Tracking the Evolution: Uncovering Concept Drift in Vulnerabilities Descriptions Over Time

Hugo van Wijngaarden
Department of EEMCS, University of Twente
h.l.vanwijngaarden@student.utwente.nl

June 29, 2025

The field of cyber security is one that is constantly changing and evolving. A part of this field consists of Common Vulnerabilities and Exposures (CVEs) and their descriptions. The nature of this field makes it difficult to keep machine learning models that map CVEs to Common Weakness Enumerations (CWEs) up to date. To keep these models relevant, this paper addresses the problem of concept drift in vulnerability descriptions. By finding an optimal training window, this paper improves the training strategy of language models on the CVE dataset to better reflect the current cyber security landscape and allows for a more accurate mapping of CVEs to CWEs. Various time windows have been evaluated, in Which the models trained on the two years immediately preceding the test set gave the best results. With this approach, a system for maintaining model relevance over time is proposed. This methodology will allow for a more accurate dataset of CVEs mapped to CWEs to be used in detecting cyber security threats.

Keywords — CVE, CWE, cyber security, machine learning, concept drift, sliding window

1 INTRODUCTION

cyber security is a rapidly evolving domain. Failure to maintain up-to-date language models risks causing detection systems to falsely identify innocent activities while overlooking actual threats. Such failures can lead to significant security compromises for both businesses and consumers.

Cyber threat analysis fundamentally relies on understanding vulnerabilities. By analysing how CVE descriptions evolve over time, defenders can not only enhance their understanding and strengthen their defenses but also improve the performance of machine learning tools built for security analysis.

Most current machine learning models that attempt to detect cyber security threats are built upon the CVE and CWE database. The problem is that not every CVE has a CWE attached to it. In 2023, nearly 5,000 CVEs (20% of all published CVEs that year) lacked association with any specific CWE [14]. This lack of mapping from CVEs to CWEs causes those models to be less extensive than can be. This study aims to solve this issue by answering the following research question:

What training window for mapping CVEs to CWEs gives the highest weighted F1-score?

To support this investigation, the following sub-research questions are proposed:

- How does concept drift manifest in CVE descriptions over time?
- How does the size of the training window influence model accuracy for different test years?
- How does the number and distribution of CWE classes vary across different training windows?

2 BACKGROUND

2.1 Common Vulnerabilities and Exposures (CVE)

CVEs are publicly disclosed cyber security vulnerabilities that have been identified and solved. The CVE database currently contains approximately 284,000 records [10]. Each record has a description explaining what vulnerability was discovered and the consequences that that vulnerability had.

2.2 Common Weakness Enumeration (CWE)

CWEs are a community-developed classification of common software and hardware weaknesses that may have security implications [9]. A ”weakness” refers to a condition within a software, firmware, hardware, or service component that, under certain conditions, could facilitate the introduction of vulnerabilities.

These CWEs serve to categorize all the different CVEs. For example, CVE-2018-15631 has the following description: ”Improper access control in the Discuss App of Odoo Community 12.0 and earlier, and Odoo Enterprise 12.0 and earlier allows remote authenticated attackers to e-mail themselves arbitrary files from the database, via a crafted RPC request.” This description is attached to CWE-862 (”The product does not perform an authorization check when an actor attempts to access a resource or perform an action”) and CWE-340 (”The product uses a scheme that generates numbers or identifiers that are more predictable than required”).

2.3 Concept Drift

Concept drift refers to the phenomenon where the statistical properties of the target variable (CVE descriptions in this case), which a predictive model is intended to learn, change over time in unforeseen ways. As a result, the model's assumptions about the data distribution become invalid, leading to performance degradation if the model is not updated accordingly [15]. For example, a model trained on online shopping behavior trained before the COVID-19 pandemic will behave differently compared to one trained during or after corona because during that time period online shopping behavior changed drastically.

3 RELATED WORK

Within the field of concept drift, there have been many different studies that have addressed the problem and proposed a solution. Some of these studies try to tackle concept drift in a general way by proposing a framework that can work for any field [11] [13] [7].

There are also studies that look specifically at concept drift in cyber security [4] [8].

This section reviews the existing literature on the topic and groups the concept drift detection methods into different categories based on this study [1].

During this review, we will discuss the relevance of these different methods for this study and highlight important papers that support the claims this paper makes.

3.1 Concept Drift Is An Issue

We first have to note that concept drift is an issue within cyber security. This has been concluded in this study [3], where significant deviations between clusters were found, indicating the presence of concept drift within cyber security.

3.2 Concept Drift Detection Methods

3.2.1 Window Based Methods

This type of approach accumulates the incoming data instances and forms a batch of data (or a window). Generally, the window-based methods contain two windows. The first window is used to store old instances, and later has new instances of the data stream. The comparison between these two window instances explained the change in data distribution and signaled the drift.

This is the methodology that this paper has followed since we are working with a static dataset, so windows are easily defined and trained on. But instead of taking two windows and comparing their distribution, a large set of different window sizes is taken, allowing for a more complete comparison of different window sizes. This method also allowed for easier answering of the main research question.

