Applied Computing and Informatics 15 (2019) 129–135

Contents lists available at ScienceDirect

Applied Computing and Informatics

journal homepage: www.sciencedirect.com

Original Article

# A look at the time delays in CVSS vulnerability scoring

Jukka Ruohonen

*Department of Future Technologies, University of Turku, FI-20014 Turun yliopisto, Finland*

ARTICLE INFO

*Article history:*
Received 5 October 2017
Revised 23 November 2017
Accepted 21 December 2017
Available online 27 December 2017

*Keywords:*
Software vulnerability
Vulnerability severity
Severity scoring
Database maintenance
Cyber security

ABSTRACT

This empirical paper examines the time delays that occur between the publication of Common Vulnerabilities and Exposures (CVEs) in the National Vulnerability Database (NVD) and the Common Vulnerability Scoring System (CVSS) information attached to published CVEs. According to the empirical results based on regularized regression analysis of over eighty thousand archived vulnerabilities, (i) the CVSS content does not statistically influence the time delays, which, however, (ii) are strongly affected by a decreasing annual trend. In addition to these results, the paper contributes to the empirical research tradition of software vulnerabilities by a couple of insights on misuses of statistical methodology.


## 1. Introduction

Software vulnerabilities are software bugs that expose weaknesses in software systems. The CVSS standard is used to classify the severity of known and disclosed vulnerabilities. Once the classification and evaluation work has been completed for a vulnerability identified with a CVE, the structured and quantified severity information is stored to vulnerability databases. Motivated by a recent empirical evaluation [16], this paper examines the time delays between the publication of CVEs and the usually later publication of CVSS information. The scope is restricted to NVD and the second revision of the CVSS standard.

The use of CVSS is mandated and recommended by many state agencies for assessments in different security-critical domains [36], including but not limited to medical devices [38] and the payment card industry [2]. The standard has been also incorporated into different governmental security risk, threat, and intelligence systems. Furthermore, CVSS information is used in numerous different commercial products [16], ranging from vulnerability scanners and compliance assessment tools to automated penetration testing and intrusion detection systems.

CVSS is also widely used in academic research. Typical application domains include risk analysis [2,14], security audit frameworks [4], so-called attack graphs [7,26], and empirical assessments using CVSS for different purposes [1,25,31,33]. To these ends, a lot of work has been done to improve CVSS with different weighting algorithms [17,40], among other techniques [9,30]. With some rare exceptions [13], limited attention has been given for examining how severity assessments are done in practice.

Practical approaches are important because CVSS has faced also challenges. Analogous to problems that have affected CVE assignments [33,34], different practical problems have influenced the severity assignments for CVE-stamped vulnerabilities. Excluding the actual content of the standard, the historical problems related to classification inconsistencies, time delays, and the proliferation of classification standards [5,24]. Some of these problems have continued to exist. For instance, proliferation has continued in recent years; new standards have been introduced for classifying software misuse and configuration vulnerabilities [3]. Some countries [45] and companies [43] have also introduced their own severity metrics. To examine whether also the problem with time delays is still present—as has been suspected [18], a brief remark is required about the CVE and CVSS publication processes in the context of NVD. Although the available documentation about these processes is limited [28], the sketch presented in Fig. 1 is not a far-fetched analytical speculation.

The process starts when security researchers, vendors, and other related actors request CVEs for vulnerabilities they have discovered or made aware of. These request-response dynamics are handled by the non-profit MITRE corporation. As is common in

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

*E-mail address:* juanruo@utu.fi

https://doi.org/10.1016/j.aci.2017.12.002
2210-8327/

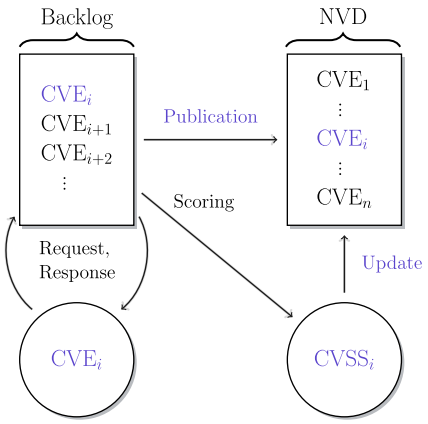


**Fig. 1.** A simplified model for CVSS processing.

software engineering, MITRE presumably maintains a backlog for the CVEs assigned, some of which may be even rejected for inclusion to NVD. Although the structure of the backlog is unknown, a simple FIFO (first-in, first-out) might be considered in order to connect the speculation to a recent theoretical work [10]. In any case, eventually the vulnerabilities accepted for archiving are published in NVD. In parallel to the coordination and archiving work related to CVEs, vulnerabilities are evaluated for their severity by the NVD team, which largely operates independently from others carrying similar evaluations [16]. Once the evaluation has been completed, the CVE-referenced vulnerability information is updated in NVD. The time lags between the initial CVE publications and the later CVSS updates constitute the empirical phenomenon examined.

