CWE-ASSIST: A framework for automating CWE classification

Ronan Oostveen
University of Twente
P.O. Box 217, 7500 AE Enschede
The Netherlands
r.oostveen@student.utwente.nl

***Abstract*—To effectively secure an organization, it is essential to understand its weakness exposure. Datasets frequently utilized to identify these weaknesses often miss relevant weaknesses in their labels. We found that a multitude of labeling organizations have different labeling practices, where in they only classify a single weakness even if multiple are present. To allow these dataset to be used for cybersecurity purposes adding these missing labels through reclassification is crucial, but doing so manually is impractical. A method that can automatically add these weaknesses labels could address this issue, therefore we propose a novel method to semi-automatically produce these extra labels.**

We do so through a classification model that generates an abundance of relevant weakness labels, in our experiments this abundance of labels already create significant F1-score improvement, however with a lower precision. We then use these generated labels as suggestions for experts to review, generating a new set of expert curated labels. These curated labels then become new dataset labels and are used retraining our classification model thresholds. Our experiments demonstrate that utilizing even small datasets of expert evaluations can lead to a significant precision improvement while maintaining a similar F1-score compared abundance of labels. Our methods both also managed to out perform the current NVD labels according to the expert labels, suggesting that this method can serve as an effective yet low-effort approach for reclassifying weakness labels.

***Index Terms*—Cybersecurity, Weakness Classification, Common Weakness Enumeration (CWE), Dataset Quality, Automated Reclassification**

I. Introduction

In the realm of cybersecurity, managing and understanding vulnerability and weakness exposure is crucial for maintaining the integrity and security of systems. The Common Vulnerabilities and Exposures (CVE)® system serves as a reference for publicly known information-security vulnerabilities and exposures, aiding organizations in identifying and addressing security threats [8]. The Common Weakness Enumeration (CWE) is a similar standard which creates a categorization of software weaknesses which can be derived from CVEs. CWEs provide a structured way to classify vulnerabilities, offering organizations clear insight into the specific weaknesses affecting their systems. [7]

Despite its importance, the process of labeling CWEs in commonly used datasets such as the National Vulnerability Database (NVD) and MITRE remains manual and flawed. The manual process is time-consuming and susceptible to human error, resulting in inconsistencies and inefficiencies in weakness management. That 23.6% of CVEs remain unclassified with any CWE indicates these limitations, this situation is exacerbated by the significant increase in reported CVEs in recent years. Additionally, 9.2% of the CVEs are duplicates, and at times these duplicates are classified with different weaknesses, highlighting possible human error in this process. Additionally, prior studies have raised concerns regarding the validity of NVD labels, indicating that CVEs frequently correlate with multiple CWEs; however, this is rarely observed in the NVD dataset. [2]

The issues present in the current dataset underscore the necessity for reclassification. However, given that the current experts already have difficulty keeping up with the demand for manual labels, manually adding the missing labels to the entire dataset would be even more impractical. Introducing automation into this process may enhance efficiency for experts, allowing them to label more weaknesses more efficiently. Furthermore, automation may enhance labeling efficiency and accuracy, facilitating improved management of weaknesses.

Recent advancements in automated classification methods for CWEs have demonstrated high accuracy rates [1], [16], [14]. However, these methods have not garnered widespread adoption among experts. Accessibility, complexity, and trust in automated outcomes may contribute to this disparity.

We propose a novel technique to address the need for an automated reclassification method. This approach involves training previous state-of-the-art methods to generate a large number of weakness labels by relaxing precision constraints. A group of experts subsequently evaluates the validity of a set of generated labels, resulting in a new set of assessed labels. The assessed labels serve to refine the dataset and, importantly, to fine-tune the classification model. This process improves the model's effectiveness for classifying weaknesses. The evaluation results demonstrate that the model outperforms the existing NVD labels in terms of performance. Consequently, through utilizing a limited set of expert-assessed labels this model can effectively reclassify the weaknesses identified in the NVD.

Automating CWE reclassification with a model that surpasses the accuracy of the NVD can enhance existing datasets, resulting in greater completeness and accuracy. This model can be further refined through additional expert-evaluated labels, which may lead to enhanced performance. With more accurate and comprehensive labels, security specialists accessing these datasets will have a greater understanding of their exposure. It also opens the door to further possibilities, such as including automatic or aided labeling when reviewers are submitting CVEs, which might remove human error and result in better labels on CVEs submitted.

A. Research questions

The challenges associated with automating the reclassification process are complex, necessitating a careful balance among data quality, automation techniques, and manual intervention. Ideally, a solution to this challenge would meet all three criteria by developing a fully automated reclassification technique that is able to use poor quality data and requires minimal manual effort. This paper addresses the following questions.

RQ1 What specific inaccuracies and inconsistencies exist in the NVD manual CWE classifications regarding accuracy and consistency?

RQ2 To what extend can we measure the performance of a classification model in the presence of missing labels in the dataset?

RQ3 To what extend is it possible to recommend relevant weakness labels that can be used in expert assisted labeling?

RQ4 Can subsequently fine-tuning a model using a restricted set of expert-validated labels improve model performance?

II. Background and Related Work

The CVE Program collaborates with partners (vendors, researchers, etc.) around the world as CVE Number Authorities (CNAs) to assign CVE IDs and publish CVE Records for vulnerabilities within their agreed upon scope [10]. When vulnerabilities are first discovered they are then reported to the CVE Program, to request a CVE ID. Once the record is confirmed, through the identification of minimum required data elements, the record is published to the CVE List [9].

Once a CVE is published in the CVE List, it can also be associated with a CWE. CWE defines a ”weakness” in a component (software, firmware, hardware, etc.) that may lead to security implications under specific conditions. NIST and MITRE provide standards and datasets for classifying CWEs; however, the NIST dataset (NVD) is significantly more comprehensive, encompassing over 250,000 classified CVEs. We utilize this NVD dataset for our experiments because of its completeness.

The National Vulnerability Dataset (NVD) enhances the Common Vulnerabilities and Exposures (CVE) List by providing additional information on published CVEs. NVD staff are responsible for enhancing CVEs by compiling data points from the description, provided references, and any publicly available supplemental information regarding a CVE at the time. This enrichment process involves classifiers based on the Common Weakness Enumeration (CWE), Common Platform Enumeration (CPE), and Common Vulnerability Scoring System (CVSS). Our research utilizes CVE descriptions as input for classification purposes, and the CWE classifications from the NVD database serve as the training labels.

Since weaknesses can be defined at many abstraction levels, CWEs are organized in a hierarchical structure that allows for several levels of abstraction. The CWEs at higher levels provide a broader view of a vulnerability, whilst the CWEs at lower levels provide finer granularity and greater specificity. An example of this is the abstract Class ”CWE-330: Use of Insufficiently Random Values” which is connected with the more specific Base CWE ”CWE-338: Use of Weak PRNG”. The defined abstraction levels for these hierarchical classes are pillar, class, base, and variant. Defined as follows:

- **Pillar:** is theme for all class/base/variant weaknesses related to it.
- **Class:** is a weakness also described in a very abstract fashion, typically independent of any specific technology.
- **Base:** is a weakness described in terms of 2 or 3 of the following dimensions: behavior, property, technology, language, and resource
- **Variant:** is more specific than a base and is linked to a product, typically involving a specific language or technology

NVD analysts classify CVE vulnerabilities by utilizing a standardized selection of both broad-grained and fine-grained CWEs. The NVD currently utilizes CWE-1003, which comprises 130 unique CWEs, enabling analysts to classify the majority of CVEs. If the expert is unable to classify a specific CWE due to the weakness not aligning with the CWE-1003 standard or insufficient information in the CVE, they utilize identifiers such as NVD-CWE-noinfo.[1]. [13], [12]

A. CWE classification

In past years, researchers have thoroughly investigated the classification of CVEs into CWE categories, resulting in notable advancements in accuracy.

The initial method examined is ThreatZoom [1], which employs a hierarchical framework to identify vulnerabilities. This approach distinguishes between coarse-grained and fine-grained weaknesses based on whether the identified CWE is classified as a Base class or a higher class, with the former representing fine-grained and the latter coarse-grained vulnerabilities. Aghaei et al. attained a classification accuracy of 92% for 116 CWEs using this method.

Wang et al. [16] proposed an alternative classification approach utilizing a BERT model, achieving an accuracy of 90.74% while categorizing only the 10 most prevalent CWEs. And another promising method was proposed by Pan et al. [14] which integrates a bidirectional Gated Recurrent Unit (Bi-GRU) with a Text Convolutional Neural Network (TextCNN), achieving an accuracy of 90.01% on a dataset containing 158 Common Weakness Enumerations (CWEs).

