A Multi-Model Approach to Enhance Automatic Matching of Vulnerabilities to Attack Patterns

Marine Sauze-Kadar and Thomas Loubier
Univ. Grenoble Alpes, CEA, LETI MINATEC Campus, F-38054 Grenoble, France

Keywords: Security Knowledge Database, CAPEC, CVE, Vulnerability Assessment, Test Automation, LLM.

Abstract: Security knowledge databases represent key information in the process of vulnerability assessment and test automation of industrial products. The CVE and CAPEC databases respectively describe vulnerabilities and attack patterns. Linking a CVE entry to CAPEC can facilitate the generation of test plans, in the context of product test automation. Unfortunately, the great majority of CVE have no direct references to CAPEC. Several research works have focused on matching automatically CVE and CAPEC by computing text similarity on their descriptions, evaluating various models, in particular the term frequency inverse document frequency (TF-IDF) technique and transformer-based models such as SBERT. Depending on CVE description characteristics and evaluation criteria, these models are likely to perform differently by capturing different information types: vocabulary, preprocessing methods, context around words, etc. Hence, we propose a new classifier-based approach to select the most adapted similarity computation model from a given selection to match a CVE description with linked CAPEC descriptions. We evaluate this method on a recent set of CVE with CAPEC labels and show an improvement of matching accuracy compared to state-of-the-art methods leveraging a single model to compute text similarity. Our results also highlight the bias in the training and test set of CVE-CAPEC pairs.

1 INTRODUCTION

The common vulnerability exposure (CVE) (MITRE.org, 2024), and common attack pattern enumeration and classification (CAPEC) (MITRE.org, 2023) databases are interconnected resources (Valence, 2023) that provide comprehensive insights into vulnerabilities, their causes, and potential exploitation methods. To help in reproducing vulnerabilities, the CAPEC database references common attack patterns and provides detailed insights on potential exploits. Every CAPEC has an *"Execution Flow"* section, which describes how to *Explore*, *Experiment*, and *Exploit* a vulnerability. Linking CVE to CAPEC is not systematically done and is essential to automate the identification and management of vulnerabilities, i.e. with the objective for an organization to enhance its ability to prioritize and mitigate threats effectively. However, these security databases contain many entries (i.e. more than 270,000 CVE (NIST, 2025) and more than 500 CAPEC (MITRE.org, 2023)). Thus, the manual mapping between CVE and CAPEC can be time-consuming and error prone for product developers and testers.

Recent research has demonstrated the ability of Large Language Models (LLM) to solve Natural Language processing (NLP) tasks by learning complex, hierarchical representations of language data (Naveed et al., 2023). For example, LLM have been deployed in common vulnerability scoring system (CVSS) to assist CVE scoring during CVE registration (Shahid and Debar, 2021). More specifically, many research works (Kanakogi et al., 2021b; Kuppa et al., 2021; Das et al., 2022; **?**; Bonomi et al., 2025) propose text similarity computation methods based on transformer-based models like SBERT (sbert.net,) to identify correlations between CVE and CAPEC descriptions.

Since the mapping of CVE to CAPEC is not straightforward for the majority of CVE, as explored by Kanakogi et al. (Kanakogi et al., 2021a) and discussed in ICAR (Valence, 2023); it is difficult to evaluate the ability of a method to accurately identify matching CVE and CAPEC. Thus, the evaluation corpus must be comprehensive and diverse enough to adequately reflect real-world scenarios where the framework will operate.

Sauze-Kadar, M. and Loubier, T.
A Multi-Model Approach to Enhance Automatic Matching of Vulnerabilities to Attack Patterns.
DOI: 10.5220/0013555900003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 658-665
ISBN: 978-989-758-760-3; ISSN: 2184-7711

Consequently, matching vulnerabilities to attack patterns implies the following challenges:

- **Building a large and relevant corpus of labelled CVE and CAPEC**, considering a good coverage of CAPEC and CVE is difficult. The manual creation of a corpus is time consuming and error-prone.
- **State-of-the-art models to match CVE and CAPEC descriptions do not perform uniformly well** across different CVE and CAPEC pairs to match. The relevance and applicability of a model potentially varies depending on the CVE characteristics. Thus, adapting the model to CVE represents a promising approach to improve CVE and CAPEC descriptions matching accuracy.

