University of Otago

Department of Computer Science

COSC480 Project Report

An Exploration of the Common Vulnerability Scoring System

Author:
Jake Norton (5695756)

Supervisor(s):
Dr. David Eyers
Dr. Veronica Liesaputra

July 26, 2024

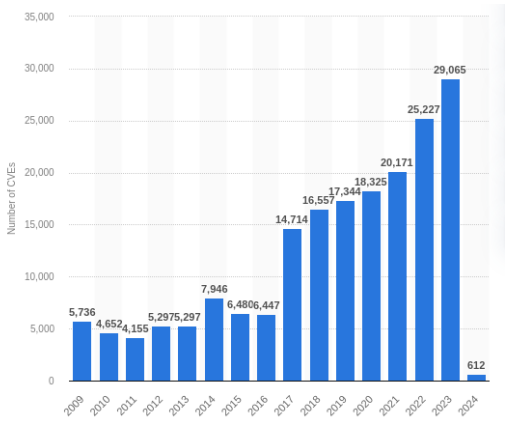

Figure 1: Number of new CVEs by year

Abstract

The Common Vulnerability Scoring System(CVSS [9]) is designed to produce scores for software vulnerabilities. Such a system is needed in order to triage the sheer number of new vulnerabilities being released every year. We cannot keep up with the amount of CVSS scores that need to be produced, as such we need a way to generate them. There is precedent to using machine learning, specifically in more recent times, large language models(LLMS) to accurately predict these CVSS scores. [6] However, there is a general focus on only using the National Vulnerability Database [6] [1] [3], it would be ideal if there was more than one source for the ratings, not only for cross validation, but also for an increase in data. Before we use any extra data sources, it will be interesting to do a comparison between the different sources, to see if we can get an estimate accuracy for each of the metrics within the scoring system. Additionally we should know how good of a system CVSS is and whether or not there are better alternatives. Unfortunately CVSS(version 3.1 [8]) is a flawed metric, hopefully once 4.0 begins to be used commonly that will solve some of these issues. However, the ability to predict a metric based on a short text description is still useful and a focus on the interpretability of such a system remains important.

1 Introduction

Last year there were 29,065 new vulnerabilities. This is a number that is only going up year on year. Now, we need a way to record these vulnerabilities, and we do that using the common vulnerabilities and exposure system called CVEs. From these CVEs, CVSS scores can be calculated or generated. The National Vulnerability Database(NVD [18]) takes the CVEs and enriches them with CVSS data. They are not the only place to do so, however in terms of research they are often the main or sole data provider. [6] [1] [3] I explored other options, the main obvious one being the MITRE [16] database, as it is the main database for CVEs and they also have a decent chunk of CVEs enriched with CVSS scores. As a guideline to my investigation between the two databases I used the same method as the paper *Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis.* [13] This paper tries to see how much different data sources agree on the scoring of a CVE, and thereby gaining insight into the potential of a ground truth value. Unfortunately since that paper in 2016, many of the databases they compared are either unavailable or in archival status. However, following their method still allows for insight between the two chosen databases, NVD and MITRE. This analysis shows that the databases do fundamentally rate CVEs differently. The uncertainty between the two can therefore be an indicator going forward when analysing generated CVE scores, as it is likely that the model will also struggle in similar places to where the human evaluators did. In addition to this analysis, I looked into the CVSS system itself. There have been many complaints laid against CVSS [11] [23] [22], the main ideas being:

- Mathmatical operations on categorical values,
- No mathmatical basis for the formula,
- Lack of likelihood,
- Scope has too much influence,
- Identity crisis,

These are all on going issues and should colour how we use the CVSS system. It should only be used as a way to triage our vulnerabilities, any further prioritisation should be done by more specific inqueries or perhaps other systems.

2 Background

Vulnerabilities are stored in a consistent system called Common Vulnerabilities and Exposures(CVE [14]).

Here is an example CVE

- Unique Identifier: CVE-2024-38526
- Source: GitHub, Inc.
- Published:06/25/2024
- Updated:06/26/2024
- Description: pdoc provides API Documentation for Python Projects. Documentation generated with ‘pdoc –math‘ linked to JavaScript files from polyfill.io. The polyfill.io CDN has been sold and now serves malicious code. This issue has been fixed in pdoc 14.5.1.

Sourced from NVD CVE-2024-38526 Detail [17]

This has a unique identifier, which is given by one of the CVE numbering authorities(CNA [15]), such as GitHub, Google and many other organizations.[CVE list of partners [5]] The description is the most important part in our case. This should give some information about the vulnerability, what can be exploited (device / software component), how is the product affected if the vulnerability is exploited. In this case we have the library PyDoc and this links to the polyfill.io CDN. Ideally there would be something in the description which relates to every metric, unfortunately these descriptions are not necessarily suited to machine learning as the people writing the descriptions are expecting a lot of intrinsic knowledge.

The Common Vulnerability Scoring System

CVSS scoring is a high level way to break up vulnerabilities into different categories. Organisations can use it to choose which vulnerability to focus on first. CVSS is broken up into 3 distinct sections, base, temporal and environmental scores.

For brevity I will only show the specifics of CVSS 3.1 [8]as this is by far the most commonly used version, even if it is not the most recent.

Base Score

- Attack Vector: Defines the avenues of attack that the vulnerability is open to. The more open a component is, the higher the score. This can have the values Network, Adjacent, Local and Physical.
- Attack Complexity: How complex the attack is to orchestrate. What are their prerequisites, how much domain knowledge / background work is necessary, how much effort does the attacker need to invest to succeed. This can have the values Low or High. Low gives a higher base score.
- Priviledges Required: The degree of privileges the user needs to complete the attack. Generally ranging from None, Low(e.g User level privilege), High(e.g Administrator). The lower the privilege the higher the base score.
- User Interaction: If the exploit requires another human user to make the attack possible, E.g clicking a phishing link. This is either None or Required, the score is highest when no user interaction is required.
- Scope: Defines if the attack can leak into other security scopes. E.g access to one machine gives the ability to elevate privileges on other parts of the system. This can take Unchanged or Changed, the score being highest when a scope change occurs.
- Confidentiality Impact: Detemines what is the impact on the information access / disclosure to the attacker. This can be High, Low or None with High adding the most to the base score.
- Integrity Impact: Refers to the integrity of the information within the component. I.e could the data have been modified by the attacker. This has High, Low or None as categories with High adding the most to the base score.
- Availability Impact: Refers to the impact of the attack on the availability of the component. E.g the attacker taking the component off the network, denying the users access. This can haved High, Low and None with High adding the most to the base score.

This is a summarized version of the 3.1 specification document provided by FIRST. [8]

Temporal

This could be:

- Exploit Code Maturity: The state of the attack itself, e.g has this exploit been pulled off in the wild or is it currently academic.
- Remidiation Level: Broadly, whether the exploit in question has been patched,
- Report Confidence: The degree of confidence in the CVE report itself, the report may be in early stages where not all of the information is known.

This is a summarized version of the 3.1 specification document provided by FIRST. [8]

Temporal metrics would be useful in general for a CVSS score, however NVD do not store these temporal metrics. As far as I can tell there is no reason given for this specifically, though discourse (Stack exchange post) [2] around the subject suggests that this is due to a lack of verifiable reporting. From my perspective both remidiation level and report confidence feel like they could have scores attributed to them, however finding verifiable reports on the exploits seen in the wild does seem more tricky. There are two relatively new organisations on this front, Cybersecurity & Infrastructure Security Agency(CISA, public sector) and inthewild.org(private sector [4]).