There is another viewpoint to the abstract CVE backlog. This viewpoint originates from the so-called switching costs, which are often high for information technology standards [37]. Such theoretical costs cover also database maintenance: even small changes made to standards may imply a lot of evaluation work particularly in case old information needs to be updated. This concern was raised also during the 2007 introduction of the second revision of the CVSS standard [36]. In other words, updates can be costly in terms of time and resources—given the nearly ninety thousand vulnerabilities currently archived in NVD. Therefore, it is relevant to ask the following research question (RQ) about the time lags affecting CVSS scoring.

| | |
|---|---|
| RQ$_1$ | *Do the time delays between CVE publications and CVSS updates vary systematically according to an annual year-to-year trend?* |

Another question relates to the content of the CVSS standard in terms of the vulnerabilities scored. Reflecting the disagreements among experts about the severity of some vulnerability types [13], it can be hypothesized that the CVSS content itself affects the time delays. Not all vulnerabilities are equally easy (or hard) to classify in terms of severity; hence, some vulnerabilities may take a relatively short (long) time to classify. This reasoning is presented as a second research question, stated as follows.

| | |
|---|---|
| RQ$_2$ | *Do the time delays vary systematically according to the content of the CVSS severity information?* |

Finally, a third and final question can be postulated for controlling the answers to the earlier two questions:

| | |
|---|---|
| RQ$_3$ | *Does the answer to RQ$_2$ hold when also the annual trend is controlled for?* |

According to the empirical results, only the answer to RQ$_1$ is positive. For predicting the time delays, the CVSS content is largely noise. The statistical effect (RQ$_2$) also fades away once the annual trend is controlled for (RQ$_3$). To elaborate how these conclusions are reached, the remainder of this paper is structured into three sections. Namely: Section 2 introduces the dataset and the operationalization of the variables used, Section 3 outlines the statistical methodology and presents the empirical results along the way, and Section 4 finally discusses the findings.

## 2. Setup

To outline the setup for the analysis, the following discussion will address the operationalization of the delay metric examined the covariates used to model the metric.

### 2.1. Response

Following the so-called vulnerability life cycle research tradition [25,33], the interest relates to a time difference

$$\Delta_i = \tau_{\text{CVSS}_i} - \tau_{\text{CVE}_i^a}, \quad \text{given} \qquad (1)$$
$$\tau_{\text{CVSS}_i} \geqslant \tau_{\text{CVE}_i^a} \quad \text{for all } i = 1, \ldots, n.$$

The integer $\tau_{\text{CVSS}_i}$ denotes the day (timestamp) at which a CVSS entry was generated for the $i$:th CVE that was published at $\tau_{\text{CVE}_i^a}$. In practice, the two timestamps map to the fields `cvss:generated-on-datetime` and `vuln:published-datetime` in the NVD's extensible markup language schema. Although the exact meaning of the fields is undocumented, the time differences can be interpreted as delays between CVE and CVSS publications.

Of the 89,465 archived vulnerabilities with both CVE and CVSS entries, the condition $\tau_{\text{CVSS}_i} \geqslant \tau_{\text{CVE}_i^a}$ fails to satisfy only for 1,375 vulnerabilities. Without loss of generality, these cases were excluded. The same applies to CVEs without severity records. At the time of retrieving the NVD content [27], there were 2,218 vulnerabilities that were published but still lacked CVSS records. Most of these cases relate either to new vulnerabilities that are still in the pipeline for severity assessments, or to already published CVEs that were later rejected as inappropriate for archiving. Either way, these had to be also excluded in order for $\Delta_i$ to be defined for all cases observed. In total, the dataset examined contains $n = 89,465 - 1,375 = 88,090$ archived cases. Given these cases, the distribution of the time delays observed is shown in Fig. 2.

The timelines exhibit a heavy-tailed distribution with extremely long right tail. A half of the vulnerabilities observed have seen

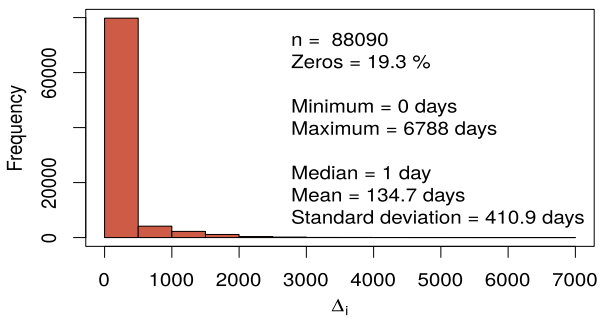


**Fig. 2.** CVE-CVSS publication time delays (Eq. 1).

severity assignments already a day after CVEs were published, but the standard deviation is still over a year. Most of this deviation is caused by a few extreme outliers for which the severity scores were assigned even a decade after the CVEs were originally published.

To briefly probe these outliers further, Fig. 3. displays the distribution of another time difference

$$\delta_i = \tau_{\text{CVSS}_i} - \tau_{\text{CVE}_i^b}, \tag{2}$$

where $\tau_{\text{CVE}_i^b}$ denotes the `vuln:last-modified-datetime` field in NVD. The large amount of negative values indicate that CVEs are often updated after these were already published with CVSS information. Interestingly, 187 outlying cases satisfy $\delta_i > 0$, which may point toward some inconsistencies in database maintenance; CVSS information was generated without updating the corresponding $\tau_{\text{CVE}_i^b}$ timestamps. About a quarter of the cases observed satisfy $\delta_i = 0$, meaning that the latest CVE modifications matched the generation of severity information.