It is difficult to accurately compare the performance of different approaches because they classify varying amounts of CWE classes. To address this disparity, we choose to utilize CWE-1003 standard, which is also the latest used NVD labeling standard. However, another issue with current research approaches is that they do not address CWE classification as a multi-label classification problem, as they categorize only one weakness per vulnerability.

However, it is critical to address the fact that vulnerabilities can typically exploit more than one weaknesses. In the NVD dataset this is also partially represented, where around 5% of CVEs are identified with multiple CWEs. However, Aota et al. [2] found in their research that still many CVEs labeled with a single CWE could be labeled with multiple CWEs instead, suggesting that the NVD dataset may be incomplete. They also continue to state the importance of this since correctly labeling

[1] Example of a CVE with the ’NVD-CWE-noinfo’ CWE classifier: https://nvd.nist.gov/vuln/detail/CVE-2023-5038

a CVE with multiple CWEs can highlight important security considerations for systems[2].

> Unspecified vulnerability in Microsoft Excel 2000 SP3 through 2003 SP2, Viewer 2003, and Office for Mac 2004 allows user-assisted remote attackers to **execute arbitrary code** via crafted Style records that **trigger memory corruption**.

Figure 1. CVE-2008-0114 Description with weakness related text highlighted

In their paper Aota et al. also highlighted this finding with the Positive Unlabeled (PU) learning multi-label classification model proposed. Figure 1 illustrates an example they had given of CVE-2008-0114 which was initially categorized in the NVD as CWE-94 (Code Injection). However, their model also classified next to CWE-94 also CWE-119 (Improper Memory Buffer Restriction), which appears to be appropriate in this scenario. For this case if a security expert only considered the single-label of CWE-94, they would be missing important context.

Recently, in another article we have fine-tuned a large language 'T5' model on CWE using both PU learning and Contrastive learning, introducing the CWE-GEM models. In this research we where able to compare these models in similar environments to the previous state-of-the-art BiGRU model, and found noticeable performance improvements. However, this research had significant limitations in regards to multi-label performance legitimacy since it was difficult to define a proper prediction thresholds using the limited amount of multi-label samples in the dataset. Which severely affects the legitimacy and applicability of the model. In this research we propose a solution for this limitation by purposefully generating to many weakness labels, and determine new thresholds using expert evaluation.

III. Preliminary Research

In order to better grasp the existing issues in the NVD and the current limitations of the manual CWE labels, we first perform a preliminary analysis on the dataset's manual labels. We expect that the NVD dataset may be of low quality and want to explore to what extent this dataset contains errors or inconsistencies. By performing a preliminary analysis we intend to measure the quality of this dataset, which allows us to reason about how it can be used in a (semi-)automatic labeling pipeline and how we can ensure that the proposed approach and subsequent evaluation yield trustworthy results. To this end, we perform 3 basic experiments:

- We examine duplicate samples in the dataset
- We compare labeling behaviours of different CVE Naming Authorities (CNAs)
- We examine inaccuracies with NVD unique labels (NVD-CWE-noinfo, NVD-CWE-Other)

A. Dataset

We utilize the NVD dataset as our dataset. The NVD is a repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, product names, and impact metrics. The NVD provides information of vulnerabilities, but also assigns a CVSS and identifies related CWEs. The Common Vulnerability Scoring System (CVSS) is a method used to supply a qualitative measure of severity, and CWE provides a common language of discourse for discussing, finding and dealing with the causes of software security vulnerabilities as they are found in code, design, or system architecture.

A typical CVE entry in the NVD consist of the following components: CVE Identifier, Status, Description, Additional References, Known Affected Software Configurations, CVSS, and CWE classification.

Through the NVD API we obtained 240.000 CVEs on August 28, 2024. The amount of CVE in the NVD has increased dramatically in recent years, shown in Figure 2.

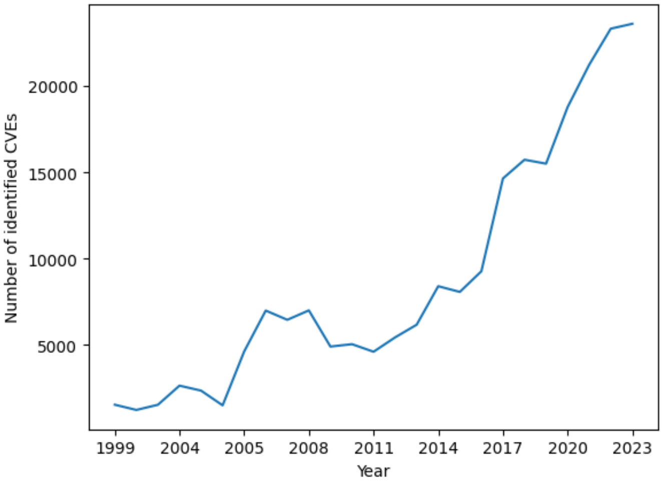


Figure 2. NVD classified CVEs by year

Status	CVE Amount
Analyzed	129.398
Modified	95.131
Rejected	13.999
Awaiting Analysis	1.455
Undergoing Analysis	17

Table I
CVE STATUSES

Each CVE in the NVD has an associated status, as shown in Table I. This status indicates the stage of the CVE within the NVD life-cycle[2]. We take this into consideration for our pre-processing, through filtering out all Rejected CVEs. These CVEs have been rejected in the CVE list, and therefore would also not be relevant for our research. We still consider CVEs from other parts of the life-cycle, such as 'Modified' or 'Awaiting Analysis', because if these CVEs have an associated CWE, they still contain useful information for training.

Additionally we identify another 1.961 CVEs without any CWE label, this lack of labels highlights an inconsistency in the dataset. After removing these we are left with 224.040 CVEs.

[2] https://nvd.nist.gov/vuln/vulnerability-status

The remaining CVEs on average have 1.16 CWE labels, with the distribution shown in Table II.

#CWE	CVE Amount
1	190.759
2	31.834
3	1.154
4	266
5	22
6	4
7	1

Table II
CWE PER CVE DISTRIBUTION

B. Duplicate CVE descriptions

Evaluating the CVE description we find that 4.18% of the CVE descriptions in our dataset are duplicates. Further, within these duplicates 26.70% have different CWE labels compared to their duplicates. This shows a serious problem in the dataset, the existence of samples with the exact same information but a different classification. This could be caused by many different reasons, but in our case studies we identify two particular reasons.

The first reason is the use of a generic CVE description for different vulnerabilities involving different weaknesses (Appendix C). This CVE is then classified by the CNA and NVD using extra external information, outside of the NVD. Or the CVE could be classified as 'NVD-CWE-noinfo', since the CVE does not clearly specify any weakness information.

The second reason is generating multiple CVE entries for labeling multiple weaknesses (Appendix B). In this case a CNA generates multiple CVE entries for one vulnerability, each entry is then only identified with one of the weaknesses exploited. This CNA does not consider that this could also have been achieved through one CVE entry with multiple CWE labels.

The existence of these duplicates and the potential causes which are discussed in the case studies, highlight a point of contention in the NVD dataset. The fact that the same CVE description is associated with multiple unique vulnerabilities, ads a lot of noise to the data that can make it difficult for an automated method to classify, or even worse make it more fault-prone for an expert to use, this highlights the need for further automation in the generating and classification of CVEs so that these human errors can be prevented.

C. CNA comparison

The NVD dataset combines the CVEs and labels from many different CVE Naming Authorities (CNAs). We explore the differences in labeling from these different CNAs to highlight another dimension of inconsistencies with the data, again highlighting the need for automation and standardization. The dataset contains CVEs from 309 unique CNAs, but to keep good oversight in our comparison we only look at 10 CNAs that labeled the most CVEs. In Table III, we first show the number of CVEs that CNA has published, to be able to highlight the distribution of CVEs. Secondly, the percentage of duplicate descriptions within the CNAs CVE list. After this, the number of distinct CWEs utilized on labeling the CNAs CVEs, this can be done by the CNA itself or other entities such as NVD. Then we show the percentage of CVEs with multiple CWEs, this is to show how many of the CNAs CVEs are multi-label. And finally we show the 3 most common CWEs labeled on the CNAs CVEs, this is to show the distribution within the weaknesses.

The first thing to observe is that the vast majority of CVEs (101.512) is are classified by MITRE, followed by Redhat which has identified 10.173. This is also understandable given that MITRE created the CVE database.