Our contributions consist of the followings:

- **A novel multi-model framework**, which combines several Large Language Models (LLM) and Term Frequency-Inverse Document Frequency (TF-IDF) models to compute similarity between CAPEC and CVE descriptions. We show how our multi-model based method is able to improve CAPEC and CVE matching compared to approaches using a single model. **To the best of our knowledge, this is the first work to propose a framework combining several models to adapt CVE-CAPEC matching according to CVE description characteristics.**
- The exploitation of **a large evaluation corpus** of CVE and CAPEC, composed of 6,167 pairs of CVE-CAPEC. We build it automatically with recent CVE from 2021 to 2024, which have CAPEC labels through the *capecId* reference.

The remainder of this paper is structured as follows: Section 2 provides an overview of related work, while Section 3 introduces our multi-model method for CVE and CAPEC matching. In Section 4 we evaluate the performance of the proposed framework on a newly introduced large corpus of recent CVE. Finally, Section 5 concludes this work.

2 RELATED WORK

Table 1 summarizes literature research in the field of security databases mapping. We compare the different works with the following criteria:

- the approach to match text descriptions;
- the type of databases they apply to;
- the corpus size used for the evaluation.

Kanakogi et al. (Kanakogi et al., 2021a; Kanakogi et al., 2021b; Kanakogi et al., 2022) compare several LLM and TF-IDF models to match CVE and CAPEC by computing similarity of text descriptions. They suggest to use ensemble learning in future work to enhance matching accuracy (Kanakogi et al., 2021b), by adapting the matching model to CVE characteristics. Our multi-model method leverages the same similarity computation models they describe (Kanakogi et al., 2021a). But **to the best of our knowledge, this is the first work to introduce and evaluate a CVE-adaptive approach for matching CVE and CAPEC descriptions.** The authors define a corpus of 58 pairs of CVE and CAPEC based on the information contained under the CAPEC description of *Examples Instances* section. The corpus is rather small and data coverage is low compared to the corpus we introduce:

- 55 CVE: this represents less than 1% of the 275,000 CVE registered between 1999 and 2024.
- 40 CAPEC: this represents 7% of the 556 CAPEC available at the time of writing (2025).

Pan et al. (Pan et al., 2023) propose a new approach called Weighted-SBERT (WSBERT), which is also based on text similarity calculation to match CVE to CAPEC. Using a pretrained BERT-BiLSTM-CRF NER model, they assign more weight to specific words in order to focus on critical information. They use a corpus of 63 pairs of CVE and CAPEC, comparing their method to TF-IDF and usual SBERT models. Their framework demonstrates a stronger generalization and an ability to catch representative keywords with maximum semantic contribution, especially for long descriptions.

Das et al. introduce two frameworks for matching CVE to CAPEC. Firstly, they present V2W-BERT (Das et al., 2021), a transformer-based model designed specifically for mapping CVE to CWE. This model leverages the bidirectional encoding capability of BERT. Secondly, Das et al. propose VWC-MAP (Das et al., 2022) for mapping CWE to CAPEC leveraging text descriptions. For CWE to CAPEC mapping, they use a Text-to-Text model (Google T5 (Raffel et al., 2023)) for learning the relationships; i.e. given a CWE description as input, the model generates a text description, and finds the closest match within CAPECs. Both works propose a complete framework for linking CVE to CAPEC via CWE.

Bonomi et al. (Bonomi et al., 2025) enrich the input data of their matching framework compared to (Kanakogi et al., 2021b), by taking more CAPEC information into consideration: CAPEC *name*, *description*, *attack execution flow*, *mitigations*, *prerequisites*, and *resources*. They also enlarge the initial corpus of Kanakogi et al. with:

Table 1: Comparison of works in the field of security databases mapping.