### 2.2. Covariates

Two types of covariates are used for modeling the time delays in (1). The first contains the CVSS information itself. The CVSS (v. 2) standard [6] classifies the impact of vulnerabilities according to confidentiality, integrity, and availability (CIA). Each letter in the CIA acronym further expands into three categories that characterize the impact upon successfully exploiting the vulnerability in question. Thus, the analytical structure behind the impact dimension can be illustrated with a diagram:

$$\text{IMPACT} \in \begin{cases} \text{CONFIDENTIALITY} \in \begin{cases} \text{NONE}^* \\ \text{PARTIAL} \\ \text{COMPLETE} \end{cases} \\ \text{INTEGRITY} \in \begin{cases} \text{NONE}^* \\ \text{PARTIAL} \\ \text{COMPLETE} \end{cases} \\ \text{AVAILABILITY} \in \begin{cases} \text{NONE}^* \\ \text{PARTIAL} \\ \text{COMPLETE} \end{cases} \end{cases}$$

The three impact metrics measure the severity of a vulnerability on a system after the vulnerability has already been exploited. However, not all vulnerabilities can be exploited; therefore, the CVSS standard specifies also an exploitability dimension for vulnerabilities. Like with the impact dimension, exploitability expands into three metrics (access vector, complexity, and authentication) that can each take three distinct values. The analytical meaning can be again summarized with the following diagram:

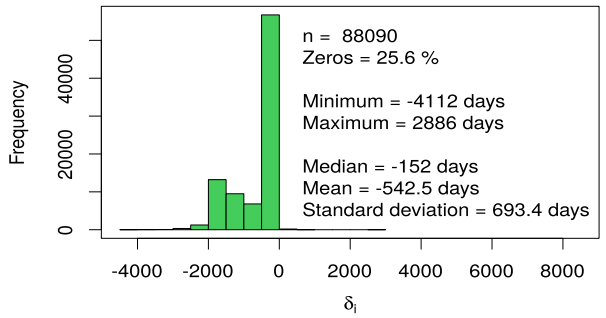


**Fig. 3.** CVE-CVSS modification time delays (Eq. 2).

$$\text{EXPLOITABILITY} \in \begin{cases} \text{VECTOR} \in \begin{cases} \text{LOCAL}^* \\ \text{NETWORK} \\ \text{ADJACENT} \end{cases} \\ \text{COMPLEXITY} \in \begin{cases} \text{LOW}^* \\ \text{MEDIUM} \\ \text{HIGH} \end{cases} \\ \text{AUTHENTICATION} \in \begin{cases} \text{NONE}^* \\ \text{SINGLE} \\ \text{MULTIPLE} \end{cases} \end{cases}$$

The rationale for the impact and exploitability metrics relate to different combinatory relationships between the different values the metrics can take. For instance, it is probable that mass-scale attacking tools target less complex vulnerabilities that can be exploited through a network without performing authentication, possibly regardless of the impact upon confidentiality, integrity, and availability. There exists also some empirical evidence along these lines [1]. However, the impact and exploitability dimensions both relate to intrinsic characteristics of vulnerabilities; they are constant across time and environments. For instance, EXPLOITABILITY cannot answer to a temporal question about whether an exploit is known to exists for the vulnerability in question [30,43]. The same point extends toward NVD in general [8]. For these and other reasons, the new (v. 3) standard for CVSS enlarges the dimensions toward temporal and environmental metrics.

For the present purposes, however, the impact and exploitability dimensions are sufficient for soliciting an answers to $\text{RQ}_2$. This choice is also necessitated by the paper's focus on NVD, which does not currently provide full CVSS v. 3 information [29]. Despite of this limitation, a correlation between the six CVSS metrics and $\Delta_i$ could be expected due to the fairly detailed criteria used for the manual classification. Complex vulnerabilities with severe impact may require more evaluation work than trivial vulnerabilities; a remote buffer overflow vulnerability is usually more difficult to interpret compared to a trivial cross-site scripting vulnerability. Also the reverse direction is theoretically possible; more effort may be devoted for high-profile vulnerabilities [18]. Either way, $\text{RQ}_2$ seems like a sensible hypothesis worth asking.

With regard to statistical modeling, the three impact metrics and the three exploitability metrics are included in the models as so-called dummy variables. For each metric, the reference category is marked with a star in the previous two diagrams. For instance, INTEGRITY is expanded into two dummy variables, INTEGRITY (PARTIAL) and INTEGRITY(COMPLETE), say, the effects of which are compared against INTEGRITY(NONE), which cannot be included in the models due to multicollinearity. The same strategy applies to the metrics used for evaluating $\text{RQ}_1$. Namely, the annual effects are proxied through 18 dummy variables that record the year at which a vulnerability was published according to $\tau_{\text{CVE}_i^a}$. Because only five vulnerabilities were published in the 1980s and a negiligle amount (about 1.8 %) in the 1990s, the reference category for the annual dummy variables is formed by collapsing all vulnerabilities published prior to 2000 into a single group. Given the two CVSS dimensions and the dummy variable approximation for the annual trend, three model matrices ($\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$) are used in the statistical computation:

$$\begin{cases} \text{M}_1 : \mathbf{X}_1 = [\mathbf{1}, \mathbf{X}_{\text{IMPACT}}], \\ \text{M}_2 : \mathbf{X}_2 = [\mathbf{X}_1, \mathbf{X}_{\text{EXPLOITABILITY}}], \\ \text{M}_3 : \mathbf{X}_3 = [\mathbf{X}_2, \mathbf{X}_{\text{ANNUAL}}]. \end{cases} \tag{3}$$

