applied sciences

MDPI

Article

Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information

Kenta Kanakogi [1,*], Hironori Washizaki [1], Yoshiaki Fukazawa [1], Shinpei Ogata [2], Takao Okubo [3], Takehisa Kato [4], Hideyuki Kanuka [4], Atsuo Hazeyama [5] and Nobukazu Yoshioka [6]

1 Department of Computer Science and Engineering, Waseda University, Shinjuku-ku, Tokyo 169-8555, Japan; washizaki@waseda.jp (H.W.); fukazawa@waseda.jp (Y.F.)
2 Institute of Engineering, Academic Assembly, Shinshu University, Nagano 380-8553, Japan; ogata@cs.shinshu-u.ac.jp
3 Institute of Information Security, Yokohama 221-0835, Japan; okubo@iisec.ac.jp
4 Hitachi, Ltd., Chiyoda-ku, Tokyo 100-8280, Japan; takehisa.kato.wx@hitachi.com (T.K.); hideyuki.kanuka.dv@hitachi.com (H.K.)
5 Department of Information Science, Tokyo Gakugei University, Koganei-shi 184-8501, Japan; hazeyama@u-gakugei.ac.jp
6 Research Institute for Science and Engineering, Waseda University, Shinjuku-ku, Tokyo 169-8555, Japan; nobukazuy@acm.org
* Correspondence: kanakogi-soft@fuji.waseda.jp

Abstract: Vulnerability and attack information must be collected to assess the severity of vulnerabilities and prioritize countermeasures against cyberattacks quickly and accurately. Common Vulnerabilities and Exposures is a dictionary that lists vulnerabilities and incidents, while Common Attack Pattern Enumeration and Classification is a dictionary of attack patterns. Direct identification of common attack pattern enumeration and classification from common vulnerabilities and exposures is difficult, as they are not always directly linked. Here, an approach to directly find common links between these dictionaries is proposed. Then, several patterns, which are combinations of similarity measures and popular algorithms such as term frequency–inverse document frequency, universal sentence encoder, and sentence BERT, are evaluated experimentally using the proposed approach. Specifically, two metrics, recall and mean reciprocal rank, are used to assess the traceability of the common attack pattern enumeration and classification identifiers associated with 61 identifiers for common vulnerabilities and exposures. The experiment confirms that the term frequency–inverse document frequency algorithm provides the best overall performance.

Keywords: cybersecurity database; CVE; CAPEC; natural language processing; sentence embeddings; TF-IDF; universal sentence encoder; sentence BERT

Citation: Kanakogi, K.; Washizaki, H.; Fukazawa, Y.; Ogata, S.; Okubo, T.; Kato, T.; Kanuka, H.; Hazeyama, A.; Yoshioka, N. Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information. *Appl. Sci.* **2022**, *12*, 3400. https://doi.org/10.3390/app12073400

Academic Editor: Vito Conforti

Received: 12 February 2022
Accepted: 22 March 2022
Published: 27 March 2022

1. Introduction

System administrators devote a significant amount of time to vulnerability management against cyberattacks due to the sheer volume of vulnerabilities. Information must be collected quickly and exactly for efficient vulnerability management. Effective management should provide information not only about known vulnerabilities, but also possible attacks. A database on cybersecurity issues can gather such information. There are two publicly available databases: Common Vulnerabilities and Exposures (CVE) [1] and Common Attack Pattern Enumeration and Classification (CAPEC) [2]. CVE is a dictionary specializing in vulnerability information. It lists and assigns a unique number to each vulnerability or defect in a system, software, or web application in information security. CAPEC is a dictionary of attack information. It systematizes attacks and exploits against vulnerabilities.