3.2.2 Distribution Based

These techniques focus on the distribution of the data to detect concept drift. For this study, the similarity and dissimilarity-based method could be used to prove that there is a concept drift in the CVE-CWE mapping. However, it falls out of the scope of this study, so the decision was taken to exclude the distribution-based concept drift detection technique.

Similarity And Dissimilarity Based Methods These approaches are based on measuring the similarity and dissimilarity among the distribution of data samples with respect to time. In the case of this paper, the distribution of CVEs that are already mapped to CWEs would be analysed. If this distribution shows large deviations year-over-year, you could conclude that concept drift has occurred.

Statistical Based Methods Statistical based methods are used to detect the concept drift by comparing the distribution of historical and current data instances using statistical tests like Mean, Median, Mode, Kurtosis, Standard Deviation, Regression, Hypothesis Testing, etc. This is a technique mostly used for data streams, so it is not as applicable to this research as the window-based methods.

3.2.3 Threshold Based

These techniques set a threshold on a certain distribution or metric of the model and detect concept drift when this threshold is surpassed. Because these thresholds are difficult to set and since it is unnecessary to detect the exact point where concept drift occurs, this method has not been used during this study.

Sequential Analysis Based Methods In this type of concept drift detection algorithm, the data instances are examined sequentially to analyse the change in the data stream context. It signals the drift when the change in data distribution exceeds the specified threshold.

Decision Boundary Based Methods Decision Boundary based Methods generally form a boundary using initial instances of the data stream. The change in decision boundary is considered to be concept drift.

4 METHODOLOGY

Building on the insights from related work, this study proposes a method for detecting concept drift and identifying an optimal time window for training machine learning models to map CVE descriptions to CWE classes. The proposed methodology, based on [2], is illustrated in Figure 1. To extend this research, a method for measuring CVE description quality is proposed and a formula for ranking different time frames is presented.

The model is trained in the following way:

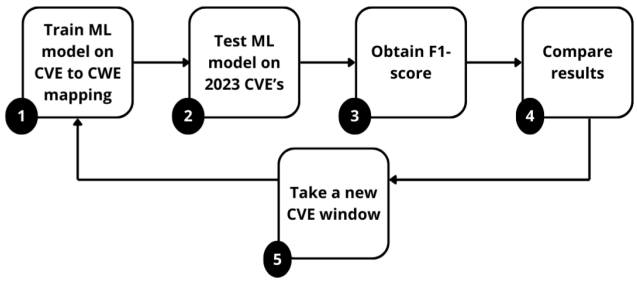

Figure 1: Pipeline showcasing the workflow for this paper.

1. The first step is to use BERT-based encoding. Because this study focuses on CVE descriptions, which are written in text instead of numbers, the descriptions have to be encoded before the machine learning model can be trained.

2. The second step is to load the encoded data. The aim of this study is to analyse a lot of different models, therefore the data is only encoded once and then saved so that the process of encoding the data does not have to be repeated, saving a lot of time.

3. After that the data gets cleaned of any CVEs that do not yet have a CWE attached to them. This step is necessary during the training and testing phase because the model is trained in a supervised way. This means that the model has to know the correct CWE for each CVE; otherwise it cannot train itself.

4. Then all CWEs with less than 10 CVE entries are removed, and depending on the test a limit on the max amount of CVE entries is set, ranging between 100, 1500 and 2500. There are six CWEs with more than 2500 CVEs, nine with more than 1500 and 64 with more than 100 CWE entries out of 117 different CWE categories in the data between 1999 and 2022. These thresholds were chosen because the very large CWE categories (with more than 1500 or 2500 CVEs) could lead to overfitting to those CWEs. To experiment with this even further the experiment with a limit of 100 CVEs was implemented, because it would limit almost half the CWEs. This creates a dataset that is very balanced, without it being too little data to train on.

5. The training data are then divided into the 80% training set and the 20% validation set.

6. After this a shallow neural network with two layers is created. The first layer consists of 128 neurons that process the embeddings and the second layer is the output layer that predicts the CWE category.

7. The model is then trained in 40 epochs with early stopping, which stops the training if the performance does not improve for 5 epochs in a row.

8. For each training dataset, three of these models are trained to get an average performance.

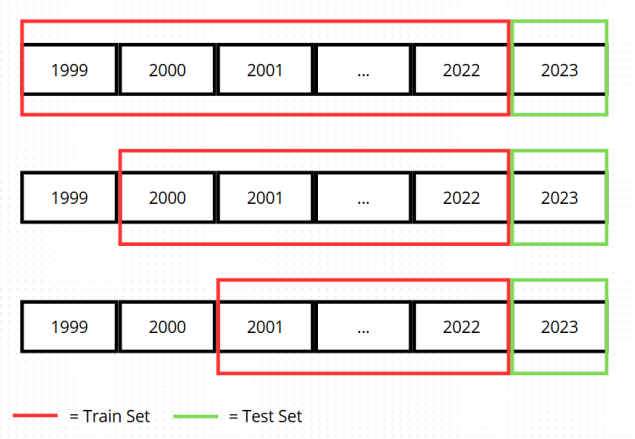

Figure 2: Visualization of the sliding window used to test different training datasets on the same test set.