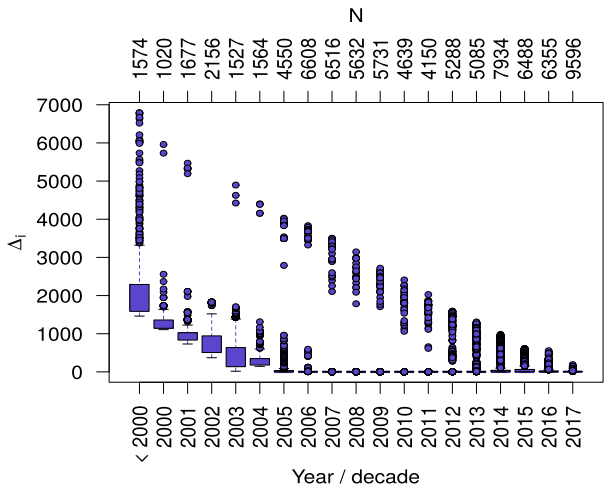


**Fig. 4.** Annual time delays (based on $\tau_{\text{CVE}_i^a}$).

The first model $M_1$ regresses $\boldsymbol{\Delta} = [\Delta_1, \ldots, \Delta_n]'$ against a constant represented by a $n$-length vector of ones, $\mathbf{1}$, and the six impact dummy variables present in the $(n \times 6)$ matrix $\mathbf{X}_{\text{IMPACT}}$. The second model is identical except that further six dummy variables are included for measuring the exploitability dimension. The third and final model includes all information used.

Despite of the growing number of CVEs processed from the circa mid-2000s onward [32], the time delays for CVSS processing have steadily decreased over the years. As can be seen from Fig. 4, there have been no extreme outliers in recent years, meaning that most of the right tail in Fig. 2 is attributable to older CVEs. A possible but speculative explanation is that the work done to update old CVEs with CVSS (v. 2) information has mostly been completed.

The strong decreasing trend is likely to support a positive answer to the research question $RQ_1$. Given this prior expectation, the main interest in the forthcoming analysis relates to the statistical effect of the impact and exploitability metrics when also the annual trend is modeled. One strategy for evaluating the research question $RQ_3$ is to compare the models $M_1$ and $M_2$ against the full information model $M_3$. If the CVSS metrics provide statistical power for predicting $\boldsymbol{\Delta}$, this power should be visible also when the decreasing annual trend is controlled for.

## 3. Results

The response $\boldsymbol{\Delta}$ represents a count data vector; each observation in the vector counts the days between CVE and CVSS publications in NVD. Thus, a Poisson regression model provides a natural starting point for modeling the time delays. The expected value of the response thus is

$$\mathrm{E}(\boldsymbol{\Delta} \mid \mathbf{X}_j) = e^{\mathbf{X}_j \boldsymbol{\beta}}, \tag{4}$$

where $\mathbf{X}_j$ is a given model matrix from (3) and $\boldsymbol{\beta}$ a $k$-length vector of regression coefficients, including the intercept $\beta_1$. This conditional mean is always positive.

However, the model assumes that $\boldsymbol{\Delta}$ is distributed from the Poisson distribution, which, in turn, implies that the mean of the time delays should equal the variance of the delays. As can be concluded from the numbers shown in Fig. 2, this assumption is clearly problematic in the current setting. While $\boldsymbol{\beta}$ is still consistently estimated, the apparent overdispersion, $\mathrm{Var}(\boldsymbol{\Delta}) > \mathrm{E}(\boldsymbol{\Delta})$, affects the standard errors of the regression coefficients, and, hence, the statistical significance of the coefficients. A common solution to tackle the overdispersion problem is to estimate a so-called negative binomial model (NBM) instead, although the conventional ordinary least squares (OLS) regression often works well in applied problems when the response is suitably transformed [15]. Thus, instead of (4), consider that the conditional mean is given by an OLS regression.

$$\mathrm{E}(\ln[\boldsymbol{\Delta} + 1] \mid \mathbf{X}_j) = \mathrm{E}(\tilde{\boldsymbol{\Delta}} \mid \mathbf{X}_j) = \mathbf{X}_j \boldsymbol{\beta}, \tag{5}$$

such that

$$\hat{\boldsymbol{\beta}}_a = \min_{\boldsymbol{\beta}} (\tilde{\boldsymbol{\Delta}} - \mathbf{X}_j \boldsymbol{\beta})' (\tilde{\boldsymbol{\Delta}} - \mathbf{X}_j \boldsymbol{\beta}) \tag{6}$$

When applied to the full model matrix $\mathbf{X}_3$, the adjusted coefficient of determination is 0.64 for this OLS regression. In other words, the general model performance is quite decent, given the limited amount of information used to model the severity assignment timelines. Moreover, only three coefficients in $\hat{\boldsymbol{\beta}}_a$ are not significant at the conventional $p < 0.05$ threshold. By further testing the joint significance of the dummy variable groups with a $F$-test, all groups are significant at a $p < 0.001$ level. Also the combined forward-stepwise and backward-stepwise algorithm (as implemented in the `step` function for R) retains all coefficients in $\hat{\boldsymbol{\beta}}_a$. As is common in applied problems [35], the $\tilde{\boldsymbol{\Delta}} = \ln(\boldsymbol{\Delta} + 1)$ transformation does not account for the high positive skew; therefore, another test can be computed by using an R implementation [44] for a consistent covariance matrix estimator [42]. However, the results do not diverge much from the plain OLS estimates; only one additional coefficient is insignificant at a $p < 0.05$ threshold. Finally, analogous conclusions can be reached by estimating a negative binomial regression model with the assumption that $\mathrm{Var}(\boldsymbol{\Delta}) = \mathrm{E}(\boldsymbol{\Delta}) + \phi[\mathrm{E}(\boldsymbol{\Delta})]^2$, where $\phi$ is a parameter to be estimated [19,41]. By again using an R implementation [20], only two coefficients attain $p \geqslant 0.05$.