There is a vulnerability scanner tool called Vuls. Although it can automatically find CVE-IDs, CVEs are poorly informed about attacks. Consequently, it needs to be supplemented with information about CAPEC attack patterns. The associated CAPEC-ID from

CVE cannot be directly identified from the associated CAPEC-ID since CVE and CAPEC are independent. Currently, CVE can be traced to CAPEC via common weakness enumeration (CWE) [3], which is a community-developed list of software and hardware weakness types. CVE is synchronized with the national vulnerability database (NVD) [4], and the associated CWE-ID is found from the NVD. CWE contains the associated CAPEC-identifier (ID). Therefore, the conventional method traces from CVE through CWE to CAPEC. There are two major issues with the conventional method:

- Even with CWE, it may not be possible to trace the CVE-ID to the associated CAPEC-ID.
- Cybersecurity databases are linked manually. The growing amount of vulnerability information makes manual handling problematic, resulting in more failures.

An explicit link from CVE to CAPEC currently does not exist. A CVE reporter should include the associated CAPEC-ID. However, accurate linking is costly and difficult. This paper aims to trace associated CAPEC-IDs directly from the CVE-ID. Specifically, a list of associated CAPEC-ID candidates is generated for a given CVE-ID. Then, the linkage is determined based on the similarity between the CAPEC document and the CVE description. This paper is an extension of papers presented at the Hawaii International Conference on System Sciences (HICSS 54) [5] and Information for Business and Management-Software Development for Data Processing and Management [6]. Here, we extend the experimental patterns and revised the analysis results and the corresponding discussion.

Nine patterns are proposed and evaluated to calculate the similarity. Three different similarity measurement algorithms are used: term frequency–inverse document frequency (TF-IDF) [7], universal sentence encoder (USE) [8], and sentence BERT (SBERT) [9]. These algorithms can be classified as context-independent algorithms or context-dependent algorithms. TF-IDF is a context-independent algorithm, which calculates similarity based on the occurrence frequency of words. It is a classical and simple method. On the other hand, USE and SBERT are context-dependent algorithms. They learn distinct embeddings of the same word in different contexts and create a model. BERT is a relatively new technology. Previous studies [10,11] have employed similarity algorithms to achieve similar goals, but similarity algorithms have yet to be accurately evaluated and compared. This study compares 9 patterns of the proposed approach using the inputs of 61 patterns of CVE-IDs and tracing the associated CAPEC-IDs.

This paper addresses the following three research questions (RQs):

RQ1. How accurately can a CVE-ID be traced to its associated CAPEC-ID following a link between cybersecurity databases? This question investigates the accuracy of tracing CVE-IDs to the associated CAPEC-IDs using the conventional method and identifies the issues with the conventional method.

RQ2. How accurate is the tracing of CVE-IDs to the associated CAPEC-IDs when using the proposed approach? This question evaluates the usefulness of the similarity algorithms.

RQ3. Which algorithm provides the most suitable tracing? This question clarifies the algorithm that is most appropriate for this task.

This paper makes the following contributions. First, the link accuracy between cybersecurity databases is elucidated. Second, candidate links of CAPEC-IDs are easily identified, aiding in the linking process. Finally, CVE reporters can determine whether the report contains sufficient vulnerability information.

The rest of this paper is organized as follows. Section 2 introduces related works, a motivating example, and current problems. Section 3 details the proposed approach and the different patterns. Section 4 describes the experiments and discusses the RQs. Finally, Section 5 presents our conclusions and future work.

2. Related Work and Problems

Herein, the related works are introduced. Additionally, a motivating example and current problems are provided.

2.1. Related Work

Previous studies have investigated the mapping between cybersecurity databases [8,9,12,13]. The aims of [8,9] are similar to this research. CVE is mapped to CAPEC using TF-IDF [8] and Doc2Vec [9]. However, [8] focuses on limited vulnerabilities. The previous studies did not provide a valid evaluation or a comparison of different algorithms. On the other hand, this study uses the dataset defined by MITRE to conduct experiments comparing multiple algorithms.

Recently, research on mapping of ATT&CK [14] and CVE [15–17] has intensified. The approach in this study should be applicable to map CVE and ATT&CK. In the future, the tracking will be expanded to include ATT&CK.