2.1 Data Options

In 2016 when Johnson et al [13] did the original paper, they had access to 5 different databases. Unfortunately only 2 of these remain for modern data, there are some others, but essentially they are either in archival status or they are proprietary. I have managed to acquire the data from the original paper, however it is in a much different format(XML vs JSON), and Pontus Langstrom, one of the contributors to the project said it would be akin to an archaeological dig. Additionally as I did not plan to make this the full focus of the project, that will sit dormant for now.

National Vulnerability Database

The National Vulnerability Database is the defacto standard dataset used for CVSS generation research. [6] [1] [3] This makes a lot of sense as it is built for the purpose with a team dedicated to enriching CVEs with CVSS scores. The dataset I am using was retrieved using the NVD API in March 2024 and contains ∼100000 CVEs enriched with CVSS scores. This comes in a easy to use format, in a consistently formatted JSON dump.

MITRE Database

MITRE is the defacto database for the storage of CVEs themselves, however they do contain ∼40000 CVEs enriched with CVSS 3.1 scores. These are also in a JSON dump retrieved also in March 2024. The format for usage is a bit more cumbersome to use. The CVSS scores are only stored as CVSS vector strings(a simple text encoding [20]). These are not hard to parse, though they are stored slightly different between versions, as well as sometimes being inconsistent(∼5000 had temporal metrics within the vector strings in the MITRE database).

2.1.1 Priliminary Data exploration

The scorers for both NVD and MITRE do rate CVEs reasonably similar, one pattern you can see as shown by Fig 2, is that NVD generally give the most common categorical output more ratings. They are less spread out across the full range of values. In addition, if we

look at the attackComplexity metric, there is a reasonably large difference in how they are rated, MITRE rate a lot more of the metrics with a low score. This points to some of the difficulty with this kind of rating system, while in theory there is a true value for these metrics, it requires knowledge of the whole space around each of the vulnerabilities, this knowledge will always vary marker to marker.

3 Related Work

1. Johnson et al. in the paper *Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis* [13] conducted a study of the state of CVSS databases and there accuracy in 2016. They found NVD to be the most correct database, and that we can trust the CVSS scoring system as a whole in the consistency of how scorers rate CVEs.

2. Costa et al. in the paper *Predicting CVSS Metric via Description Interpretation* tested generation of CVSS from CVE descriptions with a range of large language encoder-only models. The achieved state-of-the-art results with the Distilbert model [21]. They also improved the score with text preprocessing(e.g lemmatization) and looked into interpretability of the models using Shapley values.

3. Jiang and Atif in the paper *An Approach to Discover and Assess Vulnerability Severity Automatically in Cyber-Physical Systems* [12] creates a novel pipeline for vulnerability assessment. They used multiple data sources and a majority voting system to decrease the chance of badly scored CVEs.

4. Henry Howland in the paper *CVSS: Ubiquitous and Broken* broke down issues with the CVSS system, namely, "lack of justification for its underlying formula, inconsistencies in its specification document, and no correlation to exploited vulnerabilities in the wild, it is unable to provide a meaningful metric for describing a vulnerability's severity"

4 Methods

4.1 Hierarchical Bayesian Model

The analysis between the two databases is done with a hierarchical bayesian model. This type of model is suitable when you expect the population to be similar in some respects but different in others. In this case they share common knowledge towards the vulnerability mostly, but differ the experience of the people rating the metrics. [13] The model is similar to the original model(see section 4.1 of [13]), it assumes that there exists a true value for each CVSS dimension, but the database sample may not be that true value. We represent the inaccuracies with a confusion matrix. The A difference from the original paper is that there is no longer consistently 3 variables for each CVSS metric, varying from 2 to 4 categorical choices.

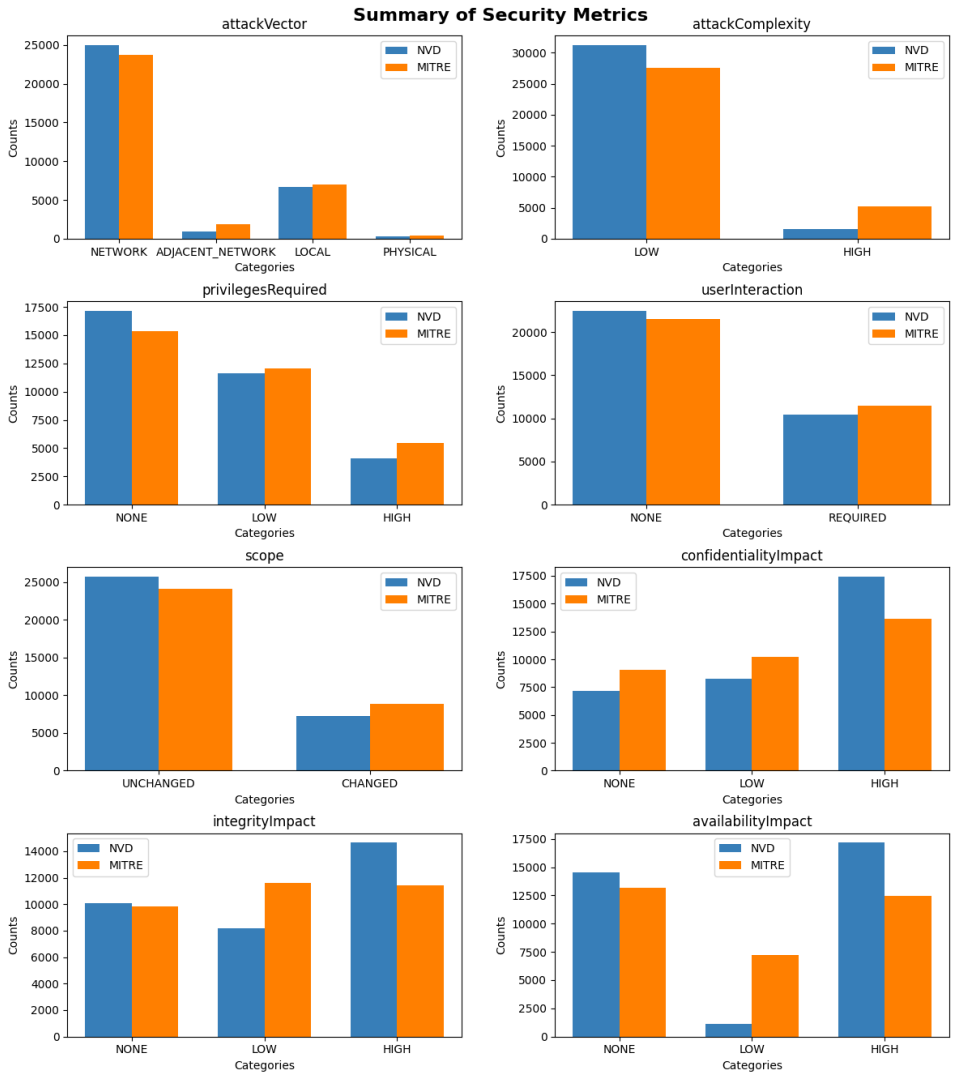

Figure 2: Comparison of CVSS ratings between MITRE and NVD

The confusion matrix CVSS dimension *confidentiality impact*

$$\Pi_c i = \begin{bmatrix} \pi_{nn} & \pi_{nl} & \pi_{nh} \\ \pi_{ln} & \pi_{ll} & \pi_{lh} \\ \pi_{hn} & \pi_{lh} & \pi_{hh} \end{bmatrix} \tag{1}$$

where π_{nn} denotes the probability that the current database correctly assigns the random vulnerability *none* when the actual value is the same. π_{nl} and π_{nh} represent when *none* was not the actual truth value.