Work	Approach	Mapping	Corpus size
(Kanakogi et al., 2021a)	TF-IDF / SBERT	CVE → CAPEC	58
(Kuppa et al., 2021)	MLTC	CVE → MITRE ATT&CK	200
(Shahid and Debar, 2021)	BERT	CVE → CVSS	45,926
(Das et al., 2022)	V2W-BERT / Google T5	CVE → CWE →CAPEC	170,000
(Pan et al., 2023)	Weighted-SBERT / SBERT / TF-IDF	CVE → CAPEC	63
(Bonomi et al., 2025)	ATTACKBERT Hyb & SBERT Hyb	CVE → CAPEC	223
This work	**Multi-model combination (TF-IDF and SBERT)**	**CVE → CAPEC**	**6,167**

- 60 representative pairs of CVE and CAPEC, using 7 CVE selected per year from 1999 to 2024;
- 223 CVE and CAPEC pairs, including 160 CVE and 118 CAPEC.

They use a hybrid approach using both SBERT and ATTACKBERT including keywords search, additionally to the similarity comparison, called SBERT Hyb and ATTACKBERT Hyb. They compare their results to the methodology adopted by Kanakogi et al. (Kanakogi et al., 2022): this approach improves the recall of the traditional SBERT models by being able to catch basic semantic similarity. It is particularly efficient to capture words context, compared to the approach by Kanakogi et al. They also propose to use keywords for filtering, as well as other CAPEC information.

Kuppa et al. (Kuppa et al., 2021) propose a Multi-Label Text Classification (MLTC) method for mapping CVE to ATT&CK (MITRE.org, 2023) techniques, using textual descriptions. Their framework is based on a multi-head joint embedding neural network architecture. Due to the lack of link between these 2 security databases, as demonstrated in (Valence, 2023), they propose a new unsupervised labeling technique, using the spaCy (spaCy, 2025) python library to extract context words around a CVE and ATT&CK descriptions. They create a labelled knowledge base of 200 CVE found in threat reports: it is composed of 150 attack scenarios exploiting vulnerabilities and 50 mitigation strategies to enrich CVE descriptions, covering 17 techniques. It helps learning attacker and defender views of a given CVE.

Shahid et Debar (Shahid and Debar, 2021) leverage recent advances in NLP to determine the CVSS vector and the associated severity score of a vulnerability. They trained BERT classifiers on 45,000 CVE and split train and test data to a 50% ratio. They prove that NLP applied to security databases could help developers and cybersecurity experts in better understanding CVE criticality.

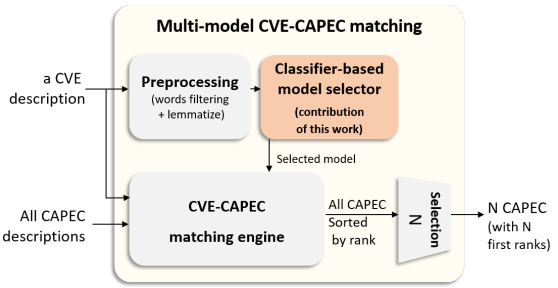

Figure 1: CVE-CAPEC matching framework architecture.

3 MULTI-MODEL CVE-CAPEC MATCHING FRAMEWORK

This section describes the multi-model framework for matching CVE and CAPEC descriptions. Section 3.1 provides an overview of the framework architecture, while Section 3.2 and Section 3.3 respectively focus on the matching model selection process and the matching engine, which are the main parts of the system.

3.1 Framework Architecture

The goal of the framework is to improve the state-of-the-art accuracy to match CAPEC to CVE. The matching process is described on Figure 1. We follow an approach similar to the one of Kanakogi et al. (Kanakogi et al., 2021a); the basic idea is to apply text similarity computation models to match CVE and CAPEC text descriptions.

While previous works focus on evaluating various similarity computation models such as TF-IDF and LLM, we propose a multi-model approach to combine several state-of-the-art similarity computation models to match CVE with CAPEC.The intuitive motivation for combining different models is to take advantage of each model's matching ability, since different models seem to perform differently depending on the input CVE description. Hence, we introduce a classifier based model selector to select the model to be ap-

Table 2: Preprocessing types.