Thus, based on statistical significance, positive answers would be given to all three research questions. This conclusion would be unwarranted, however. Most of the coefficients in the $M_3$ model are close to zero, irrespective of the estimation strategy. Since all covariates are dummy variables (and, hence, have the same scale), this observation can be illustrated in the form of Fig. 5, which plots the OLS coefficients ($y$-axis) against the corresponding NBM coefficients ($x$-axis), omitting the constant $\hat{\beta}_1$. As can be seen, there are some differences between the two regression coefficient vectors, but these differences apply mostly to the annual effects. In particular, the coefficients for the impact and exploitability dimensions are very close to zero without notable differences between the OLS and NBM estimates. The largest absolute coefficient values are obtained for the annual effects from 2005 to 2017. These coefficients exhibit also the largest differences between the OLS and the negative binomial estimates.

To examine these observations further, the so-called least absolute shrinkage and selection operator (LASSO) provides a good tool.

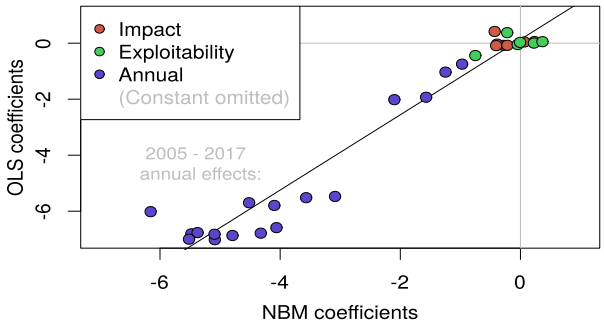


**Fig. 5.** Coefficients from the OLS and NBM regressions ($M_3$).

The LASSO method is a regression model that uses regularization in order to improve prediction accuracy and feature selection. When compared to other regularized regression models, such as the so-called Ridge regression, LASSO can shrink some coefficients exactly to zero. Although the feature selection properties are not entirely ideal for hypothesis testing [21], this property is desirable for further examining whether regularization pushes the coefficients for all of the CVSS metrics toward zero. It should be noted that dropping individual dummy variables based on feature selection is usually unwarranted because interpretation of the coefficients changes—but if all of the impact and exploitability dummy variables are regularized toward zero, there is not much to interpret. If this is the case, there is also no particular reason to consider more complex estimation strategies, such as the so-called group LASSO method [39]. A brief elaboration is required also about the more classical LASSO regressions.

Instead of minimizing the residual sum of squares in (6), LASSO minimizes penalized sum of squares given by

$$\hat{\boldsymbol{\beta}}_b = \min_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (\tilde{\Delta}_i - \mathbf{x}_{ji}' \boldsymbol{\beta})^2 + \lambda \sum_{s=2}^{k} | \beta_s | \right\}, \tag{7}$$

where $\lambda \geqslant 0$ is known as the shrinkage factor, and the scaling by $(1/2n)$ is done to ease comparisons with different sample