4.1.1 Priors

The priors for the categorical variables were set up with uninformative priors using a Dirichlet distribution, this will give a uniform prior over they probability space for all categorical options. This is done to not colour the outcome of the results based on prior belief, but either way this prior will have little impact for any categorical metric which has more than the number of options for that metric.

The confusion matrices also need priors, for this they will also be a Dirichlet distribution, however as we do want to add some initial belief to this, in that the people producing scores are not acting completely randomly, and are likely to be right more often. These are still weak priors as the number of observations is in the thousands.

4.1.2 Estimation

This follows, the Bayesian approach, where to estimate the parameters, you take the prior beliefs, take an observation, and update these believes to produce posterior beliefs. In this case the method involves using Markov Chain Monte Carlo methods. Broadly this allows for simulating the data based on the previously created distribution by sampling values that the model has high belief would be from the target distribution. This allows for accurate sampling without having all of the data. The original paper used JAGS [19] and R. As I am more familiar with python I did try pyjags [24] however I did not find great success. Instead I used the pymc library to help with the modelling, it fulfilled the same tasks that JAGS did with the original paper [13]

4.2 CVSS Prediction

Cody Airey–a classmate of mine– has been working on a similar problem. He has been repoducing some results from Costa et al. [6]. My choice of model for CVSS prediction will very much bootstrap of his work and that which is surrounding it. So far, a strong contender for state-of-the-art model for predicting CVSS metrics from CVE descriptions is the distilbert model [21]. This is a variant of BERT [7], a pre-trained deep bidirectional transformer large language model developed by Google. Distilbert has advantages over the models in terms of performance, but also on speed of training as well as size / memory footprint of the model.

4.2.1 Training

The model is trained separately for each metric. Following Airey's method, each of the eight models were trained on five different data splits to allow for a standard deviation to be calculated, in order to aid in reducing the chance of a 'good' data split effecting the results. The difference between Costa et al. & Airey's work and mine is that this model was trained on a combination of NVD and MITRE data. This was converted to the same format, a CSV containing the descriptions and the CVSS scores. This does mean there are now ∼40000 duplicate CVEs and ∼140000 CVEs enriched with CVSS scores.

5 Results

5.1 Bayesian Analysis

The results for the database analysis will be shown through the confusion matrices for the estimated accuracy of both NVD and MITRE. Unfortunately it is difficult to compare my results to the original paper as they are on a totally different dataset. However I will note that in general the estimated accuracy for both datasets is much lower than the scores they were getting. Across the board NVD often had ∼90% accuracy for the highest accuracy field of any metric, with Confidentiality as a clear outlier as seen from Table 5.1.

Some general trends are that NVD(see Figure 3) have more extreme estimated accuracies. They do better for the higher frequency options, for the *Low* on *Attack Complexity* for example NVD have 98% estimated accuracy and 66% for *High* versus MITRE(see Figure 4) for the *Low* score 87% and *High*. Instead of further analysing in this way, I will point out some of my worries around these results, not that they are wrong per se, but that they do not really tell us anything extra than that shown by in Figure 2, except perhaps it is helpful to see it in a percentage estimated accuracy instead of a proportion. Unfortunately this is outside of my strengths, I did some cursory exploration into if a doing this sort of analysis between two populations like this does make sense, the discourse here [10] suggests that such a thing can be done, though it also suggests the need for more informative priors. This may apply in my case, and I intend on getting an expert opinion closer to home, however that will be after this report.

Johnson et al. talk about the reliability of their results saying this in section 7.1 of [13]:

"reliability concerns are discussed. In this study we use five different scoring instruments the databases. If some of these are generally correct, but some are generally incorrect, will not the scores of the incorrect ones still affect our beliefs about the actual values, thus worsening reliability? It turns out that this is not the case. In a simulation, two scorers were set up to be completely aligned, scoring 90% complete impact in one of the CIA dimensions, while a third was unaligned and scoring only 10% complete impact. Initially, all scorers are equally credible, but the gradual accrual of evidence impacts scorer credibility (as well as beliefs about the actual CVSS score distributions). Thus, as the third scorer rarely matched the other two, its credibility eroded, thus shrinking the weight of its

Table 1: Confusion Matrices for NVD on CVSS version 2.0 from Johnson et al. [13]

Access vector			
	N	A	L
N	0.99	0	0
A	0.21	0.71	0.08
L	0.05	0.0	0.95

Access complexity			
	L	M	H
L	0.54	0.29	0.17
M	0.02	0.88	0.1
H	0.01	0.08	0.91

Authentication			
	N	S	M
N	0.99	0.01	0
S	0.04	0.95	0.01
M	0.19	0.2	0.6

Confidentiality			
	C	P	N
C	0.4	0.2	0.2
P	0.2	0.4	0.19
N	0.2	0.21	0.4

Integrity			
	C	P	N
C	0.91	0.08	0.01
P	0.05	0.93	0.02
N	0.02	0.07	0.92

Availability			
	C	P	N
C	0.92	0.07	0.01
P	0.07	0.91	0.02
N	0.02	0.11	0.87

advise." [13]

Unfortunately in my case I do not have the advantage of a potential third or more scorer. This leads me to believe that there is a chance that incorrect scores can end up corrupting the results.

5.2 CVSS Prediction

Below in Table 5.2 Table 5.2 show the results of the Distilbert model trained on the combination of NVD and MITRE data. Unfortunately this basically has a purely negative effect on all metrics, with the caveat that some of the standard deviations are lower. Additional note is the balanced accuracy for some metrics looks a bit weird, I believe that is due to the model not outputing some of the categorical options for that metric. This applies to all the balanced accuracy apart from the Priorities Required(PR) and Confidentiality(C). As this was done only recently I have not looked into this issue in-depth to see why this is happening, but I will in the future. As to why the model performs worse, my theory is that the added data, and therefore the overlapping CVEs with different scores confuses the model. I thought this may have given the model a better chance of

generalising on the data, however this does not appear to be the case.

Metric	Model	AV	AC	PR	UI
Accuracy	DistilBERT-Cody	**91.28 ± 0.26**	95.64 ± 0.68	**82.77 ± 0.24**	**93.86 ± 0.19**
	DistilBERT-Jake	72.81 ± 0.32	92.62 ± 0.15	81.18 ± 0.18	66.35 ± 0.24
F1	DistilBERT-Cody	**90.98 ± 0.31**	93.85 ± 1.39	**82.53 ± 0.26**	**93.82 ± 0.19**
	DistilBERT-Jake	61.36 ± 0.42	89.08 ± 0.22	80.96 ± 0.19	52.93 ± 0.31
Bal Acc	DistilBERT-Cody	**67.88 ± 2.11**	55.82 ± 7.23	**75.98 ± 0.47**	**92.46 ± 0.21**
	DistilBERT-Jake	25.00 ± 0.00	50.00 ± 0.00	75.18 ± 0.31	50.00 ± 0.00

Table 2: Comparison of the effects of the X pre-trained models on the CVSS v3.1 dataset (Part 1).