Preprocessing type	Description
No preprocessing	Nothing is changed in the original description.
Basic preprocessing	Filtering of stop words (“the”, “of”, etc.) and punctuation.
Basic preprocessing and Lemmatize	Converts word into base form which considers the context.
Basic preprocessing and Stemming	Single words are replaced by root words. This is the preprocessing option which induces the smallest vocabulary.

plied for a given CVE description to identify linked CAPEC leveraging inherent CVE characteristics: this is the main contribution of this work.

The selected model is a parameter provided as input, together with the corresponding CVE description and all CAPEC descriptions of the corpus to the matching engine. The matching engine computes text similarity between the CVE and each CAPEC descriptions. It returns the list of CAPEC of the input corpus, which are ranked by decreasing correlation. The framework returns the N best CAPEC descriptions (i.e. with lowest rank), with $N \in \mathbb{N}^*$ a fixed parameter set by the user, considering the trade-off between output size and matching accuracy. Logically, the higher N (i.e. the more CAPEC in the output set), the higher the matching accuracy.

3.2 CVE-CAPEC Matching Model Selection from CVE Description

3.2.1 Preprocessing

We apply preprocessing on the input CVE description before it is manipulated by the model selector. First, we manually remove from the text description words which we consider useless in our context: e.g. “attack”, “adversary”, “moreover”, etc. Second, we lemmatize the description (Table 2); i.e. words are replaced by their base form.

We use non-normalized embeddings to represent preprocessed CVE descriptions; i.e. each description corresponds to a vector of the vocabulary size, where each feature corresponds to the count of word occurrences in the description. The vocabulary is defined on the training set: words in the test set with no occurrence in the training set are ignored.

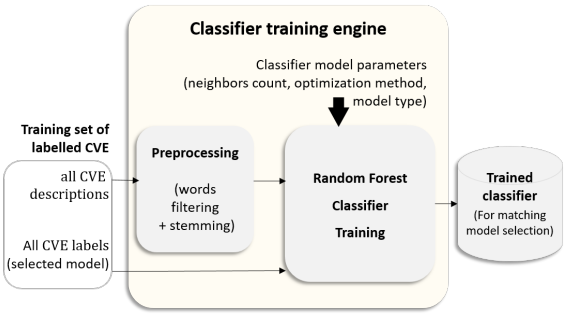

Figure 2: Classifier-based model selector training process.

3.2.2 Model Selector

The model selector takes a preprocessed CVE description as input. It returns the corresponding predicted label, which is the type of matching model to select.

For the training process, we define the label as the best model in the selection to match the input CVE description with linked CAPEC descriptions. Labels (i.e. the selected model for each CVE description) are defined in the training corpus as follows:

- For each matching model in scope, we compute the ranks of correct matching CAPEC for each CVE-CAPEC pair in the corpus.
- We set the selected model as the model inducing the lowest rank for the correct CAPEC among all matching models in scope.

The model selector is a random-forest based classifier, which we train following the process illustrated on Figure 2. Since the vocabulary is rather large, we apply dimensions reduction to reduce the number of features; we select features according to the k-nearest neighbors. We define parameters empirically, for the optimization method (`chi2`) and the number of neighbors k to consider (Section 4.2).

3.3 CVE-CAPEC Matching Engine

The multi-model framework leverages 10 text similarity computation models to match CVE with CAPEC. Each one of these 10 models corresponds to a specific label type of the modef selector. These 10 matching models can be divided in two main categories:

- **Large Language Models** (LLM), applied on text description without preprocessing. We test the 6 following LLM pretrained sentence transformers models, available from SBERT website (sbert.net,):
  - all-mpnet-base-v2
  - paraphrase-albert-small-v2

  - paraphrase-multilingual-mpnet-base-v2
  - multi-qa-mpnet-base-dot-v1
  - all-MiniLM-L12-v2
  - all-distilroberta-v1
- **Term Frequency–Inverse Document Frequency** (TF-IDF), was the method showing the best recall in previous work (Kanakogi et al., 2021a). We test 4 configurations applying TF-IDF on text description after preprocessing (Table 2):
  - no preprocessing and TF-IDF;
  - basic preprocessing and TF-IDF;
  - basic preprocessing, lemmatize, and TF-IDF;
  - basic preprocessing, stemming, and TF-IDF.