The duplicate description percentage is insignificantly low for most of the CNAs. However, this is not the case for Adobe and Microsoft where 32.1% and 28.4% of their CVE descriptions are duplicates, respectively. The variation in the number of duplicates can be explained by the organizations' use of different labeling practices. An explanation for this can be that Microsoft and Adobe tend to use more generalized descriptions for their CVEs, as discussed in Section III-B.

The number of distinct CWEs used by the organisations varies significantly, ranging from 51 to 326. This large difference in CWEs is important to highlight for our method since this difference is another inconsistency between CNAs. This does not necessarily imply that the labeling only evaluated a few CWE; it is also possible that certain organizations CVE leveraged a wider range of weaknesses than others.

In the multi-label column in the Table, you can again see that there are some outliers. For MITRE, Oracle, and Apple all have less than 1.6% of their CVEs being associated with multiple CWEs. While for GitHub and Cisco 55.2% and 53.3% of their CVEs respectively are multi-label. This can suggest that their vulnerabilities are more complicated and involve combining multiple CWEs, or that some organisations consider multiple CWE together less with their CVEs, or more likely a combination of multiple reasons.

Finally, we identify the top three CWEs utilized on the CVE, which, like the preceding two factors, vary greatly from vendor to vendor. One thing to keep in mind is that, in most of these circumstances, NIST marks its CVE in addition to the vendor's. So, in some circumstances, NIST can add the label NVD-

CNA	#CVE	Dupl.	#CWE	Multi.	Top 3 CWE		
MITRE	101.512	0.4%	208	1.6%	CWE-Other (23%)	CWE-79 (13%)	CWE-89 (7%)
Redhat	10.173	0.1%	282	24.4%	CWE-79 (9%)	CWE-20 (9%)	CWE-264 (8%)
Microsoft	9.312	28.4%	136	11.4%	CWE-noinfo (44%)	CWE-119 (11%)	CWE-200 (5%)
Oracle	7.611	12.4%	51	0.2%	CWE-noinfo (95%)	CWE-200 (1%)	CWE-284 (1%)
IBM	5.571	0.6%	129	6.4%	CWE-79 (22%)	CWE-noinfo (13%)	CWE-200 (11%)
Cisco	5.442	10.5%	233	53.3%	CWE-20 (24%)	CWE-79 (17%)	CWE-399 (9%)
Apple	5.590	16.8%	93	1.4%	CWE-119 (25%)	CWE-noinfo (20%)	CWE-787 (8%)
Adobe	5.222	32.1%	92	10.1%	CWE-119 (18%)	CWE-125 (17%)	CWE-787 (15%)
Github	4.961	0.3%	326	55.2%	CWE-79 (17%)	CWE-200 (6%)	CWE-22 (6%)
Android	4.468	0.4%	96	6.8%	CWE-noinfo (14%)	CWE-787 (14%)	CWE-125 (12%)

Table III
COMPARISON OF CNA LABELING (DUPL. = DUPLICATES & MULTI. = MULTI-LABEL)

CWE-Other when it exploits a weakness that MITRE does not use for classification, or it can use NVD-CWE-noinfo when no direct information is available on which CWE is exploited. Oracle and Microsoft have a high percentage of NVD-CWE-noinfo labels (95% and 44%), which may indicate that their descriptions lack clarity on the weakness exploited.

These differences between CNAs show that the type of vulnerabilities and weaknesses can vary significantly. Some CNAs provide a lot of CVEs which according to NVD often contain too little information to classify a CWE, such as Oracle with 95% of their CVEs having 'CWE-noinfo'. While with other CNAs significant portions of their CVE descriptions are duplicates, such as Oracle 32.1%. This signifies a high amount of human error in the data from these CNAs, which makes their data more questionable. However, other CNAs show indications of more high quality data. In our case Redhat has a low percentage of duplicates, high percentage of multi-label data, and a more evenly distributed top-3 CWEs. These indicators show good quality data since duplicates are a sign of human error, multi-label data is realistic for complex CVEs, and that an even distribution of weaknesses also shows a good variance of types of weaknesses exploited within the vulnerabilities.

D. Incorrect NIST labels

Even though in the entire dataset NIST assigns labels to CVE as well. It is important to note that this is not without its flaws. We have already discussed the 2 unique labels created by NIST 'NVD-CWE-Other' and 'NVD-CWE-noinfo'. 'NVD-CWE-Other' specifies the case where the CVE exploits a weakness that is not within current labels NVD uses (CWE-1003 standard). 'NVD-CWE-noinfo' is the case where the CVE does not give any information about the weakness exploited. However there are 3 interesting cases of things that can happen here:

- NIST identifies NVD-CWE-noinfo, but a CNA actually does label a CWE
- NIST identifies NVD-CWE-Other, but a CNA actually does label a CWE from the CWE-1003 Standard.
- NIST identifies NVD-CWE-noinfo or NVD-CWE-Other, but NVD also classifies another CWE.

For the first case in our dataset we measured 2.492 occurrences, which is 9.1% of all NVD-CWE-noinfo labels (27.457). This means that it happens that the CNA can distinguish a CWE while NVD cannot. This does not mean that this CWE is actually specified in the description or actually happens, since this data can still also be mislabeled.

For the second case in our dataset we measured 1.612 occurrences, which is 5.6% of all NVD-CWE-Other labels (28.809). This could mean that it still often can be labeled as another CWE outside the scope of CWE-1003 or the NVD labels. However, we also still found 335 cases where the CNA labeled it with a CWE-1003 standard label while NVD labeled it as NVD-CWE-Other. Which is interesting to consider since this should not be the case.

Finally for the third case we measured 635 occurrences where NVD labeled multiple labels next to either NVD-CWE-noinfo or NVD-CWE-Other. This also highlights the fact that NVD in their own labels is rather inconsistent as well.

This shows that there are often places where NIST disagrees with others labels, these labels can either be identified by other CNAs or by NIST itself. This means that even though NIST applies the label of 'noinfo' or 'Other' that it still could be possible to identify a CWE. For each of these cases, we further discuss case studies in detail in the Appendix C. And this shows that NIST labels by itself can also be questioned.

E. Proposed considerations

The highlighted case studies show that there are clear problems with the consistency and clarity of both weaknesses and vulnerabilities in the NVD dataset. For this reason there are certain considerations that need to be made with the data to ensure best results with training but also develop a method to create the possibility to improve and re-label the NVD in the future.

Firstly, we discuss what consideration we have taken for the data based on our results. To manage the duplicates in the dataset we have decided to combine all the duplicates together into one CVE with the CWE labels associated with all duplicates. This is useful considering the second case discussed in Section III-B, where one CVE can have multiple entries with different CWE labels. However, it does not work for the case of a generic CVE description used for different vulnerabilities. Therefore, we want to ad the measure that if a CVE contains either 'NVD-CWE-noinfo' or 'NVD-CWE-Other' label that we discard the CVE from our dataset. This is also expected to mitigate the cases of generic CVE descriptions, since these are more likely to include any of these generic labels. Secondly, this also deals with the edge cases where a there are other labels next to 'NVD-CWE-noinfo' or 'NVD-CWE-Other', since these are discarded as well. This improves the quality of the dataset by removing the cases of conflicting labels.

IV. Methodology

To address the challenges of automating the reclassification process of missing weakness labels, as specified in the Research Questions (Section I-A), we have developed a model pipeline containing 4 steps shown in Figure 3.

- The first step shown in the pipeline is the data preprocessing, this is done using the considerations given after the preliminary experiments in Section III-E, where identify the issues with the current manual NVD labels (RQ1).
- The second step is training the models using PU learning methods, which will aid the models for identifying relevant labels as missing label recommendations.
- The third step is computing the classification thresholds for the models, this is done using the FBR-algorithm which will be introduced in Section IV-C. This is essential to be able to recommend relevant weakness labels for the expert assisted labeling (RQ3).
- The fourth step is the expert evaluation, here the generated labels will get manually evaluated for validity, which will result in new expert curated labels. These expert curated labels are used for measuring the metrics and performance of models (RQ2). Additionally, these expert labels are used as new training data for calculating the thresholds in the third step, resulting in new expert curated thresholds.

Allowing us to measure whether this will improve the model performance (RQ4)

It is important to note that the Fine-tuning can be done over many iterations, potentially iteratively improving the performance. However, we only did one iteration of this cycle since we were limited in the amount of expert evaluation. We will now elaborate on each step in more detail.