The decision to use a sliding-window approach was motivated by several factors:

- This paper aims to evaluate different models based on performance. Concept drift detection methods like the threshold or distribution based methods do not provide any performance metrics, making it impossible to rank different models. The sliding window approach allows for easy performance comparison between different time frames.

- The CVE and CWE datasets are static. Although new data is added every year, the datasets used during this study are not data streams by nature. Where other methods often use data streams to detect when concept drift occurs and retrain the moment this happens, this is not relevant for CVE and CWE datasets. The main question of this research is to find the most accurate training window for the CVE and CWE datasets, not to detect when concept drift happens.

- Using the sliding-window approach allowed for easy control over the time frames used. By being able to easily pick what time frames were included or excluded from the training data, the process of gathering relevant data was made a lot easier. The other methods adjusted the time frame when drift occurred and would stop automatically without the user being able to set a time frame to use.

Some clarifications about decisions made for the training process:

- We only used the training data closest to the test set. As illustrated in Figure 2, with each iteration the year that is furthest away from the test set is excluded. Because concept drift is a temporal effect, the potential for a model trained on data that excludes the most recent data compared to the test set was considered to be very low. After analysing some early results however, the decision was made to do a single test on data be-

tween 1999 and 2008, but tested on 2023. This test was done to validate and explain the obtained results.

- We only used a single year as a test set. This allowed for easier generalization of the results. Using a single year was considered as sufficient, as with each year there are more CVEs in the database than in the year before that. This makes sure that there is always a sufficient amount of data to test on.
- To analyse the results, the weighted and macro F1-scores were saved. These scores balance precision (how many of the model's positive predictions are correct) and recall (how many of the actual positive cases the model correctly identifies) to give a good overview of the models performance. The weighted F1-score calculates the F1-score for each CWE class and then averages them, weighting each class by its support (the number of samples for that class in the test set). This means classes with more samples (e.g., common CWEs like CWE-79) have a bigger impact on the final score. The macro F1-score calculates the F1-score for each CWE class and then takes a simple average, giving equal weight to every class, regardless of how many samples it has. So, rare CWEs (e.g., CWE-319 with 10 samples) have the same influence as common ones (e.g., CWE-79 with 1000 samples). The weighted F1-score was prioritized during this study, as it reflects the real world the best. The macro F1-scores are mentioned in Appendix B.

This paper also analyses the decrease in CVE description quality. A template-based keyword analysis was conducted to measure the quality of CVE descriptions over time. This template, defined in earlier guidelines [6] has been used to guarantee CVE description quality and uniformity. Since it can create a bottleneck for registering a new CVE, this template does not always get adhered to. Two tests were conducted to inspect the effect that not adhering to the template has on the quality of CVE descriptions: one evaluating the presence of required keywords and another assessing whether the keywords also appeared in the correct order.

The last step to take was to analyse the results. This has been achieved by ordering the F1-scores for each test year from best to worst and looking at how many years you have to include in the training data compared to the test data to get the best scores. This method is based on [5], which proved that this method generates relevant results for ranking different models. The method gives ranks to each F1-score. These ranks then get added up and divided by the amount of datasets to get an average rank. By analysing these ranks, a definitive recommendation on which years to include in the training set was able to be given. The calculation used looks like this:
Let r_j^i be the rank of algorithm j on dataset i. We calculate the average rank for each algorithm as

$$\bar{r}_j = \frac{\sum_i r_j^i}{n}, \tag{1}$$

Table 1: Average rank of the three best 'go-back' years based on weighted F1-scores

'go-back' years	average rank
2 years back	5.17
4 years back	5.33
3 years back	5.45

where n is the number of datasets.

5 RESULTS & DISCUSSION

This section presents and interprets the results obtained through the methodology outlined earlier.

The performance of the models was evaluated using the F1-score across multiple train-test year combinations. Figure 3, Figure 4, Figure 5 and Figure 6 display the results of these experiments.

Each line in the plot corresponds to a specific test year, while each point represents a training window starting from the year indicated on the horizontal axis and ending one year before the test year.

For instance, in the 2019 test case of Figure 4 (depicted as the bottom blue line), the point aligned with 2010 on the x-axis corresponds to training on data from 2010 to 2018 and testing on data from 2019. The vertical axis reflects the resulting F1-scores, computed using the standard formulas mentioned in Appendix A. The last graph (Figure 7) was an experiment, to see what the effect of training on just old data but testing it on new data would be.

The results of the research into CVE description quality can be seen in Figures 8 and 9. The vertical axis in these graphs represent the quality of the CVE descriptions and the horizontal axis the year in which those CVE descriptions were written.

Our ranking method gave the following top three 'go-back' years as best performing. See Appendix C for the full results.

Overall, the results seem to point towards a higher performance when training on more recent years. This is also reflected in the ranking method.