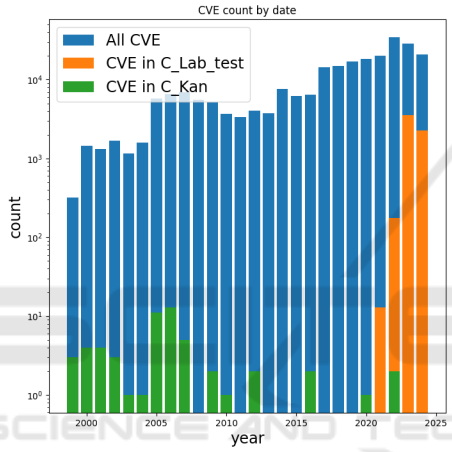

Figure 3: CVE count by year for evaluation corpuses.

4 EVALUATION OF THE MULTI-MODEL MATCHING FRAMEWORK

We describe the evaluation corpuses and metrics in Section 4.1. Experimental results related to CVE and CAPEC matching are in Section 4.2.

4.1 Experimental Setup

4.1.1 CAPEC and CVE Corpuses

We consider two corpuses of matching CAPEC and CVE for the evaluation:

- **Kanakogi et al. Database (Kanakogi et al., 2021a) (C^{Kan}):** this corpus consists of 58 pairs of matching CAPEC and CVE (i.e. $card(C^{Kan}) = 58$), which were set by Kanakogi et al. from the field "CVE examples" in the information of certain CAPEC. This corpus is not adapted to train the classifiers defined in Section 3 because of its small size. Thus, we use it as a test database for comparison with the state of the art.
- **all labelled CVE (C^{Lab}):** A new field "CAPEC ID" appears in CVE database from 2021 (according to CVE published date). Since then, many CVE have been labelled with CAPEC. We use this corpus for training classifiers since it is much bigger than C^{Kan}, i.e. $card(C^{Lab}) = 6,167$. The goal is to assess the relevance of our multi-model approach on a larger scale.

Note that the corpus C^{Kan} was built in 2021 and there is no overlap between both corpuses. Figure 3 shows the distribution of CVE by year for each case. Within the database of CVE (which contains 241,589 CVE at the time of writing), only 6,037 CVE have CAPEC labels. In the set of CVE in the corpus C^{Lab}, we consider CVE with links to CAPEC which are neither obsolete nor deprecated: this represents a total of less than 2.6% of the CVE database. In both corpuses, the relation between CAPEC and CVE is many-to-many. Considering all CAPEC and CVE pairs, the corpus C^{Lab} corresponds to 6,167 pairs. In C^{Lab}, we note a maximum of 19 CAPEC linked to a single CVE.

We split the corpus C^{Lab} into training C^{Lab}_{train} and test C^{Lab}_{test} set with a ration 80%-20%, so that $card(C^{Lab}_{train}) = 4,934$ and $C^{Lab}_{test} = 1,233$. We split CVE randomly between two sets so that we do not induce an additional bias in the training and test phases, i.e. based on CVE publication year for example.

The vocabulary used to represent preprocessed descriptions is defined over the training set C^{Lab}_{train}. For the test set, unknown words in the descriptions are ignored. Table 3 shows the relation between vocabulary size and preprocessing method.