Metric	Model	S	C	I	A
Accuracy	DistilBERT-Cody	**96.38 ± 0.09**	**86.24 ± 0.20**	**87.15 ± 0.10**	**88.70 ± 0.10**
	DistilBERT-Jake	80.21 ± 0.16	82.45 ± 0.11	45.71 ± 0.26	52.53 ± 0.23
F1	DistilBERT-Cody	**96.30 ± 0.10**	**86.09 ± 0.21**	**87.11 ± 0.10**	**88.04 ± 0.11**
	DistilBERT-Jake	71.40 ± 0.22	82.34 ± 0.12	28.68 ± 0.28	36.18 ± 0.27
Bal Acc	DistilBERT-Cody	**91.57 ± 0.43**	**82.70 ± 0.36**	**85.81 ± 0.10**	**64.01 ± 0.13**
	DistilBERT-Jake	50.00 ± 0.00	79.85 ± 0.23	33.33 ± 0.00	33.33 ± 0.00

Table 3: Comparison of the effects of the X pre-trained models on the CVSS v3.1 dataset (Part 2).

6 Discussion

6.1 Should we use CVSS?

Note, a lot of this subsection is very rough/ transcript from presentation and I will be cleaning it up

So CVSS has an identity crisis. I found that many people weren't happy with CVSS, and this is one of the reasons. So when they released CVSS 2.0, first said, IT management need to prioritize these vulnerabilities and remediate those that pose the greatest risk. The Cohen vulnerability scoring system is an open framework that addresses this issue. However,

When they released 3.1, which is the version that I've been doing all this analysis on, they said CVSS measures severity and not risk. So what is it? Well, it's kind of somewhere in between. The main takeaway is that you cannot just take the CVSS scores, especially the base scores as the BLNL. They don't tell you all the information.

The difference between severity and risk is a little bit up in the air, but generally the idea is that risk has more to do with likelihood. Unfortunately, this base score is used by many companies.

There's a few US government and large industries that use it just for this. And there's another reason why this is a problem is because you can have poor ratings like this one. So this was a CVE vulnerability that came from the curl library. And this is the description, integer overflow vulnerability and tool operate.c. In curl, this version very large value as the retry delay. So essentially they didn't handle

the user putting in a value that was over the integer limit, and so it could potentially overflow, which resulted in undefined behavior. This gave a score of 9.8. Now, Daniel Stenberg, the original founder, did not agree with this, as he said this was a bug that they had fixed back in 2019, and this is something that came out recently.

And so he went to MITRE to try and get the CVE removed, which didn't happen. He may, I have just noticed that he is now part of the CVE numbering authorities. So he's taken control of his own CVE vulnerability disclosure. So.

Maybe he's trying to get it removed. I'm not sure if he's been successful. He then went on to NVD to get the score lowered, which he was successful in doing, and now it is a 3.3. But there was a bunch of time wasted, unfortunately, doing that. And it's just something to keep in mind. You cannot just blindly trust the CVS score.

Additionally, CVSS, this is the formula for CVSS, is a little bit suspect. There are many magic values here which are in no way backed up by anything mathematical, especially considering that they are adding and multiplying categorical scores. So this is a.

distribution of all the possible CVSS scores. So we can see it looks like a vaguely normal-like distribution. However, as we just saw briefly from that formula, there's no mathematical basis as to why this is a normal distribution, and it has been posited that they just deliberately made it like this, which is interesting.

And then the scores are suspiciously similar. So if we look, this is the distribution of all the scores in 2019 of all the count of reports. And.

We can see here that they're vaguely swayed towards the upper end into these distinct buckets, probably because of some rounding. And then if we look at 2020, very similar. And we look at 2021, very similar. And if we look at the R squared...

correlation between the two, you can see that it's 0.99, which is very high and kind of un-

in this sort of setting, so a little bit strange. So then onto some future work. Surprisingly, I am gonna stay doing research on CVSS and CVEs. My primary focus will be looking into how can we make CVE, how can we clean up the CVE data so that it is more useful to machine learning models. Cody will, up next, will show you a few examples of,

CVE descriptions, but just know that there are many, many bad examples that do not give us good descriptions of the whole problem. So I'm going to be working on that as well as doing some clustering of the data and seeing how, seeing if I can maybe map it to CWEs, the Common Weakness Enumeration, which has kind of been done, but it'd be interesting to do that in a more machine learning type sense, and other things along that vein not fully figured out yet. Awesome. Thank you.

CVSS has an identity crisis. Throughout its history, when originally released it was

touted as a solution to the task of prioritising CVE remediation as well as an assessment of risk, "IT management must identify and assess vulnerabilities across many disparate hardware and software platforms. They (IT management) need to prioritize these vulnerabilities and remediate those that pose the greatest risk. The Common Vulnerability Scoring System (CVSS) is an open framework that addresses this issue"

However, due to a lot of feedback from the community and security agencies, when FIRST released version 3.1, the authors state "CVSS Measures Severity Not Risk".

Severity vs Risk

The severity ideally is a measure of the impact(worst case? Or different levels?). Risk is the likelihood of the event happening. However in CVSS this is a bit muddied as even the Base score includes some aspects which defaults to worst case risk.

Should use some way of temporal / environmental. These are included in CVSS however, they are often not used and it may be better to use EPSS, which gives a numerical value of the likelihood of the event happening in 30 days based on previous history

FIRST give this definition of severity vs risk:

- CVSS Base scores (CVSS-B) represent "Technical Severity" Only takes into consideration the attributes of the vulnerability itself It is not recommended to use this alone to determine remediation priority
- Risk is often a religious topic... but... CVSS-BTE scores take into consideration the attributes of the... Base Score Threat associated with the vulnerability Environmental controls / Criticality

"Risk is a measure of the extent to which an entity is threatened by a potential circumstance or event, and is typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence"

There have been myriad complaints about this topic, generally due to the nature of how CVSS is often used, especially in the US. There are many known occurences of the US government mandating the use of CVSS base score as the primary framework used to prioritize remediation.

6.2 Reflection

The Bayesian analysis portion of the project in my head was initially supposed to be a small diversion for interest on how the CVSS data landscape has changed since 2016. However it ended up taking a good part of the project. There were a few confounding factors that caused this.

1. The data landscape has changed so much that it is essentially impossible to recreate the results
2. I do not have a statistics background
3. I should have asked for more help

I did realise the data landscape had changed sufficiently, however the analysis still seemed useful for some interpretability of machine learngin models going forward. I am unsure if this is still true or not.

The statistics I have done has mainly been ingested through osmosis. I was under the impression–whether through ignorance or hubris– that I could learn enough statistics to be able to understand and execute on this problem. I did spend time getting an understanding on the fundamentals, Markov chains, Monte Carlo methods, and Bayesian statistics(this I had done some study on before). However, this was not enough to understand statistics enough to really understand what was going on. This led me to make many mistakes in my implementation before getting to something that seems reasonable. I think if my data had been similar to the original study, this would have been less of a problem. The issues to do with only having two different populations would not have come up and I would have not had the same unease at the results.

