2023-48599, which was assigned a lower CVSS v3 score of 4.3 by the CNA (Adobe), indicating that the vendor accounted for a subtle contextual difference in the scoring process. Specifically, for CVE-2023-48599 the attacker would need to convince the user to open a specially crafted file or visit a particular website, which introduces additional complexity and reduces the likelihood of successful exploitation. In contrast, other vulnerabilities in the batch may be exploitable through more straightforward means.

6.4.4 Example 3: Insteon Hub Buffer Overflows

Between CVE-2017-16252 and CVE-2017-16340, we identify a batch of 89 CVEs linked to buffer overflow vulnerabilities in the Insteon Hub, a consumer IoT device. These vulnerabilities were discovered as part of a systematic reverse engineering effort targeting firmware images of smart home controllers. Despite being similar in nature—often involving stack- or heap-based overflows in network-exposed services—the potential for exploitability varies based on whether the vulnerable component is accessible over the local network, through cloud relay, or only via physical access. To reflect these differences, CVSS v3 scores vary from 8.1 (*High*) to 9.9 (*Critical*), reflecting distinctions in attack surface and user impact.

6.4.5 Example 4: Qualcomm Component Issues

The batch from CVE-2021-1897 to CVE-2021-1985 consists of 81 CVEs related to various components in Qualcomm chipsets. These vulnerabilities span different modules, from memory management to data services. The CVSS scores within this batch show significant variation, as the impact of a flaw in a low-level firmware component can differ greatly from one in a higher-level driver, even if they share a similar root cause. This batch exemplifies how a vendor (Qualcomm, in this case) uses distinct CVEs to capture the specific context of a single vulnerability type across a complex hardware and software stack.

6.4.6 Example 5: TensorFlow Model Bugs

The batch between CVE-2021-29545 and CVE-2021-29619 includes 75 CVEs affecting Google's TensorFlow, a popular machine learning library. The vulnerabilities are often related to how the library handles malformed model files, leading to issues like denial-of-service or memory corruption. The CVSS scores assigned by the CNA (Google) are often lower than NVD's generic assessment. This is because Google can account for mitigating factors, such as the requirement for an attacker to first trick a user into loading a malicious machine learning model, which increases the attack complexity and lowers the overall severity in many practical scenarios.

7 Related Work

In this section, we present related work pertaining to the comparison of vulnerability severity from various sources and, more broadly, to the assessment of software vulnerabilities.

7.1 CVSS assessment

The consistency and accuracy of vulnerability assessments, particularly focusing on CVSS, have been subject of several previous studies. In Massacci [2024]; Allodi *et al.* [2020] the authors discuss an experiment comparing the accuracy of vulnerability assessments by students with varying technical education levels and security professionals, using CVSS. It highlights the feasibility of measuring the effects of expertise on the accuracy of vulnerability assessments, while suggesting that experts and students converge to similar outcomes. In contrast, more recent work Wunder *et al.* [2023]; Human Factors in Security and Privacy Group [2024] investigates the consistency of CVSS evaluations among a group of users through surveys. It reveals inconsistencies in the evaluation of specific CVSS metrics for widespread vulnerability types and discusses reasons for these inconsistencies.

We found that CNAs provide consistent CVSS subscores for most CVEs (see, for instance, Figure 2) and disagreements typically arise due to context-related elements. Additionally, leveraging machine learning techniques, such as BERT, for the task of predicting subscores from CVE textual descriptions, one can achieve prediction accuracies above 70% (see Section 5). This suggests that there are consistent patterns to be learned from the current assessments, and that learning can be transferred across NVD and CNAs.

7.2 Comparing different sources

Previous work identified inconsistencies at security-related databases, such as NVD Croft *et al.* [2022]; Zhang *et al.* [2023a]; Anwar *et al.* [2021]; Dong *et al.* [2019]; Croft *et al.* [2023]. These works are complementary to ours. Whereas those related works focus on unintentional inconsistencies, our focus is on CVSS scores that are explicitly and intentionally published as alternative assessments for the same vulnerability. In particular, the variations in CVSS scores in our dataset occur due to context-specific changes across CNAs, leading to diverse CVSS assessments.

In Croft *et al.* [2022] the authors investigate severity scores issued by three sources: bug reports (Bugzilla), advisories (Mozilla Security Advisory) and NVD. The authors observed that the severity rankings and subsequent vulnerability prioritization schemes are different across the considered sources. According to the authors, this finding indicates the complexity of assessing the severity of vulnerabilities and threatens the accuracy of prioritization schemes inferred from those sources. The authors also reported considerable differences in the severity rankings between the Mozilla Advisory and NVD, despite both of these sources receiving expert security analysis. While leveraging statistical models to predict scores, the authors noted that the NVD data source produced the best predictive performance models. This high performance is likely credited to the targeted analysis and standardized reporting of the NVD; both the descriptions and severity rankings follow consistent guidelines and are reviewed by security experts. Bugzilla data performed worse than NVD, possibly due to a lack of internal consistency, as descriptions and severity rankings were often written by various reporters.