A. Pre-processing

In the pre-processing the goal is to prepare the data appropriately so that we can get the best performance out of the model for the reclassification of the labels. For this we first consider the recommendations that resulted from the preliminary research. In this we found that the presence of duplicates should be considered and to mitigate this we can combine the duplicate data points together into one sample, combining the labels together too. The second consideration is for the NIST created quality weakness labels of ’NVD-CWE-noinfo’ and ’NVD-CWE-Other’, these labels are given to weaknesses if the vulnerability does not contain enough information or if the weakness is outside the scope of NIST. For this the consideration is that if a sample is labeled with any of these weaknesses that the sample will not be used for training. This is to ensure data quality since according to NIST these samples are invalid for labeling with an actual weakness.

Next in the back it was established that previous methods classified an inconsistent amount of weaknesses. To solve this we will use the already discussed standard of CWE-1003, encompassing 130 CWEs at two granularity levels. This standard was chosen to ensure broad applicability across various scenarios and is also adopted by the NVD itself, where 98% of CVEs are classified using these CWEs.

The labeled datasets created was done like in the preliminary research using the CVE API from NVD[3], containing approximately 240k CVE records. Then as stated we merge the duplicate entries and filtered out the samples with ’NVD-CWE-noinfo’ and ’NVD-CWE-Other’ labels. Then all other labels from weaknesses outside the CWE-1003 standard were filtered out, resulting in the dataset composition given in Table IV.

B. Model training

For our method we trained 3 models: TextCNN[5], [14], BiGRU-TextCNN[3], [14], and CWE-GEM[11], [15]. These models have shown state-of-the-art performance on CWE classification before. The inclusion of multiple models allows us to compare performance in between the models and select the best performing model for generating the eventual relevant weakness labels.

Table IV
DATASET COMPOSITION

# classes	129*
# CVEs	144,928
# CWEs	151,557 (4.6%)
Train-test split	90% - 10%

* Excluded CWE-920, since it only occurs 3 times.

These three models all operate in two stages, first the feature extraction and secondly through a fully connected layer for the classification. This is a modification form the original CWE-GEM model, which originally classified samples based on cosine similarity between the sample and weaknesses. We made this modification since in the initial testing the performance was slightly better using a fully connected layer as classification method. The training of the CWE-GEM feature extractor was done using a Positive Unlabeled version of the Binary Cross Entropy (BCE) loss, which is different than the fully connected layer. This is because the implementation of the CWE-GEM was made using the Instructor framework which only supports a BCE loss.

1) PU learning: Aota et al. [2] and the CWE-GEM method have shown using Positive Unlabeled learning outperforms other previous methods on CWE classification. In our research we also utilize PU learning for its potential to find missing labels. For this we utilize the ’Hill’ loss introduced by Zang et al [17], which alleviates the effect of missing labels through a robust loss that is insensitive to false negatives. In their research the ’Hill’ loss outperformed other losses specifically at the task of being able to correctly label missing labels. This is also the reason we utilize it for our method on the fully connected layers, BiGRU, and Text-CNN models. As this loss could potentially also recommend missing relevant weaknesses labels, which can then be classified by the experts as a new curated label.

C. FBR-algorithm

For the multi-label classification to be able to determine the rejection region, we need to optimize the probability threshold. To achieve this the FBR (F-beta Ratio) algorithm[4], [6] can be utilized, due to its capability to optimize thresholds for every label independently. The original FBR-algorithm is quite simple where it calculates an optimal threshold for every class over a k-fold using the F-Beta measure, in our case we used a stratified 5 k-fold split. The F-Beta measure here is a variant of the F-score where another Beta variable is

[3] NVD API: accessed on 28th of August 2024

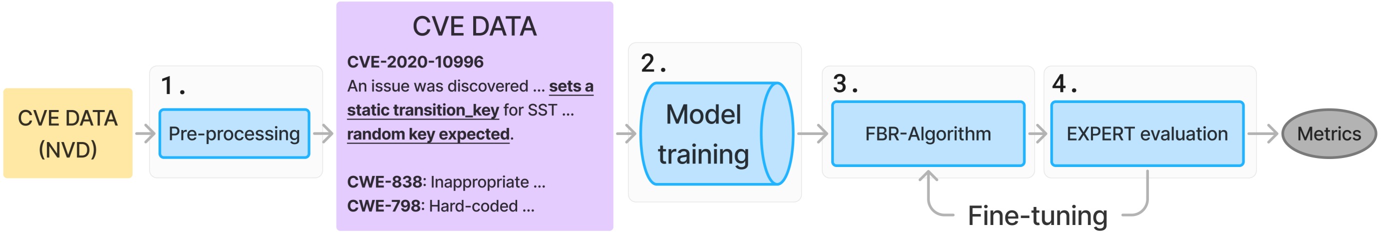


Figure 3. The pipeline of our methodology

added. This Beta variable enables us to adjust the balance between recall and precision, making it particularly suitable for assisted labeling tasks while also accommodating the needs of automated processes.

It was found in further research that the original FBR algorithm (SCut), while being a reasonable approach can easily over-fit or uncommon labels. This is important to note for our dataset since the CWE labels are unbalanced with a long-tail distribution, and some classes are very rare. For this Yu-Len Jin and Chih-Jen Lin, found that through smoothing the F-measure their algorithm had significant improved handling underrepresented labels, ensuring that less common labels in the unbalanced data are not disadvantaged. [6]

In our initial experiments, we confirmed the findings of Yu-Len Jin and Chih-Jen Lin, with the improved FBR-heuristic having a noticeable improvement on the macro performance of our models. Therefore, we chose to use this heuristic for our initial FBR threshold tuning. The FBR-algorithm aids in recommending relevant weakness labels, since restricting the precision and increasing the recall will allow for the classification of more weaknesses. During the expert evaluation we will be able to determine the relevancy of these labels, by whether they get classified by the experts.

D. EXPERT evaluation

After using the FBR algorithm to increase the classification threshold and hence increasing the amount of weakness labels, we need to find the relevant labels. For this we select a subset of CVEs to be labeled by expert reviewers, this will establish a new ground-truth through the majority vote of the expert. Finally this new ground-truth is used to measure the performance of the classification model and the original NVD labels. Here we will observe both the weighted and the macro of the F1-score, precision, and recall metrics. These metrics will give a full overview of the model performance with the weighted results showing the overall performance, and the macro results showing the per class performance, which will allow for a comprehensive evaluation of the results.

E. Fine-tuning

The expert curated labels will also be used as new training data for the FBR-algorithm, to generate a new of more curated thresholds. These expert fine-tuned thresholds will then again be evaluated by experts on another set of CVEs. This subset of CVEs will contains different CVEs having the same distribution of CWEs, and will be evaluated by a different group of experts. Finally, we compare the performance metrics of the expert fine-tuned thresholds against the original thresholds and NVD labels, to see if the expert fine-tuning impact the performance.

To give a better understanding of the fine-tuning process we can look at the example explaining this process, given in Figure 4. In the visualization you can see an weaknesses labeled for CVE-2007-3008, which is described in Figure 5. This CVE is classified by NVD as CWE-79: Cross-site Scripting and CWE-200: Exposure of Sensitive Information to an Unauthorized Actor. The CWE-ASSIST model with $\beta = 1$ has a stricter threshold and only classifies CWE-200. While, the CWE-ASSIST model with $\beta = 4$ would identify CWE-79, CWE-200, CWE-203 as Observable Discrepancy and CWE-20 as Improper Input Validation. Finally, an expert associated this CVE with CWE-79, CWE-200, and CWE-749: Exposed Dangerous Method or Function. This demonstrates that both NVD and CWE-ASSIST identified CWEs incorrectly, and as a result, we alter the CWE-ASSIST model's thresholds based on these expert labels.

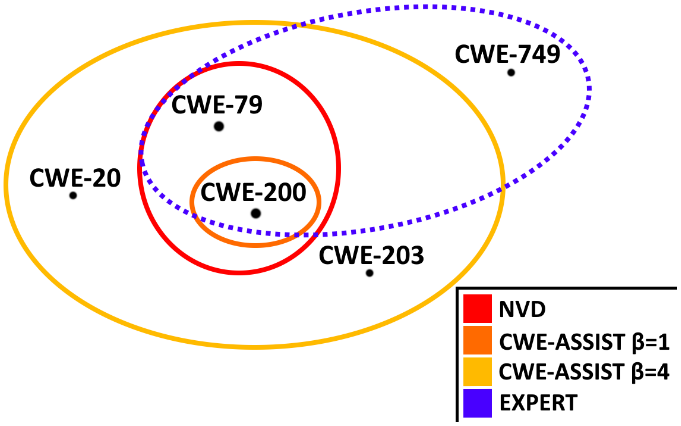


Figure 4. Example of CVE-2007-3008 thresholds

For the expert fine-tuning we have a considerable limitation, which is the amount of CVEs we can have labeled by the human experts. The test-set alone consists of more than 11.000 samples and a 129 classes. To keep the amount of CVEs manageable for the experts the subset contains 125 CVEs, which maintains the same distribution of classes as the entire test-set, since the dataset is unbalanced this only includes 73 CWE classes. The distribution of CWE classes is displayed as the Train set in the Figure 6, and in more detail in the Appendix. For the fine-tuning of the thresholds we found that for such a small set of samples that the smoothed F-measure decreased the performance and made the model more unpredictable. Therefore we chose to use the original FBR algorithm for the expert-label FBR threshold tuning. To evaluate the expert fine-tuned thresholds, we select another subset of 125 CVEs with the same distribution of classes which can be seen as the Test set in the figure.