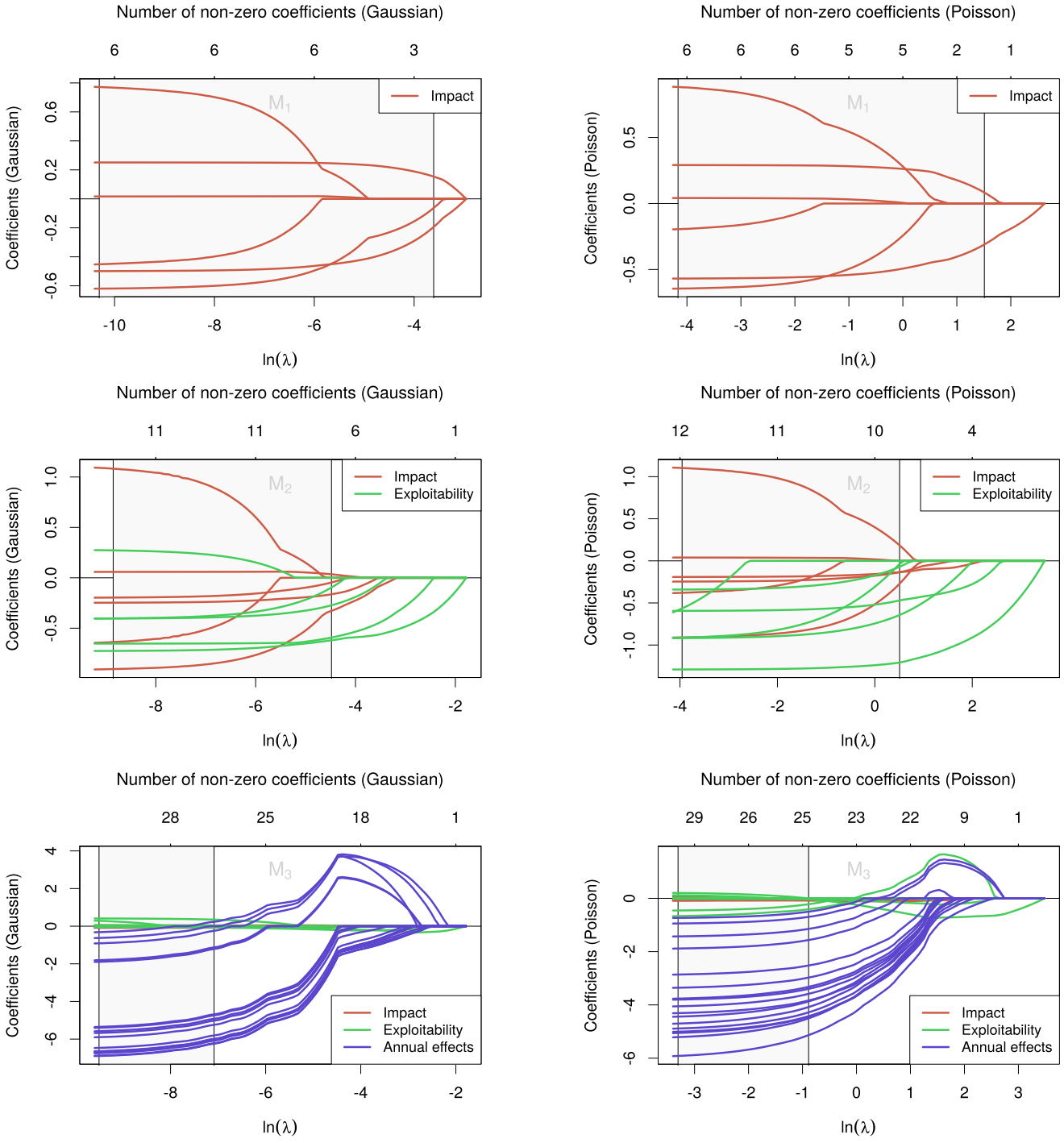


**Fig. 6.** Gaussian LASSO estimates ($\hat{\boldsymbol{\beta}}_b$).

**Fig. 7.** Poisson LASSO estimates ($\hat{\boldsymbol{\beta}}_c$).

sizes [12]. The penalty is given by the $L_1$ norm, that is, the sum of the absolute coefficient values, omitting the constant present in $\mathbf{X}_j$. If $\lambda$ is zero, the solution reduces to the OLS estimates, and when $\lambda \to \infty$, all coefficients in $\hat{\boldsymbol{\beta}}_b$ tend to zero. Despite of the overdispersion, the Gaussian LASSO in (7) can be accompanied with a Poisson LASSO as an additional robustness check.

The so-called quasi log-likelihood for Poisson regression can be obtained by left-multiplying the logarithm of the expected values in (4) by $\boldsymbol{\Delta}$ and subtracting $\mathrm{E}(\boldsymbol{\Delta} \mid \mathbf{X}_j)$ from the result [23]. Given this quasi log-likelihood, for the Poisson regression [12].

$$\mathrm{L}(\boldsymbol{\beta} \mid \boldsymbol{\Delta}, \mathbf{X}_j) = \boldsymbol{\Delta}\mathbf{X}_j\boldsymbol{\beta} - \exp(\mathbf{X}_j\boldsymbol{\beta}), \tag{8}$$

LASSO optimizes

$$\hat{\boldsymbol{\beta}}_c = \min_{\boldsymbol{\beta}} \left\{ -\frac{\mathrm{L}(\boldsymbol{\beta} \mid \boldsymbol{\Delta}, \mathbf{X}_j)}{n} + \lambda \sum_{s=2}^{k} \mid \beta_s \mid \right\}, \tag{9}$$

By again using an R implementation [11], the results from the LASSO computations are shown in Figures 6 and 7 for the Gaussian and Poisson specifications. The coefficient magnitudes are shown in the $y$-axes, the lower $x$-axes represent different values of $\lambda$ in logarithm scale, and the upper $x$-axes denote the number of coefficients not regularized to zero. The shaded region is based on a 10-fold cross-validation: in each plot, the left endpoint of the region corresponds with the value of $\lambda$ that gives the minimum cross-validation error, while the right endpoint is one standard error from this minimum.

In both figures, the models $\mathrm{M}_1$ and $\mathrm{M}_2$ yield large absolute coefficient magnitudes for the CVSS metrics. Furthermore, the coefficients retain their magnitudes rather long as the shrinkage factor increases. For instance, the upper-left plot indicates that none of the impact metrics are regularized to zero in the Gaussian specification until about $\lambda = \exp(-6)$. However, when the annual affects are included in $\mathrm{M}_3$, all of the CVSS metrics are very close to zero particularly with respect to $\hat{\boldsymbol{\beta}}_b$. Although a couple of exploitability metrics retain their magnitudes within the cross-validation region shown in the lower-right plot in Fig. 7, the same conclusion applies more or less also to the Poisson LASSO model. Furthermore, within the cross-validation regions, both $\hat{\boldsymbol{\beta}}_b$ and $\hat{\boldsymbol{\beta}}_c$ compare well to the OLS and NBM coefficient vectors illustrated in Fig. 5. To conclude: when predicting the time delay from CVE publications to CVSS assignments, the actual CVSS content is largely noise; the most relevant readily available information comes with the decreasing annual trend.

## 4. Discussion

This short empirical paper examined the time delays that affect CVSS scoring work in the context of NVD. Three research questions were presented for guiding the empirical analysis based on regression methods. The results are easy to summarize. The CVSS content is correlated with the time delays ($\mathrm{RQ}_2$), but the correlations are spurious; the decreasing annual trend affecting the time delays ($\mathrm{RQ}_1$) also makes the effects of the CVSS content negiligle ($\mathrm{RQ}_3$). Three points are worthwhile to raise about the significance of these empirical findings.