Although a 'go-back' year like 23 years back occurs less often (only for test year 2023) than a 'go-back' year like two years back for example (which occurs for every test set), this does not have any effect on the ranking because the ranking is based of average rank across the different models. If 23 years back would perform very well for those test years that it does occur in it would have a very high average rank. From these scores the conclusion is drawn that the best result comes from training on the past two years. Training on the past three or four years also generates good results and can be used as a backup or to validate if

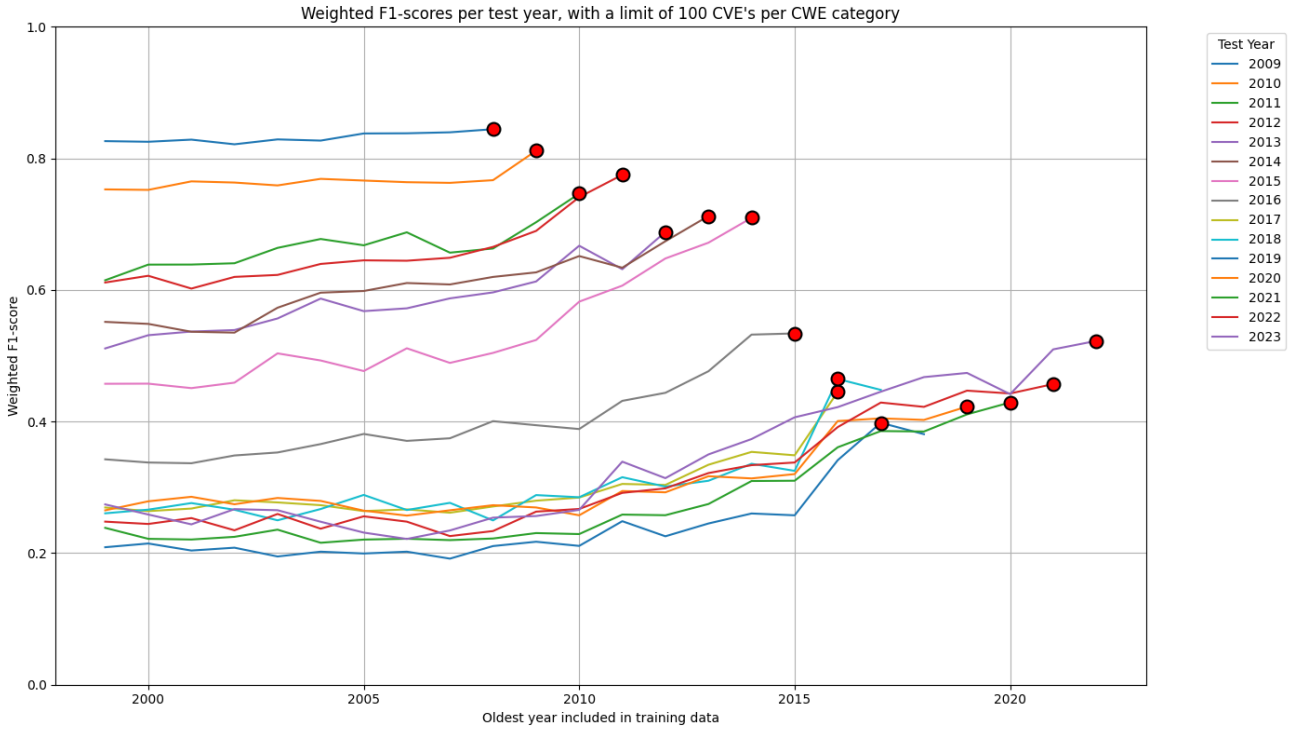

Figure 3: F1-scores obtained from testing different datasets on different test years with a max of 100 CVEs per CWE.