V. EVALUATION

To validate our method, we compare the F1-score, recall, and precision of several approaches. The metrics chosen are used to address performance in terms of accuracy and consistency, as well as to investigate the influence of the FBR algorithm on enhancing performance, as defined in RQ3. We then evaluate the framework's performance against the current NVD labels, using expert labels as new ground truth to address RQ2.

> Mbedthis AppWeb before 2.2.2 enables the HTTP TRACE method, which has unspecified impact probably **related to remote information leaks and cross-site tracing (XST) attacks**, a related issue to CVE-2004-2320 and CVE-2005-3398.

Figure 5. CVE-2007-3008 Description with weakness-related text highlighted

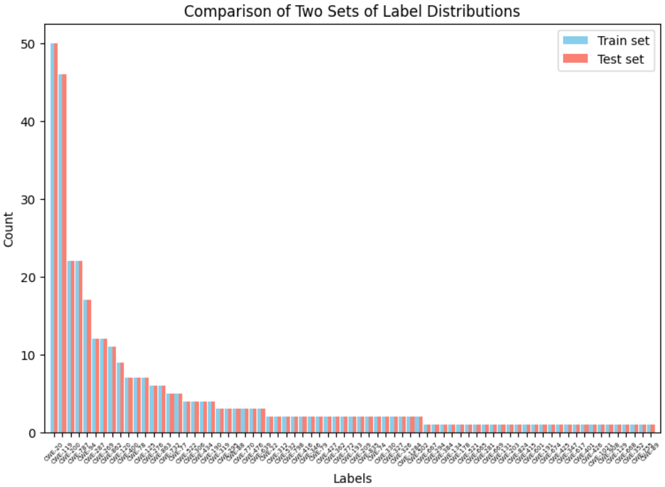


Figure 6. Expert fine-tuning dataset CWE distribution

Finally, we fine-tune the thresholds based on the expert labels and re-evaluate the model to determine whether this improves performance, which addresses RQ4.

A. Experimental Setup

In our experiments, we compare the model's multi-label recall and precision. We compare the TextCNN, BiGRU, TextCNN-BiGRU, and CWE-GEM models. After the training, we use the enhanced FBR algorithm to optimize the models' thresholds on a training dataset. For the FBR optimization, we experiment with two Beta values, 1 and 4, to examine if it is indeed possible to recommend more relevant weaknesses to solve the missing label problem. This is followed by expert evaluation on a test dataset, further fine-tuning of the thresholds, and another expert evaluation to observe any performance changes.

B. Experiment Results

The results of our experiments are summarized in Table V, where we compare the performance of different models across two Beta values (1 and 4). The findings are presented using both weighted and macro-averaged precision and recall to provide a balanced picture of model performance across frequent and infrequent classes.

The effect of the Beta parameter in the FBR method is visible in all models. Increasing Beta from 1 to 4 improves recall but reduces precision. For TextCNN this is seen with Beta=4 where the weighted recall is 84.73% and precision is 58.41%, with Beta=1 these are 59.24% and 66.58% respectfully. This demonstrates that a larger Beta, which focuses more on recall, helps capture more CWEs but also introduces more false positives. Similar patterns can be seen for the BiGRU-TextCNN model, a Beta of 4 results in a macro recall of 57.78% and precision of 41.94%, with Beta=1 these are 48.12% and 51.38% respectfully. This stresses the importance of correctly choosing the appropriate Beta value for a certain use case. For instanced with fully automated classification you would ideally require both recall and precision to be high, here a Beta=1 would make sense. However, for assisted labeling which is more similar to our expert you would not mind a lower precision if this results in a higher recall, in this case a higher Beta of for instance 4 could be used.

Finally, the performance of the CWE-GEM model appears to fluctuate more considerably between the two Betas. With a Beta=1, the weighted precision is 82.95% while the recall is only 67.02%, for the macro these are 62.45% and 35.78% in comparison. In both these cases the precision is significantly higher than the other models, while the recall performs similar if not worse. This completely changes for the Beta=4, where the model does manage to outperform on weighted recall but scores very similarly on all other metrics. Which actually shows that for the Beta=4, the differences in performance are actually less. However, since CWE-GEM model consistently outperforms the other models in the metrics, is this model also the one used for expert evaluation and fine-tuning.

C. Expert evaluation

To continue with evaluation and fine-tuning or model we first need to establish a new ground-truth. As discussed in Section IV-D, this is achieved by making a group of experts manually evaluate the generated labels on 125 CVEs, here at least 3 experts label each CVE and weaknesses that got a vote from majority of the experts is established as new ground-truth. These labels are all unique labels from the NVD and CWE-GEM at beta=1 and beta=4. For this 3 experts to classified a sub-set of 125 vulnerabilities with the selected CWEs from both CWE-ASSIST and NVD. Finally we compare the performance of the NVD predictions and CWE-ASSIST performance on the new ground-truth, depicted in Table VI.

Table VI
EXPERT EVALUATION (R = RECALL; P = PRECISION)

		Weighted			Macro		
Model	β	F1	R	P	F1	R	P
NVD labels		48.52	43.06	**66.30**	37.66	36.27	**43.48**
CWE-GEM	1	28.95	27.78	36.64	24.08	23.73	26.31
	4	**69.70**	**90.97**	60.53	**47.85**	**59.55**	43.28

The first thing to note in the evaluation, is the performance of the NVD labels. While NVD manages to get the highest precision, it achieves a low recall with a score 42.41% and 37.40%, on weighted and macro recall respectively. This shows that according to the experts a significant amount of the labels are missing from the NVD. For the CWE-GEM method, the

Table V
EXPERIMENT RESULTS (R = RECALL; P = PRECISION)

	$\beta = 1$				$\beta = 4$			
	Weighted		Macro		Weighted		Macro	
Model	R	P	R	P	R	P	R	P
TextCNN	59.24	66.58	40.37	49.61	84.73	58.41	59.86	40.94
BiGRU-TextCNN	66.92	64.90	**48.12**	51.38	83.69	59.33	57.78	**41.94**
CWE-GEM	**67.02**	**82.95**	35.78	**62.45**	**88.57**	**59.87**	**59.89**	41.31

differences in the performance is again quite different for the different Beta's. The Beta=1 model has by far the worst performance, with none of the metrics higher than 37% the performance is way worse of what would be expected based on the performance on the NVD labels. This also clear indication that only evaluating the models against the NVD labels does not show the full picture. For the CWE-GEM Beta=4 model it manages to excel on recall, with it being 90.97% and 59.55% on weighted and macro averages respectively. This is to be expected since this model significantly over-labels the data, providing most of the labels seen by the experts. However, while the precision is worse than NVD it still manages to improve the precision of the Beta=1 version. This highlights how badly missing labels can effect model performance. In this case over-labeling clearly improved the performance according to the expert labels.

Even when observing the F1-scores of these models the CWE-GEM Beta=4 model scores significantly higher with a 20% weighted F1 improvement and a 10% macro F1 improvement over the NVD. This already shows that according to the generally accepted F1-metric our over-labeling model is better than the NVD. However, this does not show the fact that the precision is worse than the NVD.

D. Expert fine-tuning

Using the expert curated label from the evaluation for fine-tuning, can show whether the expert feedback can help improve model performance. For this we used the FBR-algorithm without smoothing, since with only 125 samples the smoothing seemed to worsen performance. This new CWE-GEM expert fine-tuned model is then also evaluated against the other models, by manual labeling of another group of experts on different CVEs, with the results depicted in Table VII.