The authors of [18,19] propose a method to combine similarity algorithms. In [18], a hybrid method, combining Doc2Vec weighted by TF-IDF and a vector space model weighted by TF-IDF, is proposed. Here, hybrid metaheuristic and machine learning methods are considered as a growing research domain [20–23] extract hidden topics from the textual description of each attack pattern. Although the approach in this study uses a simple similarity algorithm, this work builds upon previous studies.

Some studies have used cybersecurity databases to create vulnerability ontology models [24–26] and to analyze and assess risk and security [27–32]. However, they do not describe the information retrieval process from cybersecurity databases. Others have focused on mining methods and information retrieval from each cybersecurity database [33–37]. In particular, [33,35] introduce information retrieval processes by following the relationships among cybersecurity databases. Unlike previous studies, this study explicitly evaluates the accuracy of the links between databases.

2.2. Motivating Example and Problems

Following the links between cybersecurity databases, CVE-ID can be traced to CAPEC-IDs. In some cases, tracing back to CAPEC is not possible. An example is CVE-2020-10108, which is a vulnerability related to HTTP request splitting. The description of CVE-2020-10108 is as follows: In Twisted Web through 19.10.0, there was an HTTP request splitting vulnerability. When presented with two content-length headers, it ignored the first header. When the second content-length value was set to zero, the request body was interpreted as a pipelined request [38].

There is an attack pattern identifier for HTTP request splitting in CAPEC-105. CVE-2020-10108 is linked to CWE-20, but CAPEC-105 is not. Therefore, CWE-20 cannot be traced from CVE-2020-10108 to CAPEC-105. The exact number of CVE-IDs that cannot be traced to CAPEC via CWE is unknown. Since the issue is the link between cybersecurity databases, it is preferable to directly trace from CVE to CAPEC. The motivation of this study is to create an approach to directly trace from CVE to CAPEC.

3. Tracing Method from CVE-ID to CAPEC-ID

The proposed approach involves four steps. First, a corpus of CVE descriptions and 546 CAPEC-ID documents are generated. Figures 1–3 show examples using CVE-2020-10108 as the input data. Second, a document embedding is created using a similarity algorithm. The document measure in Figure 1 created 547 vectors, while the section measure in Figures 2 and 3 created 2692 vectors. Third, the CVE-ID vector and all CAPEC vectors are used to calculate the cosine similarity (Equation (1)). Finally, the CAPEC documents are sorted by the similarity score and N CAPEC-IDs are selected. Then, candidates for the associated CAPEC-IDs are obtained from CVE-IDs.

$$\cos\left(\vec{p}\cdot\vec{q}\right) = \frac{p_1q_1 + p_2q_2 + \ldots + p_nq_n}{\sqrt{{p_1}^2 + {p_2}^2 + \ldots + {p_n}^2}\sqrt{{q_1}^2 + {q_2}^2 + \ldots + {q_n}^2}} = \frac{\vec{p}\cdot\vec{q}}{\left|\vec{p}\right|\left|\vec{q}\right|} \quad (1)$$