Table 3: Vocabulary size (on C^{Lab}_{train})

Preprocessing type	Words count	
No preprocessing	11,426	(100%)
Basic preprocessing	11,299	(99%)
Basic preprocessing and lemmatize	10,693	(94%)
Basic preprocessing and stemming	9,464	(83%)

4.1.2 Evaluation Metrics

We use several metrics to test our multi-model approach. First, we consider the prediction accuracy to evaluate the ability of the classifier-based model selector to make correct predictions. For a given set of label predictions and corresponding actual labels, the

prediction accuracy is the number of predictions that exactly match the corresponding actual labels. We show the normalized ratio of prediction accuracy so that:

$$\text{prediction accuracy} = \frac{\text{count of all correct predictions}}{\text{count of all labels}} \quad (1)$$

Second, following the work of Kanakogi et al. (Kanakogi et al., 2021a), we leverage *recall@n* metric to evaluate the matching accuracy of the approach over a test corpus. Intuitively, *recall@n* represents the probability of having the correct matching CAPEC in the final output set of n CAPEC returned by the multi-model framework.

4.2 Experimental Results

4.2.1 Prediction Accuracy of the Model Selector

As described in Section 3.2.2, we apply k-nearest neighbors-based features selection to reduce the number of features of the random-forest classifier from the vocabulary size to k. Hence we test various k values between 1 and 9,464, which is the vocabulary size after preprocessing (Table 3); results are shown in Table 4. In the remainder of the experiments, we set $k = 1{,}500$, which corresponds to the best prediction accuracy observed on the test corpuses C^{Lab}_{test} and C^{Kan}.

The prediction accuracy is very high on the training corpus (higher than 95% compared to the results on test corpuses ($< 65\%$), with a tendency to overfitting for high values of k. A random prediction accuracy would be 10%, since the classifier defines 10 classes corresponding to the 10 matching models of the framework. In that regard, prediction accuracy on both test corpuses C^{Lab}_{test} and C^{Kan} is above this random reference in most of the configurations for $k \geq 100$. However, the model selector is much more accurate on the test corpus we introduced, C^{Lab}_{test}, than on the state-of-the-art corpus C^{Kan}.

Table 4: Prediction accuracy for the classifier-based model selector for various dimension reduction configurations (k features).

k	Prediction accuracy		
	C^{Lab}_{train}	C^{Lab}_{test}	C^{Kan}
20	0.62	0.61	0.07
50	0.82	0.61	0.07
100	0.90	0.61	0.14
500	0.96	0.63	0.19
1,000	0.96	0.64	0.12
1,500	0.97	0.64	0.22
2,000	0.97	0.64	0.19
2,500	0.97	0.64	0.12
3,000	0.97	0.64	0.17

4.2.2 Matching Accuracy of the Multi-Model Approach

As described in Section 3.1, we propose a multi-model based CVE-CAPEC matching approach, because different models perform differently depending on the pair of CVE and CAPEC to match. The *recall@n* results for both C^{Lab}_{test} and C^{Kan} corpuses are shown respectively on Figure 4 and on Figure 5. A focus on the results for C^{Lab}_{test} is provided in Table 5.

Our results validate the initial intuition of improving matching by combining various matching models: for both corpuses, the best case (ideal models combination, in red) shows a significantly higher *recall@n* rate compared to every tested single model.

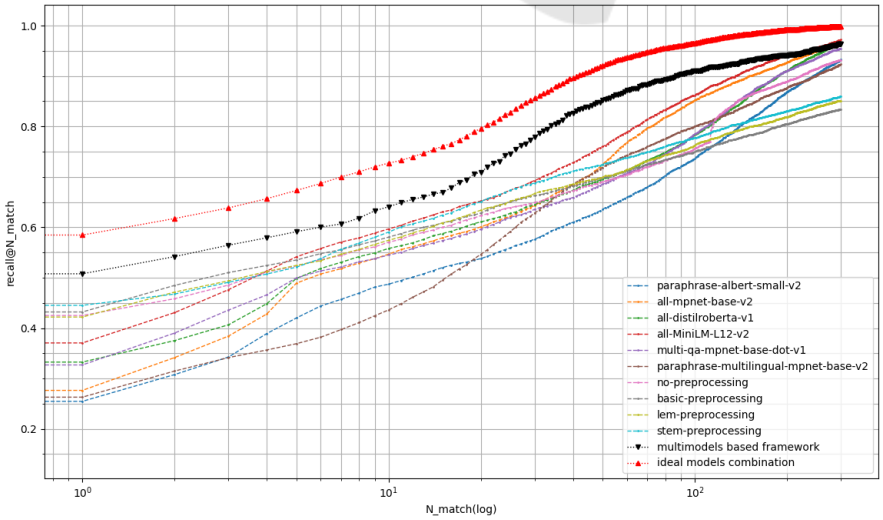

Figure 4: *recall@n* as a function of n for C^{Lab}_{test}.