First, the negative answers to $\mathrm{RQ}_2$ and $\mathrm{RQ}_3$ are positive findings in terms of practical applications using CVSS information. Whether the application context is governmental security intelligence systems or commercial security assessment tools, there is currently no particular reason to worry that a NVD data feed would show significant delays for the CVSS information. Likewise, in 2017, there is no reason to suspect that information for severe vulnerabilities would tend to arrive later (or earlier) than information for mundane vulnerabilities. However, this conclusion does not apply to historical contexts, and, moreover, the historically long delays affect also academic research.

Second, the positive answer to $\mathrm{RQ}_1$ is a negative finding in terms of existing academic research; the historically long time delays presumably translate into selection biases in some existing empirical studies using CVSS information. Without naming any particular academic study, consider that a hypothetical article published in the late 2000s used a NVD-based dataset of CVE-referenced vulnerabilities published between 2000 and 2007, say. The long time delays during this period imply that a lot of the vulnerabilities in the dataset could not have had CVSS information. Consequently, some existing academic studies are exposed to difficult questions related to sample selection and missing values, among other issues. This concern is particularly pronounced regarding studies that examine time-sensitive topics such as vulnerability disclosure.

Third, the results echo the recently raised concern about the misuse of statistical significance in the software vulnerability context [22]. It seems that the size of archival material stored to vulnerability databases has surpassed a point after which statistical significance starts to lose its usefulness for inference in applied research. The current rate of new vulnerabilities archived—about 17 per day in 2016—implies that the problem with statistical significance is only going to get worse. The point is particularly important in case CVEs are referenced with other datasets, including big data outputted by intrusion detection and related systems. The regularized regression models used in this paper offer one solution to consider in further applications, but more research is required to assess the existing biases and the potential means for moving forward.

## References


[1] L. Allodi, F. Massacci, Attack potential in impact and complexity, in: Proceedings of the International Conference on Availability, Reliability and Security (ARES 2017), ACM, Reggio Calabria, 2017, pp. 32:1–32:6.

[2] L. Allodi, F. Massacci, Security events and vulnerability data for cybersecurity risk estimation, Risk Anal. 37 (8) (2017) 1606–1627.

[3] M.N. Alsaleh, E. Al-Shaer, Enterprise risk assessment based on compliance reports and vulnerability scoring systems, in: Proceedings of the Workshop on Cyber Security Analytics, Intelligence and Automation (SafeConfig 2014), ACM, Scottsdale, 2014, pp. 25–28.

[4] M. Aslam, C. Gehrmann, M. Björkman, ASArP: automated security assessment & audit of remote platforms using TCG-SCAP Synergies, J. Inform. Secur. Appl. 22 (2015) 28–39.

[5] C. Eiram, B. Martin, The CVSSv2 Shortcomings, Faults, and Failures Formulation, Risk Based Security and the Open Security Foundation (OSF), 2013. Available online in September 2017, <http://www.riskbasedsecurity.com/reports/CVSS-ShortcomingsFaultsandFailures.pdf>.

[6] FIRST, A Complete Guide to the Common Vulnerability Scoring System Version 2.0, FIRST.ORG, 2007. Available online in June 2015: <https://www.first.org/cvss/cvss-v2-guide.pdf>.

[7] L. Gallon, J.-J. Bascou, CVSS attack graphs, in: Proceedings of the Seventh International Conference on Signal Image Technology & Internet-Based Systems (SITIS 2011), IEEE, Dijon, 2011, pp. 24–31.

[8] M. Garcia, A. Bessani, I. Gashi, N. Neves, R. Obelheiro, Analysis of operating system diversity for intrusion tolerance, Software: Pract. Exp. 44 (6) (2014) 735–770.

[9] J. Geng, D. Ye, P. Luo, Predicting severity of software vulnerability based on grey system theory, in: Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Lecture Notes in Computer Science, vol. 9532, Springer, Zhangjiajie, 2015, pp. 143–152.

[10] K. Haldar, B.K. Mishra, Mathematical model on vulnerability characterization and its impact on network epidemics, Int. J. Syst. Assur. Eng. Manage. 8 (2) (2017) 378–392.

[11] T. Hastie, J. Qian, Glmnet Vignette, 2014. Available online in September 2017: <https://web.stanford.edu/hastie/glmnet/glmnet_alpha.html>.

[12] T. Hastie, R. Tibshirani, M. Wainwright, Statistical Learning with Sparsity: The Lasso and Generalizations, CRC Press, Taylor & Francis, Boca Raton, 2015.

[13] H. Holm, K.K. Afridi, An expert-based investigation of the common vulnerability scoring system, Comput. Secur. 53 (2015) 18–30.

[14] S.H. Houmb, V.N.L. Franqueira, E.A. Engum, Quantifying security risk level from CVSS estimates of frequency and impact, J. Syst. Software 83 (2010) 1622–1634.

[15] A.R. Ives, For testing the significance of regression coefficients, go ahead and log-transform count data, Meth. Ecol. Evol. 6 (7) (2015) 828–835.