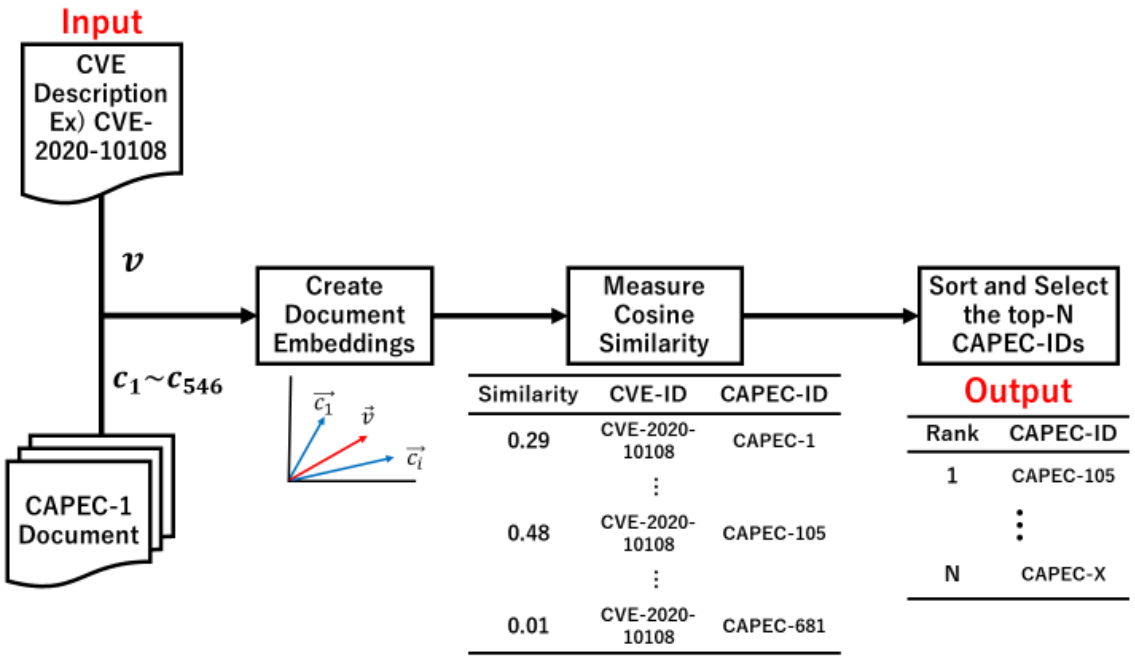


Figure 1. Overview of the proposed approach using the entire document.

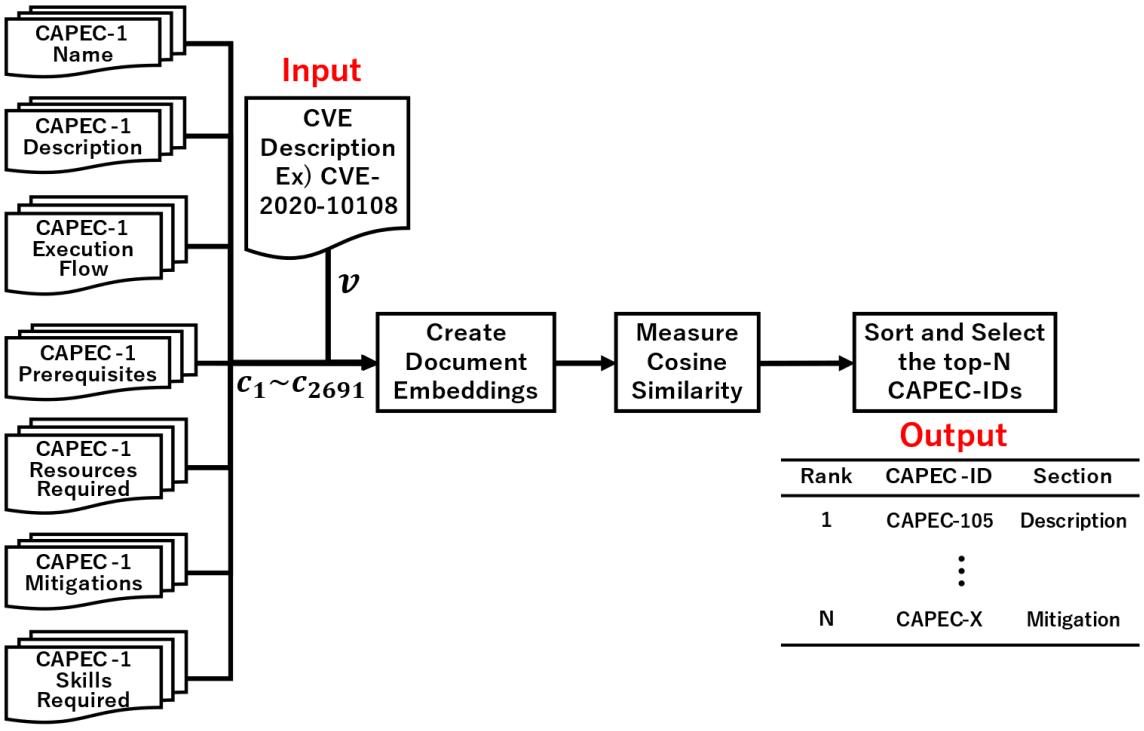


Figure 2. Overview of the proposed approach using all sections.