I have asked for some help with my understand of the topic, but I did not go to statisticians who could have given me access to a deeper level of knowledge on the specific topic, that being hierarchical bayesian models and MCMC Gibbs sampling.

7 Conclusion

Need to do this!

References

[1] M Ugur Aksu et al. "Automated generation of attack graphs using NVD". In: *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*. 2018, pp. 135–142.

[2] Anonymous. *cvss v3 and v3.1 missing temporal metrics exploit code maturity and remediation*. https://security.stackexchange.com/questions/270257/cvss-v3-and-v3-1-missing-temporal-metrics-exploit-code-maturity-and-remediation. [Online; accessed July-2024]. 2024.

[3] Hodaya Binyamini et al. "A framework for modeling cyber attack techniques from security vulnerability descriptions". In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 2574–2583.

[4] CISA. *Known Exploited Vulnerabilities Catalog*. https://www.cisa.gov/known-exploited-vulnerabilities-catalog. [Online; accessed June-2024]. 2024.

[5] The MITRE Corporation. *List of Partners*. https://www.cve.org/PartnerInformation/ListofPartners. [Online; accessed June-2024]. 2024.

[6] Joana Cabral Costa et al. "Predicting CVSS Metric via Description Interpretation". In: *IEEE Access* 10 (2022), pp. 59125–59134. DOI: 10.1109/ACCESS.2022.3179692.

[7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[8] FIRST. *Common Vulnerability Scoring System v3.1: Specification Document*. https://www.first.org/cvss/v3.1/specification-document. [Online; accessed February-2024]. 2024.

[9] FIRST. *CVSS landing page*. https://www.first.org/cvss/. [Online; accessed February-2024]. 2024.

[10] Andrew Gelman. *Hierarchical modeling when you have only 2 groups: I still think it's a good idea, you just need an informative prior on the group-level variation*. https://statmodeling.stat.columbia.edu/2015/12/08/hierarchical-modeling-when-you-have-only-2-groups-i-still-think-its-a-good-idea-you-just-need-an-informative-prior-on-the-group-level-variation/. [Online; accessed July-2024]. 2024.

[11] Henry Howland. "CVSS: Ubiquitous and Broken". In: *Digital Threats* 4.1 (Feb. 2022). DOI: 10.1145/3491263. URL: https://doi.org/10.1145/3491263.

[12] Yuning Jiang and Yacine Atif. "An Approach to Discover and Assess Vulnerability Severity Automatically in Cyber-Physical Systems". In: *13th International Conference on Security of Information and Networks*. SIN 2020: 13th International Conference on Security of Information and Networks. Merkez Turkey: ACM, Nov. 4, 2020, pp. 1–8. ISBN: 978-1-4503-8751-4. DOI: 10.1145/3433174.3433612. URL: https://dl.acm.org/doi/10.1145/3433174.3433612 (visited on 02/28/2024).

[13] Pontus Johnson et al. "Can the Common Vulnerability Scoring System be Trusted? A Bayesian Analysis". In: *IEEE Transactions on Dependable and Secure Computing* 15.6 (2018), pp. 1002–1015. DOI: 10.1109/TDSC.2016.2644614.

[14] MITRE. *Common Vulnerabilities and Exposures — CVE® The Standard for Information Security Vulnerability Names*. https://cve.mitre.org/docs/cve-intro-handout.pdf. [Online; accessed July-2024]. 2024.

[15] MITRE. *CVE Numbering Authorities (CNAs)*. https://www.cve.org/ProgramOrganization/CNAs. [Online; accessed May-2024]. 2024.

[16] MITRE. *MITRE landing page*. https://cve.mitre.org/. [Online; accessed February-2024]. 2024.

[17] NVD. *CVE-2024-38526 Detail*. https://nvd.nist.gov/vuln/detail/CVE-2024-38526. [Online; accessed June-2024]. 2024.

[18] NVD. *NVD landing page*. https://nvd.nist.gov/. [Online; accessed February-2024]. 2024.

[19] Martyn Plummer. *JAGS: Just Another Gibbs Sampler*. https://sourceforge.net/projects/mcmc-jags/. [Online; accessed April-2024]. 2024.

[20] Qualys. *CVSS Vector Strings*. https://qualysguard.qualys.com/qwebhelp/fo_portal/setup/cvss_vector_strings.htm. [Online; accessed July-2024]. 2024.

[21] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL]. URL: https://arxiv.org/abs/1910.01108.

[22] Jonathan Spring et al. "Time to Change the CVSS?" In: *IEEE Security & Privacy* 19.2 (2021), pp. 74–78. DOI: 10.1109/MSEC.2020.3044475.

[23] Jonathan Spring et al. "Towards improving CVSS". In: *SEI, CMU, Tech. Rep* (2018).

[24] Michael Nowotny Tomasz Miasko. *PyJAGS: The Python Interface to JAGS*. https://github.com/michaelnowotny/pyjags. [Online; accessed April-2024]. 2024.