Table VII
FINE-TUNED EVALUATION (R = RECALL; P = PRECISION)

		Weighted			Macro		
Model	β	F1	R	P	F1	R	P
NVD labels		57.57	50.00	**76.49**	44.64	42.86	**51.71**
Original	1	33.63	30.90	42.82	21.98	20.91	25.61
Expert	1	62.31	66.29	75.13	48.09	54.86	47.13
Original	4	**73.87**	**91.57**	64.14	**49.62**	57.33	45.83
Expert	4	72.89	84.83	68.28	48.85	**57.53**	45.99

This table we can observe that the original models from the first evaluation vary up to 10% on the metrics in this evaluation. This variance could be from our evaluation sets being too small, or that the new group of experts has a significant different opinion than the previous group. However, for Beta=1 the expert model shows a more consolidated performance, with both the precision and recall improving significantly from the model before, showing a very balanced model which is competitive to the NVD on both recall and precision and even outperforming on F1-score. However, the expert Beta=4 model very similar performance to the original Beta=4 model. For this model, the precision was marginally higher for a slight drop in recall while keeping the F1 score very similar. These results show that while the expert evaluation can result in proper model for Beta=1, that it does not improve the performance noticeably of the over-labeled model with Beta=4.

1) Label evaluation: Looking into the assigned of labels by model can also more insights into whether the different methods generate more labels, shown in Figure 7. The blue bars show how how many percent of the labels are assigned by each model. Here you can see that compared to the NVD the FBR Beta=1 model generates less labels, while the Beta=4 model generates significantly more labels. After fine-tuning the thresholds of the models we can see with the gray bars that the expert Beta=1 model also generates more labels than NVD. Showing that fine-tuning the thresholds achieves the goal of being able to compensate for the missing labels. For the Beta=4 model the fine-tuning seemed to lower the amount of assigned labels, which can also explain the lower recall on this model. These results also seem to be representative from what we saw in the evaluation in Table VII.

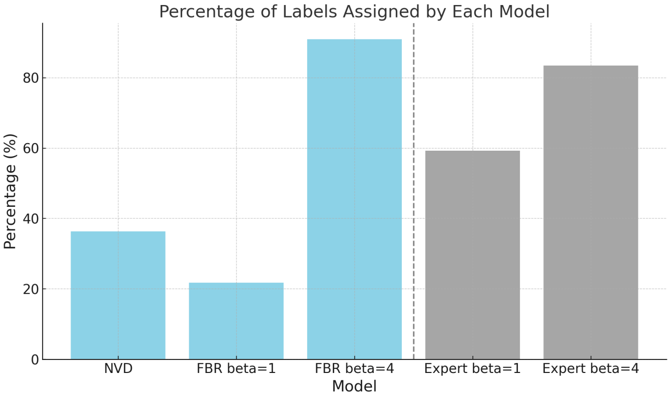


Figure 7. Model label distribution

VI. DISCUSSION

The goal of this study was to improve the automation of CWE classification by tackling four main problems: the problems with manual labels (RQ1); the difficulty in measuring performance on a dataset with missing labels (RQ2); finding methods to suggest relevant missing weakness labels (RQ3); and making a classification model better by using chosen expert labels (RQ4). The findings of our study provide valuable insights into potential problems and solutions for weakness classification, which are also more generally relevant for practical Natural Language Processing (NLP) within other cybersecurity applications.

A. Manual label limitations (RQ1)

In our preliminary research, we addressed several limitations of the NVD dataset. This step is crucial for automating CWE labeling, as the performance of a model depends on the data used for training, and a high level of inconsistencies in the data could potentially reduce the method's overall performance. A clear case for this is that 4.18% of the CVE descriptions are duplicates and occur more than once. Of these duplicates, 26.70% have different labels for the duplicate description, which is already conflicting data. When a model undergoes training using these descriptions, the varying labels could significantly effect the learning process and limit the model's performance. The second established limitation is the significant differences in labeling practices between different CVE Naming Authorities (CNAs) for their respective CVEs. This is shown with some

CNAs having 0.1% duplicate descriptions, while others have over 30%. Similarly, some CNAs only have 51 CWEs labeling their CVEs, while others have 326, a significant range that creates a lot of variance in the dataset. Finally, we showed that inconsistent application of the NIST unique labels 'NVD-CWE-Other' and 'NVD-CWE-noinfo' can result in incorrect classification of a CVE. The NVD dataset uses these labels over 56.266 times, which is a significant portion of the 224.040 CVEs we analyzed, highlighting the importance of these labels.

We have tried to mitigate the effects of these limitations in two main ways. Firstly, we added all duplicate CVEs together to become one comprehensive CVE. Secondly, we eliminated all 'NVD-CWE-Other' and 'NVD-CWE-noinfo' labels due to their abstract and inconsistent nature. This made our data more uniform; however, it does not fully address the structural problems with the NVD dataset and varying labeling methods between CNAs. The decision to filter out the 'NVD-CWE-noinfo' label does affect the labeling capabilities of the model. When the model encounters real-world CVEs, it might not specify any weakness information. However, in the absence of an 'NVD-CWE-noinfo' label or a similar substitute, the model would incorrectly attempt to classify a weakness. The risk of this mislabeling should be carefully considered before using the model without an expert in the loop.

Another aspect that was not implemented in this model is the inclusion of information beyond the CVE description for the CWE classification. Recommending weaknesses while taking into account the CNA who posted the CVE could lead to more relevant recommendations. Additionally, CVE's are usually posted into the NVD with references to sources; these sources were also not utilized in this method. External sources, in addition to the CVE description of the NVD, could provide crucial information about the weaknesses, potentially leading to improved performance.

While simply merging the duplicates and excluding 'non-relevant' CWE labels does limit the scope and capability of our model, it does provide a simple method for cleaning up the dataset. In some cases, duplicate CVE descriptions do address different vulnerabilities, and ideally the model would be able to address these as such; however, this would require more sources and a structured approach to solve. Within the scope of our research, the analysis of the NVD dataset clearly revealed some potential causes and symptoms of the bad labels. We utilized this knowledge in the pre-processing of our model, which likely helped create a more competitive model to the NVD labels based on our evaluation.

B. Measuring performance (RQ2)

We were able to evaluate our model more independently by using manual labels created by experts, as opposed to the NVD labels. Previously, methods would rely solely on NVD labels as the ground truth, despite the limitations of these labels, as shown in our preliminary experiments. Despite the inherent flaws in our expert manual labeling process, such as over-labeling and other common human biases, we argue that many of these biases are also present in current manual NVD labels. Additionally, we would argue that in the realm of security, we prefer false positives over false negatives, particularly in the area of high-risk weaknesses. Our evaluated CWE-GEM Beta=4 clearly adheres to this concept, which is shown by its high recall in sacrifice for a lower precision, a key characteristic of having many false positives.

Our manual evaluation method for measuring performance is not without its own biases. The majority of labels assigned to the participants were for models with more generous thresholds. This can create a bias that, through random selection by the experts, these models might seem to perform better. A consideration could have been adding an additional fake label as a fail-safe to be able to see if the experts filled it in genuinely. However, implementing this would require some nuance, as it remains possible for an expert to interpret the 'fake' weakness label as relevant. For this reason, the failsafe was not implemented, which sadly did not allow us to measure the validity of the expert labelers. Furthermore, we should not underestimate the abstract semantics involved in assigning weakness labels. This was also observable in our evaluation because there are multiple cases of three experts each assigning a different weakness label, as shown in Appendix D.

This method of manual expert evaluation provides a solution for measuring label performance in the presence of missing labels in the dataset. However, the need to rely on the expertise of expert reviewers and the potential for varying interpretations pose significant limitations. This also might seem slightly hypocritical after elaborating on how the manual labeling of NVD is a cause for concern. Despite this, this method proved sufficient for the scope of our research, enabling us to obtain new curated labels and address the research question.

C. Solving missing labels (RQ3)

Enhancing the recall is essential when transitioning from single-label data, such as the current NVD, to a more representative multi-label scenario. The issue of missing labels presents a significant challenge in this field, prompting the use of PU learning and the FBR algorithm to enhance recall. The key advantage of PU learning, particularly for training data with missing labels, is its noise resistance, which limits the negative impact of missing labels on model performance. This aligns perfectly with the FBR algorithm, enabling us to prioritize recall over precision. Even though PU learning aids in labeling missing labels, the chosen thresholds ultimately dictate the classification process. Missing labels also negatively impact this process, as the threshold calculation does not take into account the absence of labels. Therefore, in our research, we showed that by using the FBR algorithm with an increased beta of 4, it is possible to increase the recall performance of our models. In this case, we want to lower our false negatives by increasing our false positives, which is not ideal but would help significantly with assisted labeling.