Three similarity measures are proposed to trace CAPEC from CVE. The first one measures the similarity of all sections as a single document (Figure 1). The second measures the similarity for each section (Figure 2). The third measures the similarity for each section and then calculates the average of the similarity for all sections per CAPEC-ID (Figure 3).

In addition, three algorithms are considered to create document embeddings: TF-IDF, USE, and SBERT. Below is an explanation for finding the associated CAPEC-ID using CVE-2020-10108 as the input data. All algorithms are demonstrated using the flow of tracing in combination with the document patterns.

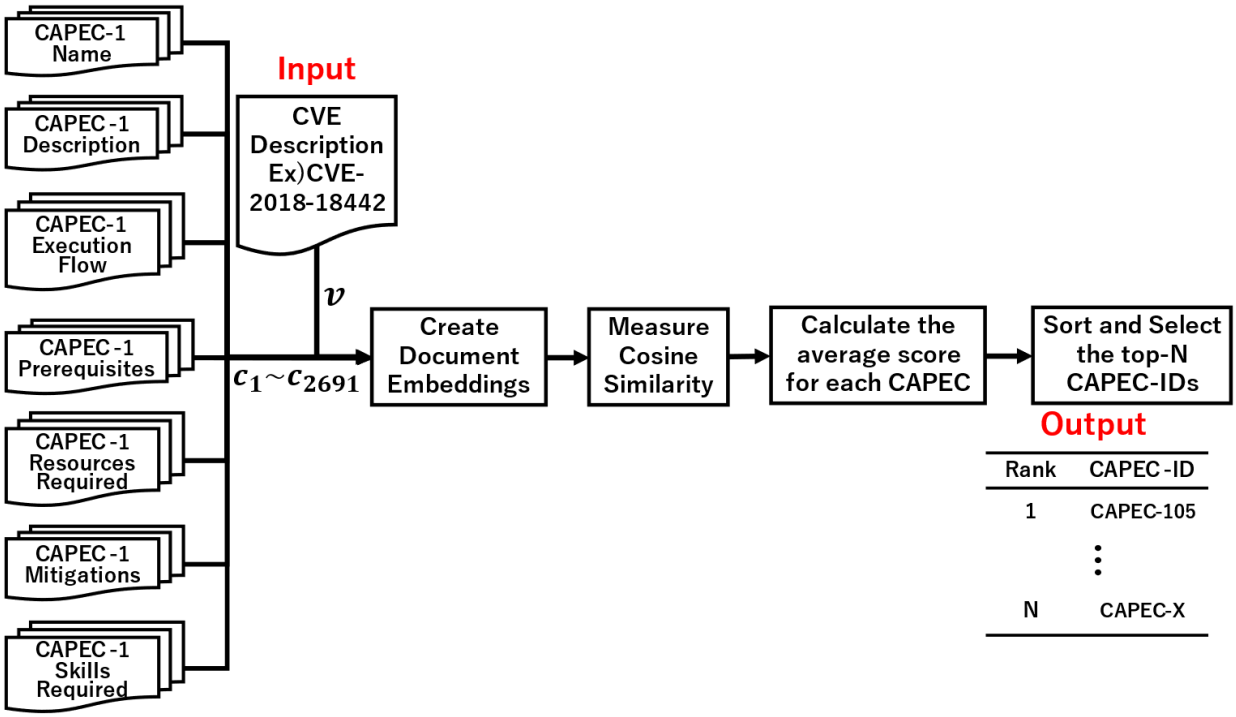


Figure 3. Overview of the proposed approach using the section average.

3.1. Tracing Based on TF-IDF

TF-IDF evaluates the importance of words in a document. The TF-IDF score is obtained by multiplying the term frequency and the inverse document frequency. Here, TfidfVectorizer from scikit-learn [39] is employed. TfidfVectorizer converts each document into a vector based on the TF-IDF score when given a set of documents. Figure 4 shows the approach using TF-IDF as the algorithm with CVE-2020-10108 as the input data.