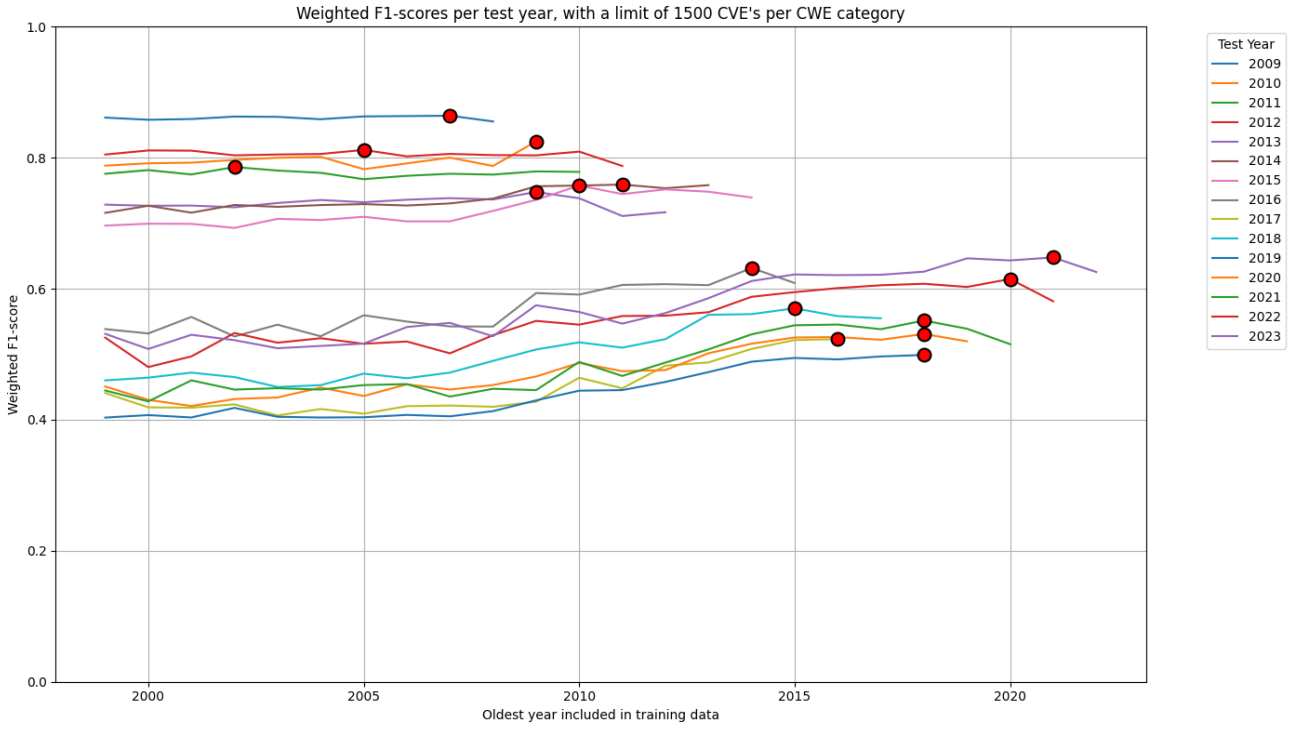

Figure 4: F1-scores obtained from testing different datasets on different test years with a max of 1500 CVEs per CWE.

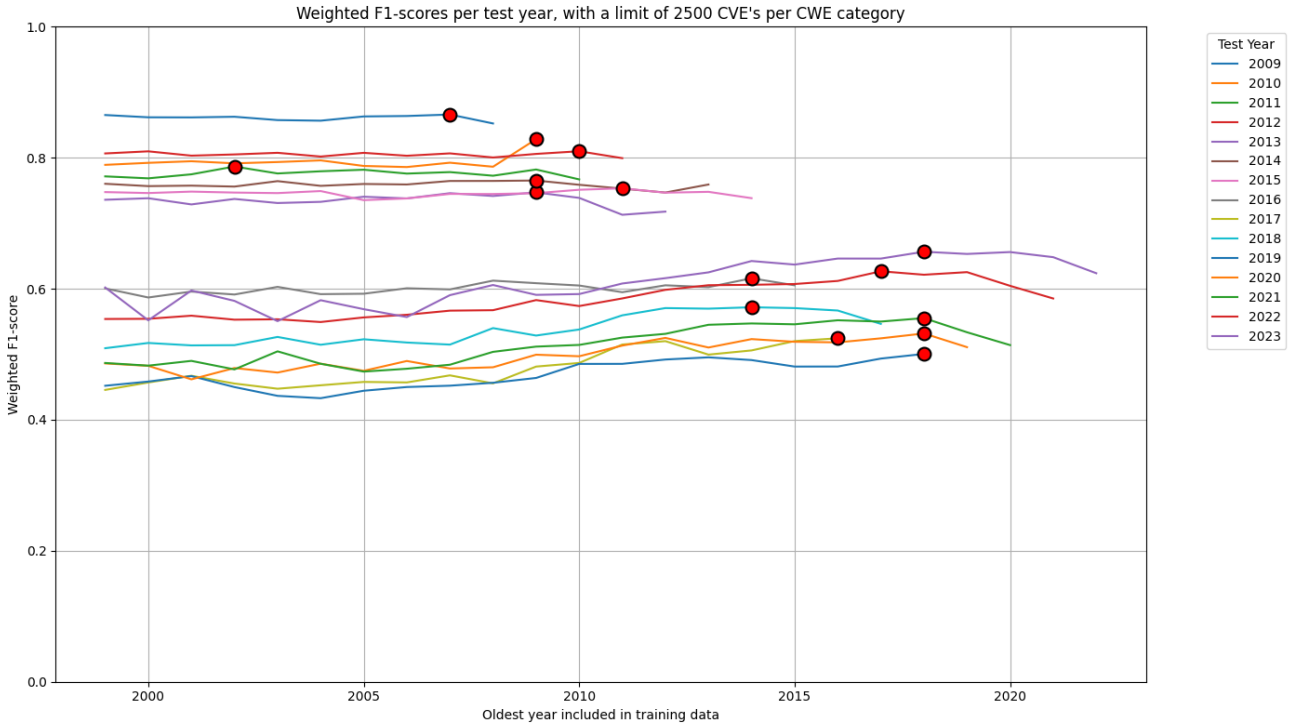

Figure 5: F1-scores obtained from testing different datasets on different test years with a max of 2500 CVEs per CWE.

Figure 6: F1-scores obtained from testing different datasets on different test years with unlimited CVEs per CWE.