Table 5: *recall@n* for different models on C_{test}^{Lab} corpus.

n	*recall@n*		
	all-MiniLM-L12-v2	multi-model based method	Ideal models combination
1	0.37	0.44	0.52
2	0.45	0.51	0.58
3	0.48	0.54	0.62
10	0.59	0.63	0.72
20	0.65	0.71	0.79
50	0.76	0.85	0.92
100	0.86	0.91	0.96
200	0.94	0.95	0.99
300	0.97	0.97	1.00

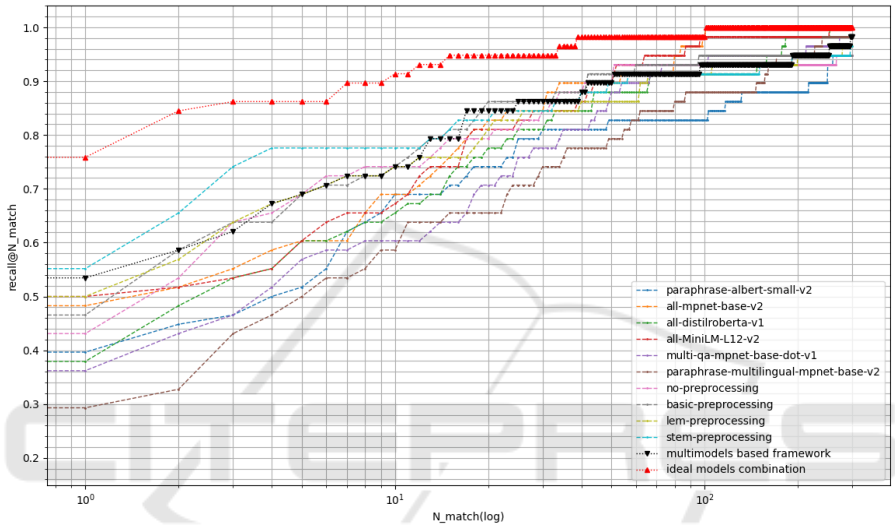

Figure 5: *recall@n* as a function of *n* for C^{Kan}.

After training the multi-model matching system on C_{train}^{Lab}, we show that it performs well on C_{test}^{Lab} (i.e. *recall@n* is higher than for any of the single models), but poorly on C^{Kan}: as seen on Figure 5, the model based on stemming and TF-IDF (as described in Section 3.3) performs comparatively better. As shown in Table 4, the reason for the difference of performance between both corpuses is caused by the poor prediction accuracy of the model selector (22% of accuracy for C^{Kan}, 64% for C_{test}^{Lab}). The degradation of inference for C^{Kan} means that there is a bias in the training corpus (C_{train}^{Lab}) or in the test corpus (C^{Kan}), i.e. the CVE in C^{Kan} are different from the CVE in C_{train}^{Lab}, which are used to train the classifier of the model selector.

5 CONCLUSION AND FUTURE WORK

We proposed a new classifier-based multi-model approach to select the most adapted similarity computation model from a given selection to match a CVE description with linked CAPEC descriptions. We evaluated the multi-model method on a recent set of CVE with CAPEC labels. We leveraged 10 models, from various LLM models and TF-IDF computation using several preprocessing approaches. We tested matching accuracy for every single model and our multi-model method. Our results show an improvement of matching accuracy compared to state-of-the-art methods leveraging a single model to compute text similarity: considering a number of 20 matching CAPEC, the correct CAPEC is in the output set in 71% of cases, 65% in the best case using one single model. Our results also highlight the bias in the training and test sets of CVE-CAPEC pairs: because of the poor accuracy of the classifier to select the model according to the CVE description, the matching accuracy decreases globally on a different state-of-the-art small test corpus introduced by Kanakogi et al.