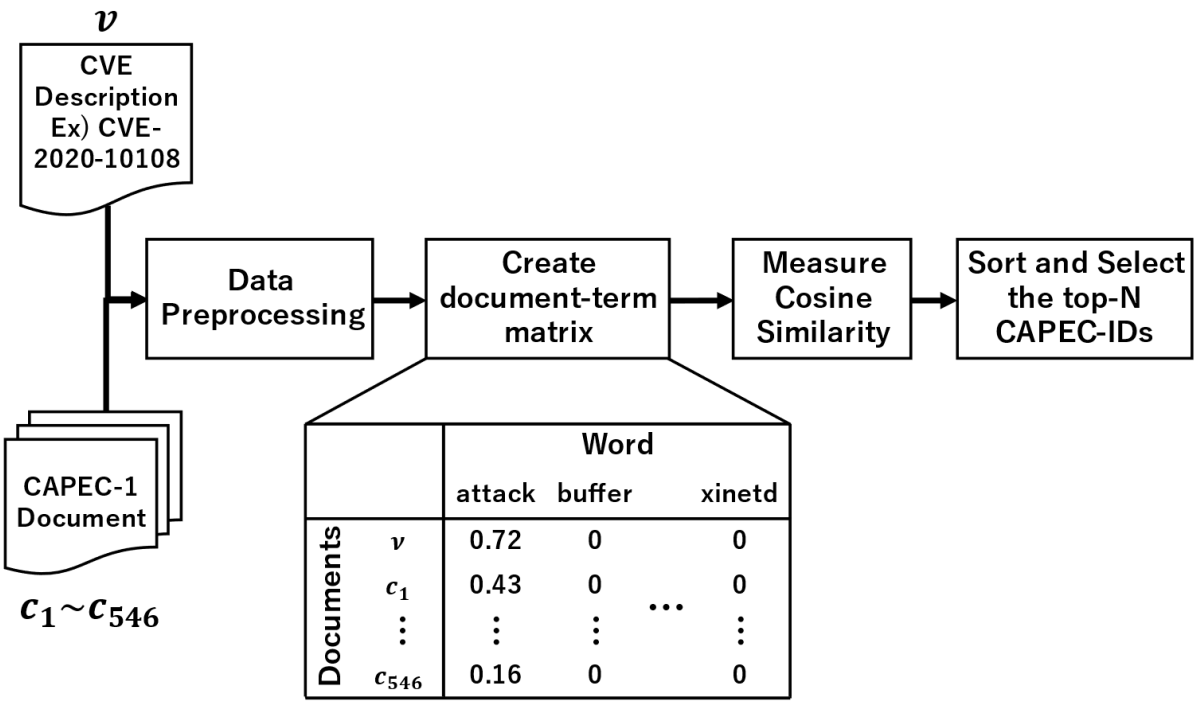


Figure 4. Overview of the tracing based on the TF-IDF algorithm.

3.2. Tracing Based on USE

USE sums the vectors of each word considering the context and normalizes them by the length of the sentence to obtain a vector of sentences. There are two methods to obtain context-sensitive word vectors: the transformer encoder and the deep averaging network (DAN). Both methods take a paragraph as the input and output a 512-dimensional vector. Pre-trained models are available on the Tensorflow Hub for both methods. This study employs DAN because transformers are used in SBERT. Figure 5 shows the approach using the USE algorithm with CVE-2020-10108 as the input data. The pre-training model is the universal-sentence-encoder-v4 [40].