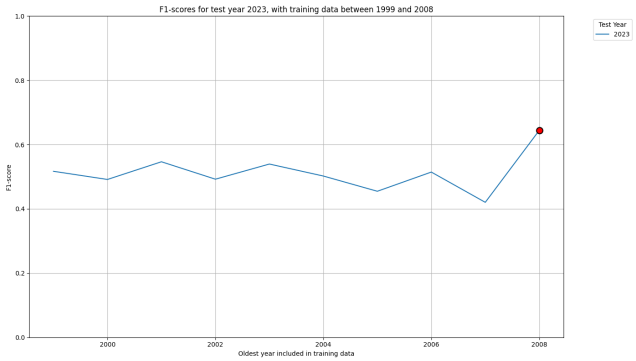

Figure 7: F1-scores obtained from training on data between 1999 and 2009, tested on 2023.

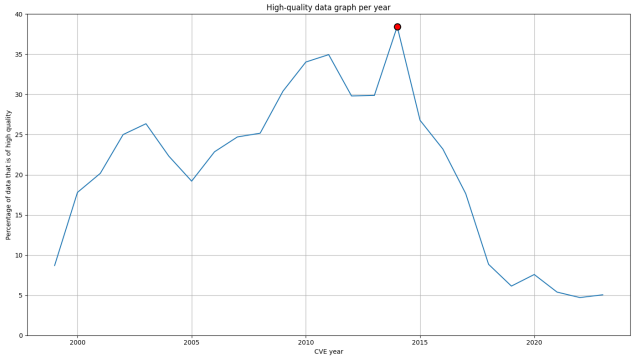

Figure 8: Percentage of high quality CVE descriptions per year not in order.

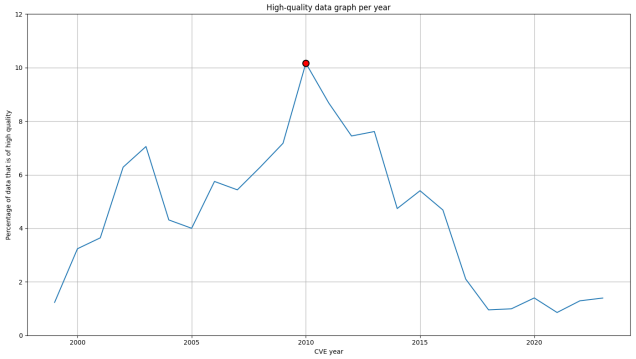

Figure 9: Percentage of high quality CVE descriptions per year in order.

training on just the past two years is the correct choice for a CVE dataset.

Interestingly, performance tends to improve for older test sets, likely due to the reduced complexity in those datasets. To investigate this further, the additional experiment was conducted, in which a model was trained solely on data from 1999 to 2008 and tested on 2023. Contrary to expectations, this model achieved reasonable F1-scores (see Figure 7). This outcome was attributed to the model being trained on a dataset that covered only 18 CWE classes, whereas models trained on more comprehensive windows (e.g., 1999–2022) addressed up to 128 CWE classes. Thus, although the simplified model appears accurate, its practical utility is limited due to its inability to generalize to a wider range of classes. This was accounted for during the testing phase, and any CVE that it could not categorize in the correct CWE category was counted as being predicted wrong. The problem however, is that while this study balanced the training sets by limiting the maximum amount of CVEs per CWE, this was not done for the test set. So the models trained on the old data were still able to classify the very large CWE categories that the training set contained, therefore still achieving a relatively high F1-score while it was unable to classify around 30% of CVEs because the model had not seen their CWE before. This effect underscores a critical issue in interpreting evaluation metrics: higher F1-scores can misleadingly reflect simpler classification tasks.

A similar trend is observed when examining models trained on only the most recent year of data. In the 2023 test case of Figure 3 for example, a noticeable increase in F1-score is observed when the model is trained exclusively on 2022 data. The training set for 2022 consisted of 9,256 CVE entries, compared to 47,586 entries when training from 1999 to 2022. Despite this difference in size, both configurations handled a similar number of CWE classes. This contradicts the common expectation that larger datasets lead to better model performance [12], and instead suggests the presence of concept drift: older data introduces patterns that no longer align with the distribution of current threats, thereby degrading performance.

Furthermore, a consistent decrease in F1-score is observed as the test years become more recent. This degradation is partly attributable to the increasing number of CWE classes introduced over time, which inherently complicates classification, as described above. However, another contributing factor could be a decline in the quality of CVE descriptions. Figures 8 and 9 both reveal a downward trend in description quality over the past decade. Lower-quality descriptions may hinder embedding models like BERT from effectively capturing the semantic information necessary for accurate CWE classification, thereby contributing to lower model performance in recent years.

6 FUTURE WORK

Future research in this area warrants further investigation into alternative methods for concept drift detection beyond the sliding window approach used in this study. Specifically, it would be valuable to investigate whether excluding years identified as having high levels of drift could improve model performance. Additionally, a promising direction involves the development of models that assign dynamic weights to training data from different years based on their relevance to the target distribution. Such an approach could enable models to prioritize recent and contextually relevant data while still being aware of critical information from historical cyber security vulnerabilities.