A first open problem to address in future work is how to build a representative evaluation corpus of matching CVE and CAPEC, so that the multi-model framework is generic enough to be applied

on any CVE description. Building a representative corpus would require to understand biases in current databases in use. Second, in our current multi-model approach, we manually fix the number of best matching CAPEC to be returned by the framework. We propose to investigate how to reduce and adapt this number, filtering irrelevant CAPEC descriptions using meta information from CAPEC and CVE databases (e.g. abstraction level, keywords, etc.). Third, it would be interesting to study the practical integration of such CVE-CAPEC matching tool, which potentially induces matching errors, in a fully automated test generation process towards full test automation of industrial products.

ACKNOWLEDGEMENTS

The authors wish to express their gratitute to Raphael Collado, Maxime Lecomte, Ulysse Vincenti, Victor Breux, and Lalie Arnoud for the helpful discussions and to the anonymous reviewers for their useful comments. The work presented in this paper was funded by the "France 2030" government investment plan managed by the French National Research Agency, under the reference "ANR-22-PECY-0005".

REFERENCES

Bonomi, S., Ciavotta, A., Lenti, S., and Palma, A. (2025). Beyond the Surface: An NLP-based Methodology to Automatically Estimate CVE Relevance for CAPEC Attack Patterns. arxiv (https://arxiv.org/abs/2501.07131), Version Number: 1.

Das, S. S., Dutta, A., Purohit, S., Serra, E., Halappanavar, M., and Pothen, A. (2022). Towards automatic mapping of vulnerabilities to attack patterns using large language models. In *IEEE International Symposium on Technologies for Homeland Security (HST)*.

Das, S. S., Serra, E., Halappanavar, M., Pothen, A., and Al-Shaer, E. (2021). V2w-bert: A framework for effective hierarchical multiclass classification of software vulnerabilities. https://arxiv.org/abs/2102.11498.

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A., and Yoshioka, N. (2021a). Tracing CAPEC Attack Patterns from CVE Vulnerability Information using Natural Language Processing Technique.

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A., and Yoshioka, N. (2021b). Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques. volume 12, page 298.

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A., and Yoshioka, N. (2022). Comparative evaluation of nlp-based approaches for linking capec attack patterns from cve vulnerability information. In *Applied Sciences*.

Kuppa, A., Aouad, L., and Le-Khac, N.-A. (2021). Linking CVE's to MITRE ATT&CK Techniques. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–12, Vienna Austria. ACM.

MITRE.org (2023). CAPEC - Common Attack Pattern Enumeration and Classification (CAPEC™). https://capec.mitre.org/.

MITRE.org (2024). MITRE ATT&CK®. https://attack.mitre.org/.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2023). A Comprehensive Overview of Large Language Models. arXiv (https://arxiv.org/abs/2307.06435), Version Number: 10.

NIST (2025). NVD - Vulnerabilities. https://nvd.nist.gov/vuln.

Pan, M., Li, B., Zou, Y., Yang, W., Wang, D., and Zhang, T. (2023). Tracing Vulnerability to Attack Patterns Using Text Similarity. In *2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 1330–1334, Changchun, China. IEEE.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2023). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs] (http://arxiv.org/abs/1910.10683).

sbert.net. Pretrained Models — Sentence Transformers documentation. https://www.sbert.net/docs/sentence_transformer/pretrained_models.html.

Shahid, M. and Debar, H. (2021). CVSS-BERT: Explainable Natural Language Processing to Determine the Severity of a Computer Security Vulnerability from its Description. arxiv (https://arxiv.org/abs/2111.08510) Version Number: 1.

spaCy (2025). spaCy · Industrial-strength Natural Language Processing in Python. https://spacy.io/.

Valence, A. (2023). ICAR, a categorical framework to connect vulnerability, threat and asset managements.