Appendix A CVSS figures from other versions

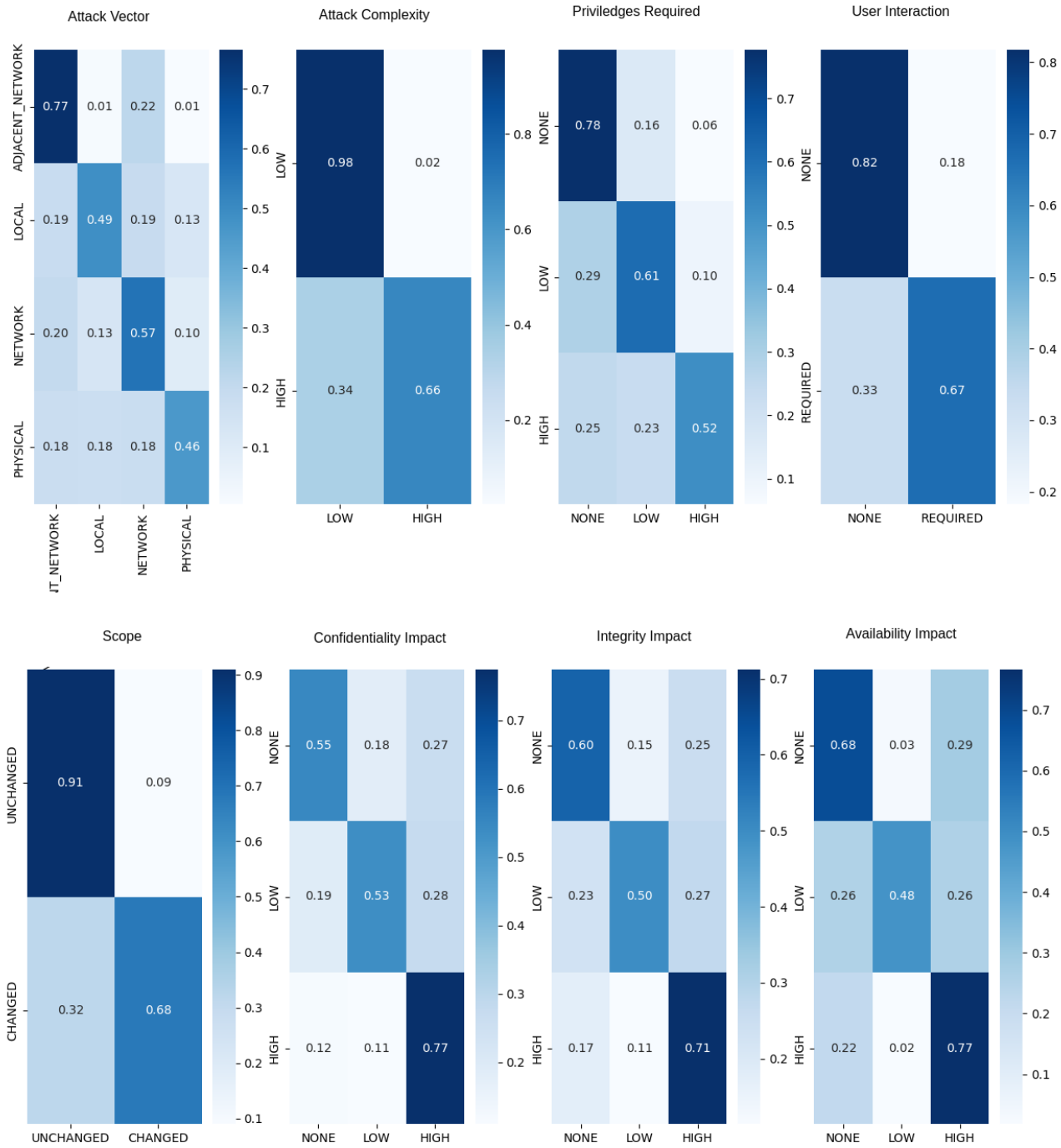

Figure 3: Confusion Matrices for NVD for CVSS version 3.1

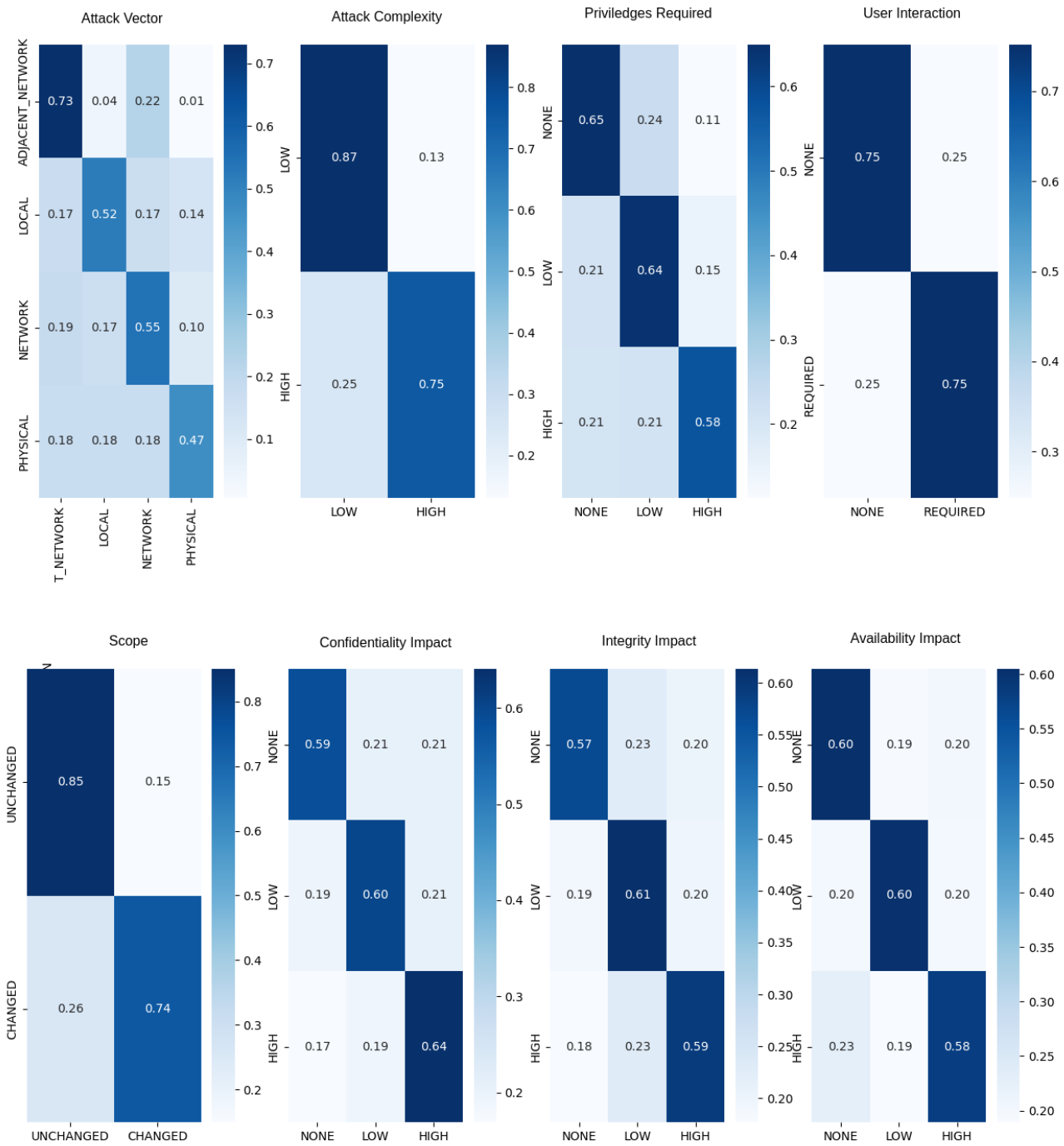

Figure 4: Confusion Matrices for MITRE for CVSS version 3.1