7 CONCLUSION

In this study, a classification approach was tested to map CVE descriptions to their corresponding CWE categories. The analysis involved evaluating model performance across multiple training windows and test years. The results indicate that, in the majority of cases, training on data from the two years immediately preceding the test year yields the most effective model performance. This conclusion accounts for both the variation in the number of CWE classes present in different training windows and the observed decline in the structural quality of CVE descriptions in recent years, both of which were found to influence classification outcomes.

References

[1] Supriya Agrahari and Anil Kumar Singh. Concept drift detection in data stream mining : A literature review. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part B):9523–9540, 2022.

[2] Massimiliano Albanese, Olutola Adebiyi, and Frank Onovae. Cve2cwe: Automated mapping of software vulnerabilities to weaknesses based on cve descriptions.

[3] Ezza Ali, Nighat Batool, Muhammad Rizwan, and Sohail Sarwar. Assessing concept drift in malware: A comprehensive review and analysis. In *2024 21st International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 564–569, 2024.

[4] Muhammad Amin, Feras Al-Obeidat, Abdallah Tubaishat, Babar Shah, Sajid Anwar, and Tamleek Ali Tanveer. Cyber security and beyond: Detecting malware and concept drift in ai-based sensor data streams using statistical techniques. *Computers and Electrical Engineering*, 108:108702, 2023.

[5] Pavel B Brazdil and Carlos Soares. A comparison of ranking methods for classification algorithm selection. In *European conference on machine learning*, pages 63–75. Springer, 2000.

[6] Jonathan Evans. Key details phrasing.

[7] Fabian Hinder, Valerie Vaquet, and Barbara Hammer. One or two things we know about concept drift—a survey on monitoring in evolving environments. part a: detecting concept drift. *Frontiers in Artificial Intelligence*, Volume 7 - 2024, 2024.

[8] Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. Transcend: Detecting concept drift in malware classification models. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 625–642, Vancouver, BC, August 2017. USENIX Association.

[9] MITRE. Common weakness enumeration: Cwe. 2025.

[10] MITRE. Cve: Common vulnerabilities and exposures. 2025.

[11] Nidhi, Veenu Mangat, Vishal Gupta, and Renu Vig. *Methods to Investigate Concept Drift in Big Data Streams*, pages 51–74. Springer Singapore, Singapore, 2018.

[12] Christopher A. Ramezan, Timothy A. Warner, Aaron E. Maxwell, and Bradley S. Price. Effects of training set size on supervised machine-learning land-cover classification of large-area high-resolution remotely sensed data. *Remote Sensing*, 13(3), 2021.

[13] Methaq A. Shyaa, Noor Farizah Ibrahim, Zurinahni Zainol, Rosni Abdullah, Mohammed Anbar, and Laith Alzubaidi. Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. *Engineering Applications of Artificial Intelligence*, 137:109143, 2024.

[14] Stefano Simonetto, Thijs Sebastiaan van Ede, Peter Bosch, and Willem Jonker. Text2weak: mapping cves to cwes using description embeddings analysis. In *4th Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics*, 2024.

[15] Alexey Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 106(2):58, 2004.

8 APPENDIX

A Formula's

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{4}$$

Precision is a measurement for how accurate the positive predictions are and recall is a measurement for how complete the positive predictions are. Lets say you build a model that detects spam emails. Precision tells you when the model says something is spam and it is actually spam. Recall tells you how many of the spam emails you caught. The higher these scores the better. By combining both these measurements, a more balanced score of the different models was obtained. Red markers in the plot highlight the training set that resulted in the highest F1-score for each test year.

B Graphs

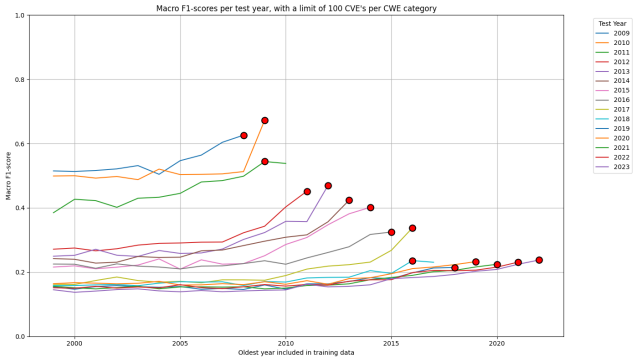

Figure 10: Macro F1-scores obtained from testing different datasets on different test years with a max of 100 CVEs per CWE.

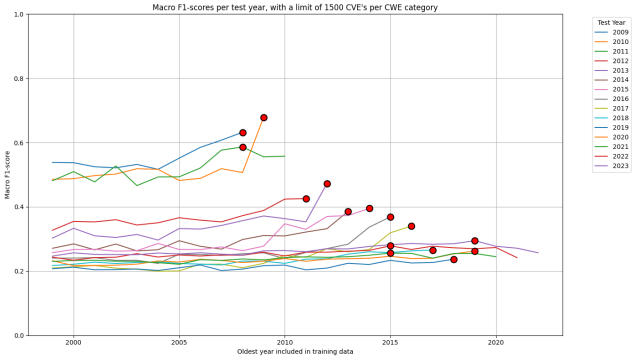

Figure 11: Macro F1-scores obtained from testing different datasets on different test years with a max of 1500 CVEs per CWE.