In this work, we focused on information provided by NVD, including the NVD assessment of vulnerabilities CVSS scores and CNAs assessment as reported through NVD. Therefore, we were able to contemplate more sources than Croft *et al.* [2022], providing a birds-eye-view of the differences between vulnerability CVSS scores. In future work, we plan to also investigate additional sources, e.g., by curating security advisories and bug reports from different CNAs.

7.3 Accounting for environment in the CVSS score

In Maidl *et al.* [2021, 2019] the authors indicate how to account for environmental aspects for the assessment of severity scores. In particular, the authors propose an algorithm that aims to calculate the system-specific severity of a vulnerability based on a system model and a risk profile. The latter takes into account particular aspects of deployments, including system-specific exposure and the system's purpose, especially in critical infrastructure scenarios, for impact assessment. This approach allows for a more nuanced and targeted response to vulnerabilities, addressing the challenges of patching in complex operational environments. Our work extends this prior effort, leveraging public data to show how CVSS may change depending on the context.

7.4 Risk aggregation

The idea of leveraging different sources and dimensions to produce a final score falls in the realm of risk aggregation Bjørnsen and Aven [2019]. In particular, we envision that risk can be aggregated to accommodate contextual aspects, including the influence of different vulnerabilities that are related to each other, and the impact of different components of a product that together comprise the whole system. In this work, we showed how different CNAs may differ in their vulnerability risk assessments. In future work, we envision that the various risk assessments may be taken as prior knowledge to be tuned based on the context to produce more meaningful assessments of risk, a posteriori.

7.5 Explaining and predicting CVSS scores

There is a growing literature on explaining and predicting CVSS scores, e.g., based on the textual description of vulnerabilities Elbaz *et al*. [2020]; Costa *et al*. [2022]; Kühn *et al*. [2023]; Shahid and Debar [2021]; Costa *et al*. [2022]; Han *et al*. [2017]; Khazaei *et al*. [2016]. However, none of those prior works accounted for different CNA assessments while producing their predictions. We indicate that different CNAs may require different models for CVSS assessment. We envision that the models presented in Elbaz *et al*. [2020]; Shahid and Debar [2021] can be tuned on a CNA-basis, and leave that as subject for future work.

8 Conclusion

The assessment of software vulnerability severity calls for the evaluation of context-specific attributes. In this paper, we report a tendency at NVD, wherein CNAs increasingly tailor their assessment to specific products linked with vulnerabilities. This raises a crucial question: the ongoing discussion about maintaining consolidated assessments for each vulnerability, subdividing vulnerabilities into additional CVEs to accommodate intricate context-specific characteristics, or providing multiple context-specific CVSS ratings for a single vulnerability. Regardless of the direction this discussion takes, the takeaway is the pressing need for understandable assessments that supports users in taking decision regarding mitigative actions, accounting for diverse CNA perspectives towards CVSS.

Indeed, ownership of CVSS assessment should increasingly rely more on CNAs as opposed to a central authority. CNAs know more details about relevant libraries and codes, and about how integration and interactions occur.

In summary, depending on how a component is embedded inside a product, its CVSS score, reflecting the CNA assessment, may need to be adjusted. Fostering open and public discussion around these assessments can yield more fruitful outcomes. Importantly, empowering more CNAs to report CVSS scores, accompanied by justifications for their scores, affords users greater freedom and, ultimately, more influence in deciding which CNAs to follow.

This work opens up many avenues for future research. From the data collection standpoint, we envision the continuous crawling of security advisories from additional vendors to augment our dataset. From the methodological standpoint, tools such as VIET Zhang *et al.* [2023b] can be used to improve the transfer learning between sources. Finally, from an analytical standpoint, we envision that this work is a first step towards a more fundamental understanding about how to leverage multiple perspectives towards vulnerabilities.

Acknowledgements

This project was sponsored by CAPES, CNPq, and FAPERJ (315110/2020-1, E-26/211.144/2019, and E-26/201.376/2021).

Authors' Contributions

AR, LGM, LSC and DSM contributed to the conception of this study. LGM and LSC performed the experiments and evaluations. LGM, LSC, DSM, EL, GKS, AK and TL are this manuscript's main contributors and writers. All authors read and approved the final manuscript.

Competing interests

The authors declare that they do not have competing interests.

Availability of data and materials

All the data used in this work is available through NVD.

References

Allodi, L., Cremonini, M., Massacci, F., *et al*. (2020). Measuring the accuracy of software vulnerability assessments: experiments with students and professionals. *Empirical Softw. Engin.*, 25:1063–1094. DOI: 10.1007/s10664-019-09797-4.

Allodi, L. and Massacci, F. (2014). Comparing vulnerability severity and exploits using case-control studies. *ACM TISSEC*, 17(1):1–20. DOI: 10.1145/2630069.

Anwar, A. *et al*. (2021). Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses. *IEEE Transactions on Dependable and Secure Computing*, 19(6):4255–4269. DOI: 10.1109/dsn-s52858.2021.00011.