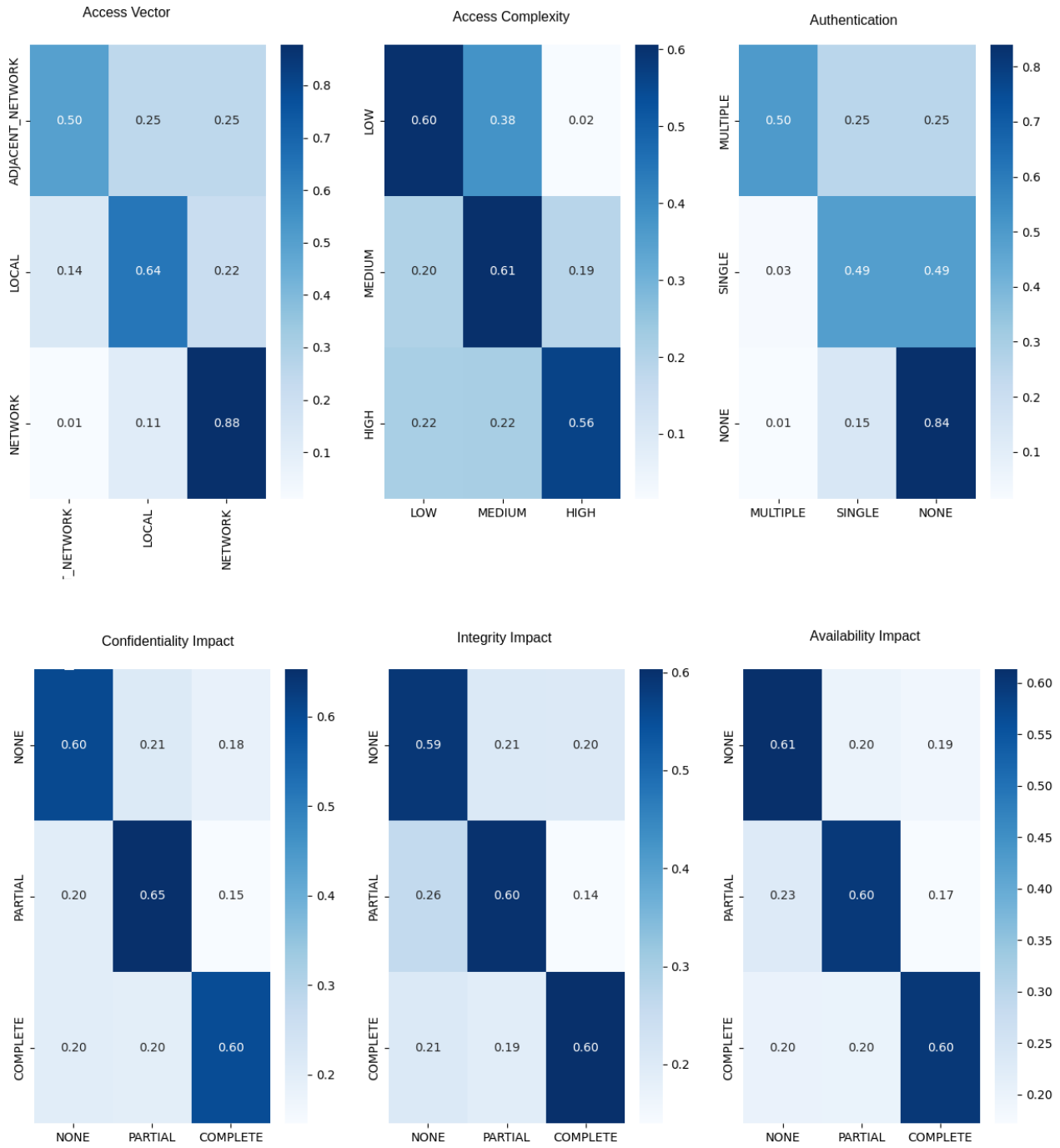

Figure 5: Confusion matrix of estimated accuracy for CVSS metrics for version 2.0 for NVD

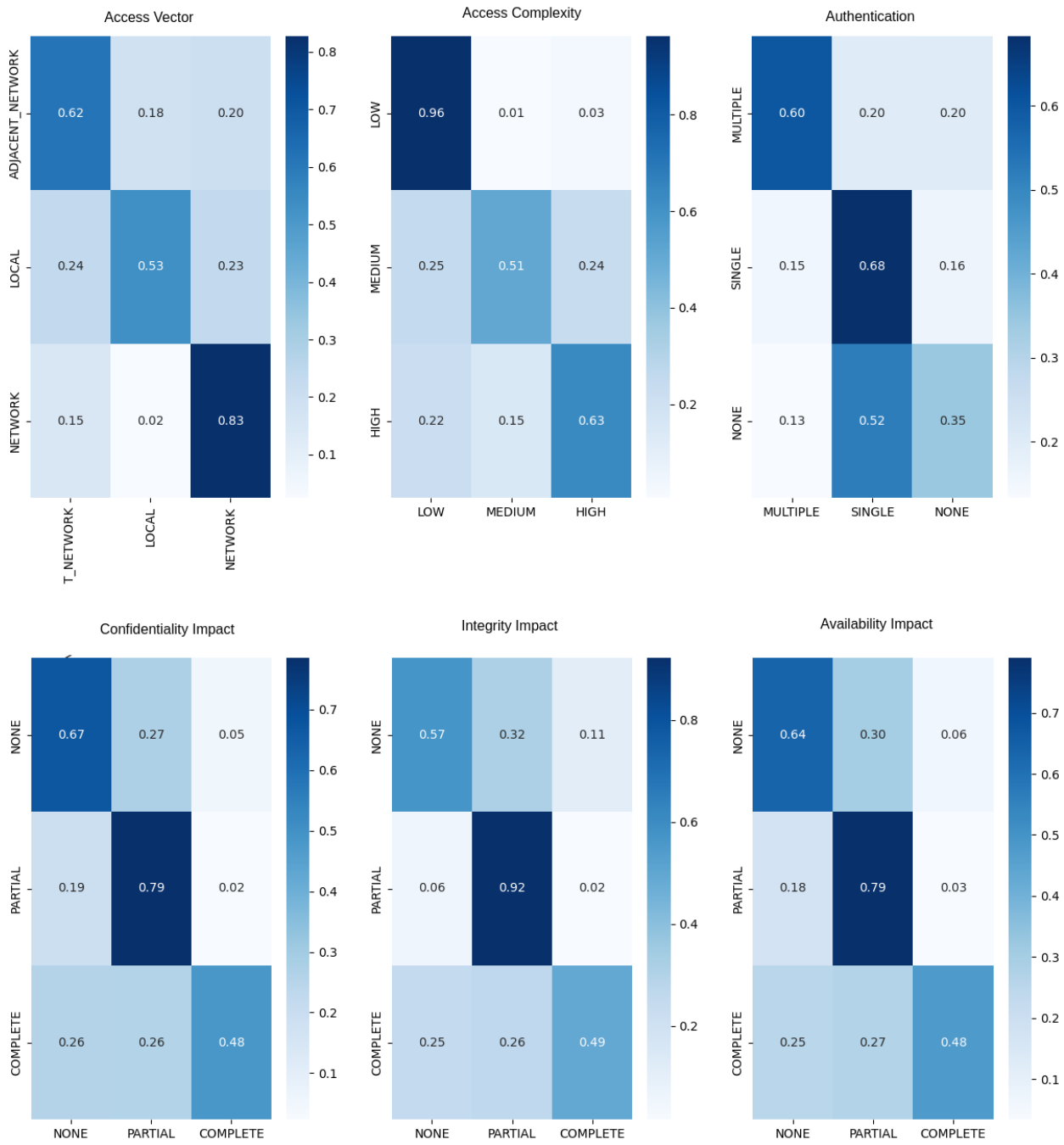

Figure 6: Confusion matrix of estimated accuracy for CVSS metrics for version 2.0 for MITRE