The idea of using PU learning along with over-labeling is based on the idea that the NVD's positive labels are broad enough to let the model converge on identifying the right weaknesses. The validity evaluation of the NVD labels (RQ1) already casts doubt on whether this assumption holds true in the current dataset. In addition, certain weaknesses, like CWE-920, are extremely rare, appearing only three times in total. Other techniques, like combining self-learning with active learning, could potentially enhance performance, enabling experts to assist the model only in situations of extreme uncertainty. This could have been a more promising approach for further improvements. The current approach of positive unlabeled learning was shown to be effective in previous research for finding missing labels; this allowed us to experiment with the effect of over-labeling using the FBR-algorithm. In our research, we found that the over-labeling method outperformed the previous methods on both recall and F1 scores, addressing the question of whether it is possible to recommend relevant missing labels.

D. Expert fine-tuning (RQ4)

We also introduce considerations for preventing excessive over-labeling by introducing active learning in the form of using expert evaluated labels for fine-tuning. After refining the models using expert labels and reevaluating them again, we observed a significant improvement on both recall and precision compared to the NVD. This feedback mechanism enables us to develop a method that performs similar, if not better, than the NVD in both metrics. This means that this method is a low-effort way to build a model that can sometimes improve on the original dataset, utilizing only a limited number of manually assigned expert labels.

It should be noted that this situation exacerbates the limitations discussed for the manual evaluation (RQ2). Poor labeling by experts on the expert-curated labels used for the training could potentially deteriorate the model's performance. This is a big issue for this method and can even happen when the experts are genuine but have certain biases. Using more experts and generating more labels can mitigate this issue, ultimately resulting in a stabilizing effect that enhances predictability and accuracy in the model's performance, potentially leading to a significantly improved model. However, in our method, we found that using a small subset of the labels to generate fine-tuned thresholds resulted in slight improvements in precision, and the model scored higher on both recall and precision compared to previous iterations. Which does prove that expert evaluation could be used for improving the model; however, this definitely depends on the quality of the labels assigned by the experts and the amount of labels used.

E. Limitations

Several limitations in this study warrant discussion. The reliance on the accuracy and completeness of labels from the CWE-1003 standard and the CVE entries was a significant constraint. The case studies discussed in SectionIII showed that key information can be missing or unclear in CVE descriptions. This could hinder the model's ability to identify a weakness, leading to a decline in the model's performance. Also, some loss functions, like Positive Unlabeled (PU) learning, can help with uneven data, but they might not always be able to handle poorly labeled or irrelevant CWEs well, which could lead to wrong classifications.

Another big limitation of this research was that it focused more on a model-centric approach for solving the missing labels than a data-centric approach. Machine learning research has consistently demonstrated in recent years that data quality is the bottleneck, commonly referred to as the 'garbage in equals garbage out' phenomenon. Our method tried to work around the low-quality data by adapting it to compensate for the shortcomings, such as missing labels. However, in a data-centric approach, a more thorough filtering of bad quality data would be used. Such as generating more labels through regex or automated labeling through more traditional active learning methods.

Lastly, we suffer from the limitations of the Common Weakness Enumeration itself. Discussing the limitations of single-label CWEs with experts from the CISCO Talos group, we found that the protocol they use only wants them to label a CVE with the most important CWE. They proposed that other forms of integrating CWEs could allow for a more thorough understanding of the relationship between the weaknesses. A good example for this is TALOS-2024-2004[4], which involves both CWE-125: Out-of-bounds Read and CWE-200: Uncontrolled Resource Consumption. Experts prioritize CWE-125 because it triggers the CWE-200 vulnerability, leading to the labeling of this vulnerability solely with CWE-125. According to those experts, this could be mitigated by creating a relational format within the listing of CWEs, such as a tree where CWE-125 facilitates CWE-200. Such a new format would allow researchers to create chains of weaknesses, which then allow for deeper understanding.

F. Future Work

Looking ahead, there are several promising avenues to further enhance CWE classification. A natural extension of this research would be to evaluate if the performance improvements we observed with expert feedback persist across more iterations. By continuing this feedback loop, we could fine-tune model predictions and incrementally improve both recall and precision in a systematic way.

Integrating more traditional active learning with self-learning methods could significantly expand the labeled dataset. For example, the model could label high-confidence predictions from unlabeled CVEs, which would help increase the dataset size. For low-confidence cases, expert reviewers could provide manual labels, similarly to our method, allowing the model to iteratively improve its labels and generate more accurate CWE classifications.

Another crucial area for future research is improving data quality. As demonstrated in this study, data inconsistencies, such as duplicate CVEs and differing CNA labeling practices, impact model performance. Another approach could involve leveraging regex-based methods to generate additional CWE labels based on the CVE descriptions. In our experiments, this approach yielded a 5% improvement in model performance by adding approximately 15,000 new labels. However, future work should focus on validating these labels to minimize the risk of inadequate quality data and ensuring data integrity.

The ultimate objective is to implement these methods in the industry. Deploying small, explainable models could aid experts in labeling CWEs during the creation of CVEs. These models could then also be integrated directly into the CVE life-cycle, providing real-time feedback and improving the quality of the data over time. By continuously improving these models based on both human input and feedback learning, we can improve performance and reliability.

VII. CONCLUSION

In this research, we introduced the CWE-ASSIST framework for automating CWE classification, addressing key issues in

[4] https://talosintelligence.com/vulnerability_reports/TALOS-2024-2004

manual labeling, and improving the recall using expert feedback. Our findings demonstrate that CWE-ASSIST not only competes but can outperform current NVD labels, providing a semi-automated tool for generating relevant weakness labels. Within these advancements, certain limitations remain. Variations in labeling practices across different CVE Naming Authorities (CNAs) and structural issues within the CWE system still pose challenges. Future improvements could focus on addressing these inconsistencies through more data-driven approaches, such as active learning and self-learning, which could further improve model accuracy and performance. However, the CWE-ASSIST framework has significant potential for real-world application. Through integrating assisted labeling models into the CVE life-cycle, professionals could automate and improve on the CWE labeling, ultimately improving the quality and consistency of CVE data.

References

[1] E. Aghaei, W. Shadid, and E. Al-Shaer. Threatzoom: Hierarchical neural network for cves to cwes classification. In *International Conference on Security and Privacy in Communication Systems*, pages 23–41. Springer, 2020.

[2] M. Aota, T. Ban, T. Takahashi, and N. Murata. Multi-label positive and unlabeled learning and its application to common vulnerabilities and exposure categorization. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 988–996, 2021.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[4] R.-E. Fan and C.-J. Lin. A study on threshold selection for multi-label classification. *Department of Computer Science, National Taiwan University*, pages 1–23, 2007.

[5] Y. Kim. Convolutional neural networks for sentence classification, 2014.

[6] Y.-J. Lin and C.-J. Lin. On the thresholding strategy for infrequent labels in multi-label classification. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2023.

[7] B. Martin. Common vulnerabilities enumeration (cve), common weakness enumeration (cwe), and common quality enumeration (cqe) attempting to systematically catalog the safety and security challenges for modern, networked, software-intensive systems. *ACM SIGAda Ada Letters*, 38(2):9–42, 2019.

[8] P. Mell and T. Grance. Use of the common vulnerabilities and exposures (cve) vulnerability naming scheme. *NIST Special Publication*, 800:51, 2002.

[9] MITRE. Cve lifecycle.

[10] MITRE. Cve program partners.

[11] J. Ni, C. Qu, J. Lu, Z. Dai, G. H. Ábrego, J. Ma, V. Y. Zhao, Y. Luan, K. B. Hall, M.-W. Chang, et al. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*, 2021.

[12] NIST. Cves and the nvd process, Sep 2022.

[13] NIST. Nvd cwe slice, Sep 2022.

[14] M. Pan, P. Wu, Y. Zou, C. Ruan, and T. Zhang. An automatic vulnerability classification framework based on bigru-textcnn. *Procedia Computer Science*, 222:377–386, 2023.

[15] H. Su, W. Shi, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-t. Yih, N. A. Smith, L. Zettlemoyer, and T. Yu. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*, 2022.

[16] T. Wang, S. Qin, and K. P. Chow. Towards vulnerability types classification using pure self-attention: A common weakness enumeration based approach. In *2021 IEEE 24th International Conference on Computational Science and Engineering (CSE)*, pages 146–153, 2021.

[17] Y. Zhang, Y. Cheng, X. Huang, F. Wen, R. Feng, Y. Li, and Y. Guo. Simple and robust loss design for multi-label learning with missing labels, 2021.