[16] P. Johnson, R. Lagerström, M. Ekstedt, U. Franke, Can the common vulnerability scoring system be trusted? A Bayesian analysis, IEEE Trans. Depend. Secur. Comput. (2017). Published online in December 2016.
[17] J. Ko, S. Lee, T. Shon, Towards a novel quantification approach based on smart grid network vulnerability score, Int. J. Energy Res. 40 (3) (2016) 298–312.
[18] B. Ladd, The Race Between Security Professionals and Adversaries, Recorded Future Blog, 2017. Available online in November 2017: <https://www.recordedfuture.com/vulnerability-disclosure-delay/>.
[19] J.F. Lawless, Negative binomial and mixed Poisson regression, Can. J. Stat. 15 (3) (1987) 209–225.
[20] M. Lesnoff, R. Lancelot, aod: Analysis of Overdispersed Data, R Package Version 1.3, 2012. Available online in September 2017: <https://cran.r-project.org/web/packages/aod/index.html>.
[21] Z. Li, M.J. Sillanpää, Overview of LASSO-related penalized regression methods for quantitative trait mapping and genomic selection, Theor. Appl. Gen. 125 (3) (2012) 419–435.
[22] F. Massacci, How do you know that it works? The curses of empirical security analysis, in: T.W. Moore, C.W. Probst, K. Rannenberg, M. van Eeten (Eds.), Assessing ICT Security Risks in Socio-Technical Systems (Dagstuhl Seminar 16461), vol. 6, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, 2017, pp. 77–78. Available online in September 2017: <http://drops.dagstuhl.de/opus/volltexte/2017/7039>.
[23] P. McCullagh, Quasi-likelihood functions, Ann. Stat. 11 (1) (1983) 59–67.
[24] P. Mell, K. Scarfone, S. Romanosky, Common vulnerability scoring system, IEEE Secur. Privacy 4 (6) (2006) 85–89.
[25] P.J. Morrison, R. Pandita, X. Xiao, R. Chillarege, L. Williams, Are vulnerabilities discovered and resolved like other defects?, Emp Software Eng. (2017) 1–39, Published online in September 2017.
[26] L. Muñoz-González, D. Sgandurra, M. Barrère, E.C. Lupu, Exact inference techniques for the analysis of bayesian attack graphs, IEEE Trans. Depend. Secure Comput. (2017), Published online in March 2017.
[27] NIST, NVD Data Feed and Product Integration, National Institute of Standards and Technology (NIST), Annually Archived CVE Vulnerability Feeds: Security Related Software Flaws, NVD/CVE XML Feed with CVSS and CPE Mappings (Version 2.0), 2017a. Retrieved in 23 September 2017 from: <https://nvd.nist.gov/download.cfm>.
[28] NIST, NVD Frequently Asked Questions. National Institute of Standards and Technology (NIST), 2017b. Available online in November 2017: <https://nvd.nist.gov/general/faq>.
[29] NIST, Vulnerability Metrics. National Institute of Standards and Technology (NIST), 2017c. Available online in November 2017: <https://nvd.nist.gov/vuln-metrics>.
[30] D.M. Ross, A.B. Wollaber, P.C. Trepagnier, Latent feature vulnerability ranking of CVSS vectors, in: Proceedings of the Summer Simulation Multi-Conference (SummerSim 2017), ACM, Washington, 2017, pp. 19:1–19:12.
[31] J. Ruohonen, Classifying web exploits with topic modeling, in: Proceedings of the 28th International Workshop on Database and Expert Systems Applications (DEXA 2017), IEEE, Lyon, 2017, pp. 93–97.
[32] J. Ruohonen, S. Hyrynsalmi, V. Leppänen, An outlook on the institutional evolution of the European union cyber security apparatus, Govern. Inform. Quart. 33 (4) (2016) 746–756.
[33] J. Ruohonen, S. Hyrynsalmi, V. Leppänen, Modeling the delivery of security advisories and CVEs, Comput. Sci. Inform. Syst. 14 (2) (2017) 537–555.
[34] J. Ruohonen, S. Rauti, S. Hyrynsalmi, V. Leppänen, Mining social networks of open source CVE coordination, in: Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement (IWSM Mensura 2017), ACM, Gothenburg, 2017, pp. 176–188.
[35] J. Rydberg, D.M. Carkin, Utilizing alternate models for analyzing count outcomes, Crime Delinq. 61 (1) (2017) 61–76.
[36] K. Scarfone, P. Mell, An analysis of CVSS version 2 vulnerability scoring, in: Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009), IEEE, Lake Buena Vista, 2009, pp. 516–525.
[37] D.-H. Shin, H. Kim, J. Hwang, Standardization revisited: a critical literature review on standards and innovation, Comput. Stand. Interf. 38 (2015) 152–157.
[38] I. Stine, M. Rice, S. Dunlap, J. Pecarina, A cyber risk scoring system for medical devices, Int. J. Crit. Infrastruct. Protect (2017). Published online in April 2017.
[39] D. Vidaurre, C. Bielza, P. Larrañaga, A survey of $L_1$ regression, Int. Stat. Rev. 81 (3) (2013) 361–387.
[40] J.A. Wang, M. Guo, H. Wang, L. Zhou, Measuring and ranking attacks based on vulnerability analysis, Inform. Syst. e-Bus. Manage. 10 (4) (2012) 455–490.
[41] F. Wei, G. Lovegrove, An empirical tool to evaluate the safety of cyclists: community based, macro-level collision prediction models using negative binomial regression, Accid. Anal. Prevent. 61 (2013) 129–137.
[42] H. White, A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity, Econometrica 80 (4) (1980) 817–838.
[43] A.D. Younis, Y.K. Malaiya, Comparing and evaluating CVSS base metrics and Microsoft rating system, in: Proceedings of the IEEE International Conference on Software Quality, Reliability and Security (QRS 2015), IEEE, Vancouver, 2015, pp. 252–261.
[44] A. Zeileis, Econometric computing with HC and HAC covariance matrix estimators, J. Stat. Software 11 (10) (2004) 1–17.
[45] X. Zhu, C. Cao, J. Zhang, Vulnerability severity prediction and risk metric modeling for software, Appl. Intell. 47 (3) (2017) 828–836.