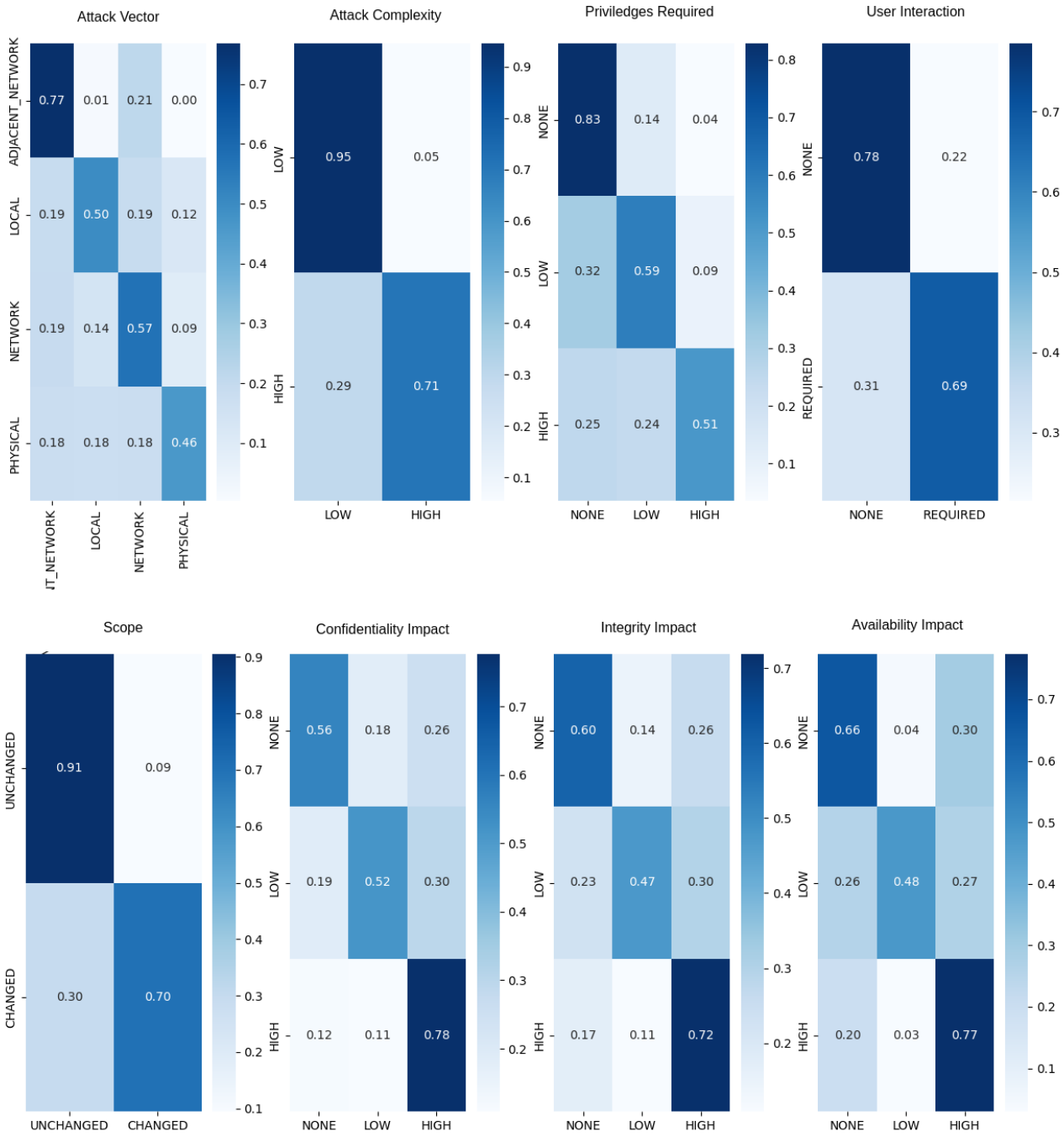

Figure 7: Confusion matrix of estimated accuracy for CVSS metrics for version 3.0 for NVD

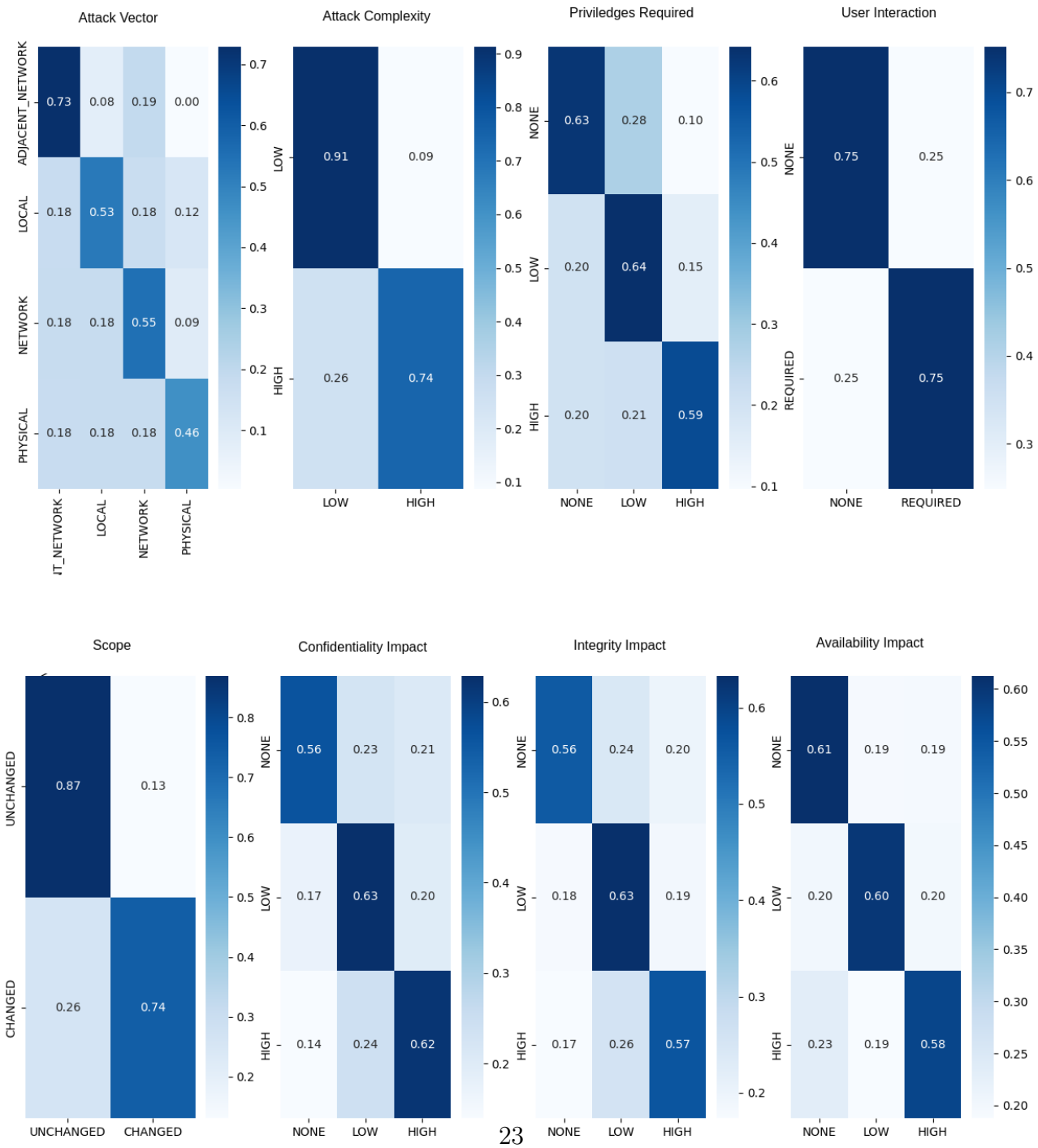

Figure 8: Confusion matrix of estimated accuracy for CVSS metrics for version 3.0 for MITRE