Bjørnsen, K. and Aven, T. (2019). Risk aggregation: What does it really mean? *Reliability Engineering & System Safety*, 191:106524. DOI: 10.1016/j.ress.2019.106524.

Costa, J. C., Roxo, T., Sequeiros, J. B., Proenca, H., and Inacio, P. R. (2022). Predicting CVSS metric via description interpretation. *IEEE Access*, 10:59125–59134. DOI: 10.1109/access.2022.3179692.

Coutinho, L. S., Menasche, D., Miranda, L., Lovat, E., Kumar, S. G., Ramchandran, A., Kocheturov, A., and Limmer, T. (2024). How context impacts vulnerability severity: An analysis of product-specific cvss scores. In *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, pages 17–27. DOI: 10.1145/3697090.3697109.

Croft, R., Babar, M. A., and Kholoosi, M. M. (2023). Data quality for software vulnerability datasets. In *ICSE*, pages 121–133. IEEE. DOI: 10.1109/icse48619.2023.00022.

Croft, R., Babar, M. A., and Li, L. (2022). An investigation into inconsistency of software vulnerability severity across data sources. In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 338–348. IEEE. DOI: 10.1109/saner53432.2022.00050.

Dong, Y., Guo, W., Chen, Y., *et al.* (2019). Towards the detection of inconsistencies in public security vulnerability reports. In *USENIX Security*, pages 869–885. Available at:`https://dl.acm.org/doi/10.5555/3361338.3361399`.

Elbaz, C. *et al.* (2020). Fighting n-day vulnerabilities with automated cvss vector prediction at disclosure. In *Int. Conf. Availability, Reliability and Security*. DOI: 10.1145/3407023.3407038.

FIRST (2024). Available at:`https://www.first.org/cvss/v3.1/specification-document`.

HackRead (2024). NIST NVD Halt Leaves Vulnerabilities Untagged. Available at: `https://www.hackread.com/nist-nvd-halt-leaves-vulnerabilities-untagged/`.

Han, Z., Li, X., Xing, Z., *et al.* (2017). Learning to predict severity of software vulnerability using only vulnerability description. In *ICSME*, page 125. DOI: 10.1109/icsme.2017.52.

Human Factors in Security and Privacy Group (2024). Consistency of CVSS. Available at:`https://www.cs1.tf.fau.de/research/human-factors-in-security-and-privacy-group/consistency-of-cvss/`.

Khazaei, A. *et al.* (2016). An automatic method for CVSS score prediction using vulnerabilities description. *Journal of Intelligent & Fuzzy Systems*, 30(1). DOI: 10.3233/ifs-151733.

Kühn, P., Relke, D. N., and Reuter, C. (2023). Common vulnerability scoring system prediction based on open source intelligence information sources. *Computers & Security*, 131:103286. DOI: 10.1016/j.cose.2023.103286.

Le, T. H. M. and Babar, M. A. (2022). On the use of fine-grained vulnerable code statements for software vulnerability assessment models. In *Intl. Conference on Mining Software Repositories*, pages 621–633. DOI: 10.1145/3524842.3528433.

Maidl, M., Kröselberg, D., Zhao, T., and Limmer, T. (2021). System-specific risk rating of software vulnerabilities in industrial automation & control systems. In *2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 327–332. IEEE. DOI: 10.1109/ISSREW53611.2021.00097.

Maidl, M., Wirtz, R., Zhao, T., Heisel, M., and Wagner, M. (2019). Pattern-based modeling of cyber-physical systems for analyzing security. In *Proceedings of the 24th European Conference on Pattern Languages of Programs*, pages 1–10. DOI: 10.1145/3361149.3361172.

Massacci, F. (2024). The holy grail of vulnerability predictions. *IEEE S&P*, 22(1):4. DOI: 10.1109/msec.2023.3333936.

Mead, N. R. and Stehney, T. (2005). Security quality requirements engineering (square) methodology. *ACM SIGSOFT Software Engineering Notes*, 30(4):1–7. DOI: 10.21236/ada443493.

Shahid, M. R. and Debar, H. (2021). CVSS-BERT: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In *ICMLA*, pages 1600–1607. IEEE. DOI: 10.48550/arXiv.2111.08510.

Wunder, J., Kurtz, A., Eichenmüller, C., Gassmann, F., and Benenson, Z. (2023). Shedding Light on CVSS Scoring Inconsistencies: A User-Centric Study on Evaluating Widespread Security Vulnerabilities. In *IEEE Security and Privacy*, page 58. DOI: 10.1109/sp54263.2024.00058.

Zhang, S., Cai, M., Zhang, M., Zhao, L., *et al.* (2023a). The Flaw Within: Identifying CVSS Score Discrepancies in the NVD. In *CloudCom*, pages 185–192. IEEE. DOI: 10.1109/cloudcom59040.2023.00039.

Zhang, S., Zhang, M., and Zhao, L. (2023b). Viet: A tool for extracting essential information from vulnerability descriptions for cvss evaluation. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 386–403. Springer. DOI: 10.1007/978-3-031-37586-$6_2$3.