Appendix

A. *Generic CVE Descriptions*

The CVE description: ’Windows Kernel Elevation of Privilege Vulnerability’, which has 78 distinct CVE-IDs linked to 17 distinct CWEs, is an extreme example of a duplicate (Figure A). Given the generic nature of the description, it is likely that these vulnerabilities were unique, but the description did not adequately address this. This notion is supported by the fact that the CVE description spans five different years.

NVD created a class named ’NVD-CWE-noinfo’ for circumstances where a CVE does not provide enough information to be classed. However, as demonstrated in our example, this designation is not always used appropriately.

Furthermore, NVD includes additional sources for investigation, such as Microsoft’s supplementary documentation [5], although mostly contain identical information. Additionally, some of these CVEs include an Analysis Description, which frequently merely mentions the following. ”There is a vulnerability in the Windows kernel that allows for privilege elevation. This CVE ID is separate from: CVE-2023-21747,...”,

[5] https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-21675

CVE ID	CWE
CVE-2020-17035	NVD-CWE-noinfo
CVE-2021-1682	CWE-269
CVE-2021-31979	CWE-119
CVE-2022-21881	CWE-362
CVE-2022-34707	CWE-416
CVE-2023-21675	CWE-843, NVD-CWE-noinfo
CVE-2023-21749	CWE-20, NVD-CWE-noinfo
CVE-2023-21750	CWE-284, NVD-CWE-noinfo
CVE-2023-21754	CWE-190, NVD-CWE-noinfo
CVE-2023-21772	CWE-125, CWE-269
CVE-2023-28222	CWE-59, NVD-CWE-noinfo
CVE-2023-28236	CWE-591, NVD-CWE-noinfo
CVE-2023-28272	CWE-191, NVD-CWE-noinfo
CVE-2023-35304	CWE-122, NVD-CWE-noinfo
CVE-2023-35359	CWE-23, NVD-CWE-noinfo
CVE-2023-38141	CWE-367, NVD-CWE-noinfo
CVE-2024-21338	CWE-822, NVD-CWE-noinfo

Table VIII
DUPLICATE GENERAL CVE DESCRIPTIONS WITH DIFFERENT CWE LABELS

Table IX
DUPLICATE SPECIFIC CVE DESCRIPTIONS WITH DIFFERENT CWE LABELS

CVE ID	CWE
CVE-2023-39544	CWE-862
CVE-2023-39545	CWE-552
CVE-2023-39546	NVD-CWE-noinfo, CWE-836
CVE-2023-39547	CWE-294
CVE-2023-39548	CWE-434

which does not help our analysis. It is possible that certain information was accessible to NVD and Microsoft for classification but was not made public. CVE-2024-21338, labeled by NVD as ’NVD-CWE-noinfo’ and by Microsoft as ’CWE-822’ (Untrusted pointer de-reference), is a sample that demonstrates this.

For this vulnerability, the NVD page says that the CVE has also been disclosed in the ’CISA’s Known Exploited Vulnerabilities Catalog’, which also provides mitigating advice. The CISA Catalog gives the following additional information: ”Microsoft Windows Kernel contains an exposed IOCTL with insufficient access control vulnerability within the IOCTL (input and output control) dispatcher in appid.sys that allows a local attacker to achieve privilege escalation.” This description is undoubtedly helpful, and it is unclear why it was not included in the initial CVE description. As additional information the NVD page also includes a link to a blog[6] that goes into great detail about how the Lazarus Group exploited this vulnerability as a zero-day, allowing them to perform direct ’kernel object manipulation’.

B. Multi CVE labeling

It is also possible for highly particular vulnerability to be classified with distinct CWE. The vulnerability described as ’CLUSTERPRO X Ver5.1 and earlier and EXPRESSCLUSTER X 5.1 and earlier, CLUSTERPRO X SingleServerSafe 5.1 and earlier, EXPRESSCLUSTER X SingleServerSafe 5.1 and earlier allows a attacker to log in to the product may execute an arbitrary command.’ has 5 unique CVE-IDs associated with 6 different CWE (Figure IX). As revealed on their website [7], it appears that the relevant CNA, the ’NEC Corporation’, creates a new CVE for each vulnerability exploited. This is, of course, not what was expected, as according to NVD protocol, this should be designated as a single CVE ID that corresponds to numerous CWEs.

C. NVD mislabeling

A clear example of the first case is shown in CVE-2023-44253, which NIST labeled as ’NVD-CWE-noinfo’ and Fortinet as ’CWE-200’, here CWE-200 was specified in the description. ”An exposure of sensitive information to an unauthorized actor vulnerability [CWE-200] in Fortinet FortiManager version 7.4.0 through 7.4.1 and before 7.2.5, FortiAnalyzer version 7.4.0 through 7.4.1 and before 7.2.5 and FortiAnalyzer-BigData before 7.2.5 allows an adom administrator to enumerate other adoms and device names via crafted HTTP or HTTPS requests.”

However a bad example of the first case is shown CVE-2024-21371, which NIST again labeled as ’NVD-CWE-noinfo’and Windows as ’CWE-367’, but there is no weakness specified. ”Windows Kernel Elevation of Privilege Vulnerability”

A clear example of the second case is shown in CVE-1999-0059, here NIST labeled it as ’NVD-CWE-Other’ while CISA-ADP correctly labeled it as ’CWE-200’. ”IRIX fam service allows an attacker to obtain a list of all files on the server.”

However a bad example of the second case is shown in CVE-2020-35167, which NIST labeled as ’NVD-CWE-Other’ while Dell as ’CWE-200’ which is incorrect in this case. ”Dell BSAFE Crypto-C Micro Edition, versions before 4.1.5, and Dell BSAFE Micro Edition Suite, versions before 4.6, contain an Observable Timing Discrepancy Vulnerability.”

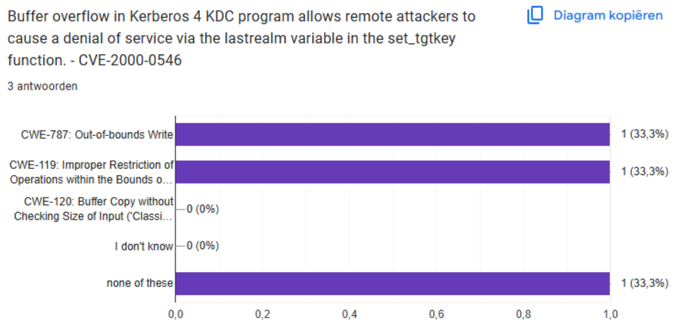
A clear example of the third case is shown in CVE-2002-2374, here NIST labeled it as ’NVD-CWE-noinfo’, ’CWE-59’, ’CWE-362’. ”Unspecified vulnerability in pprosetup in Sun PatchPro 2.0 has unknown impact and attack vectors related to ”unsafe use of temporary files.””

Another example of the third case is shown in CVE-2022-33715, here NIST labels it as ’NVD-CWE-Other’, ’CWE-22’ and Samsung labels it as ’CWE-22’. Where ’CWE-22’ and ’CWE-20’ are correct. ”Improper access control and path traversal vulnerability in LauncherProvider prior to SMR Aug-2022 Release 1 allow local attacker to access files of One UI.”

D. Experts disagreeing

In our evaluation we have multiple samples in which each of the 3 experts selected a different weakness. Two of these cases are given below Figure 8 and Figure 9, these highlight that through semantic ambiguity and different interpretations experts can disagree on relevant weaknesses.

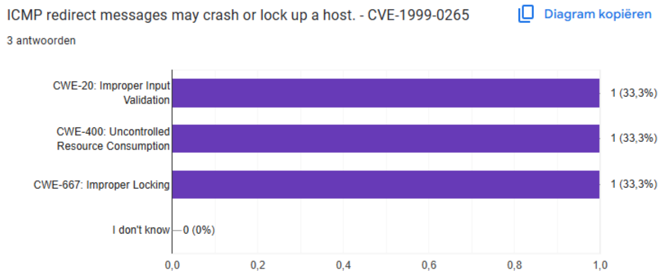
[6] https://decoded.avast.io/janvojtesek/lazarus-and-the-fudmodule-rootkit-beyond-byovd-with-an-admin-to-kernel-zero-day/
[7] https://jpn.nec.com/security-info/secinfo/nv23-009_en.html



Buffer overflow in Kerberos 4 KDC program allows remote attackers to cause a denial of service via the lastrealm variable in the set_tgtkey function. - CVE-2000-0546

3 antwoorden

Figure 8. First sample of experts disagreeing



ICMP redirect messages may crash or lock up a host. - CVE-1999-0265

3 antwoorden

Figure 9. Second sample of experts disagreeing