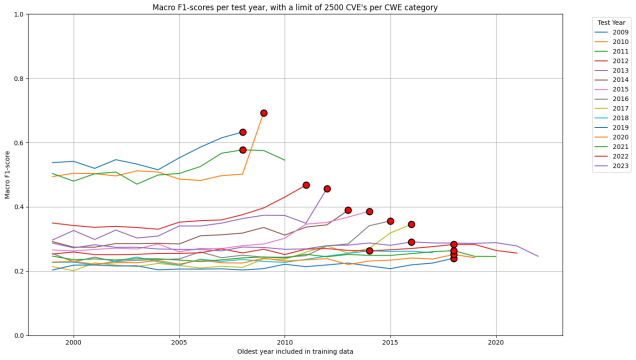

Figure 12: Macro F1-scores obtained from testing different datasets on different test years with a max of 2500 CVEs per CWE.

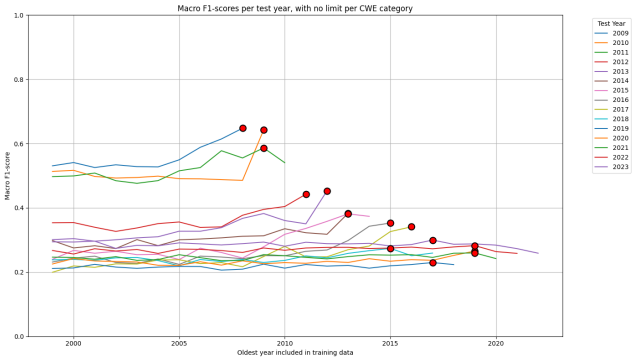

Figure 13: Macro F1-scores obtained from testing different datasets on different test years with unlimited CVEs per CWE.

C Full Ranking Results

Table 2: Ranking for all the weighted F1-scores for models trained with CVE limit of 100, 1500, 2500 and unlimited

'go-back' years	average rank
2 years back	5.17
4 years back	5.33
3 years back	5.45
7 years back	5.93
6 years back	6.10
5 years back	6.18
1 years back	6.77
8 years back	7.55
9 years back	8.47
10 years back	9.45
11 years back	10.43
12 years back	11.50
13 years back	12.10
24 years back	13.25
15 years back	14.05
16 years back	14.56
14 years back	14.52
17 years back	14.66
19 years back	15.29
18 years back	15.57
22 years back	16.42
20 years back	16.65
23 years back	16.88
21 years back	17.38

Table 3: Ranking for all the weighted F1-scores for models trained with no CVE limit

'go-back' years	average rank
7 years back	5.20
6 years back	7.67
10 years back	7.60
8 years back	7.80
4 years back	7.93
12 years back	8.54
9 years back	8.67
3 years back	8.73
24 years back	9.00
23 years back	9.00
2 years back	9.40
19 years back	9.50
13 years back	10.17
17 years back	10.25
11 years back	10.29
5 years back	10.33
15 years back	10.40
16 years back	10.89
1 years back	12.07
14 years back	12.91
18 years back	13.71
20 years back	14.20
21 years back	16.25
22 years back	11.67

Table 4: Ranking for all the weighted F1-scores for models trained with CVE limit of 100

'go-back' years	average rank
1 years back	1.13
2 years back	2.20
3 years back	3.73
4 years back	4.27
5 years back	4.87
6 years back	5.87
7 years back	7.13
8 years back	7.87
9 years back	8.87
10 years back	11.87
11 years back	11.86
12 years back	13.31
13 years back	13.42
24 years back	13.00
19 years back	14.33
18 years back	15.86
17 years back	16.25
14 years back	16.36
15 years back	16.40
20 years back	16.40
21 years back	16.50
16 years back	16.78
23 years back	17.00
22 years back	17.00

Table 5: Ranking for all the weighted F1-scores for models trained with CVE limit of 1500

'go-back' years	average rank
2 years back	3.80
3 years back	3.80
4 years back	3.93
5 years back	4.87
1 years back	5.53
7 years back	5.87
6 years back	6.13
8 years back	7.80
9 years back	8.67
10 years back	9.20
11 years back	9.71
12 years back	12.38
13 years back	13.17
14 years back	14.91
16 years back	15.11
17 years back	15.25
15 years back	15.50
18 years back	16.14
24 years back	17.00
20 years back	17.80
19 years back	18.50
21 years back	19.50
23 years back	20.00
22 years back	20.33

Table 6: Ranking for all the weighted F1-scores for models trained with CVE limit of 2500

'go-back' years	average rank
5 years back	4.67
6 years back	4.73
4 years back	5.20
2 years back	5.27
7 years back	5.53
3 years back	5.53
8 years back	6.73
9 years back	7.67
1 years back	8.33
10 years back	9.13
11 years back	9.86
12 years back	11.77
13 years back	11.67
15 years back	13.90
14 years back	13.91
24 years back	14.00
16 years back	15.44
18 years back	16.57
22 years back	16.67
17 years back	16.88
21 years back	17.25
20 years back	18.20
19 years back	18.83
23 years back	21.50