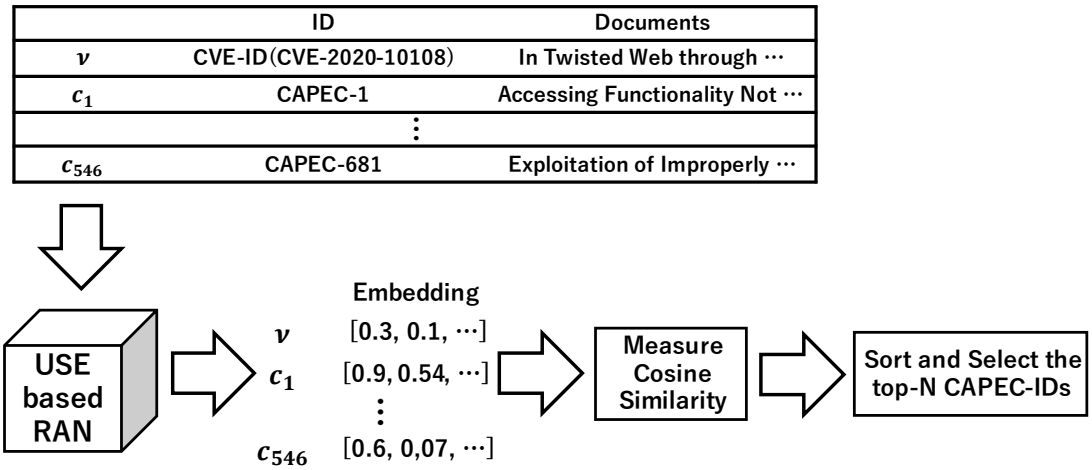


Figure 5. Overview of the tracing based on the Universal Sentence Encoder (USE) algorithm.

3.3. Tracing Based on SBERT

SBERT uses a pre-trained BERT model and Siamese Network to create highly accurate sentence vectors. SBERT adds a layer to the output layer of BERT to perform a pooling operation. The output of BERT is a sequence of variable numbers of embedded vectors. Each vector corresponds to the tokens that make up a sentence. The pooling operation converts the sequence of variable vectors into a vector of one fixed-length dimension. SBERT uses a Siamese network for fine-tuning. The loss function is important for fine-tuning. There are 13 different loss functions, but the appropriate one depends on the training data and the target task. Here, CosineSimilarityLoss is employed because it is easy to prepare. In-house data is used for fine-tuning. The experiments use only the pre-training model to ensure the presence of failed data. The guessed security data is fine-tuned so that the failed data can be successfully traced. Figure 6 shows the approach using SBERT with CVE-2020-10108 as the input data. The pre-training model is all-distilroberta-v1 [41].

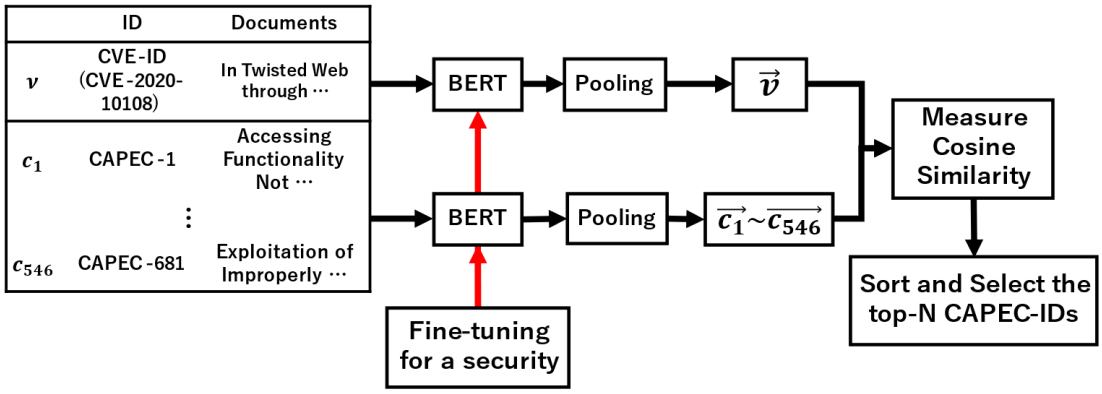


Figure 6. Overview of the tracing based on the Sentence BERT (SBERT) algorithm.

4. Experiments and Results

4.1. 61 CVE-IDs

CAPEC has the Example Instance field. Figure 7 shows the Example Instance for CAPEC-60, where CVE-1999-0428 and CVE-2002-0258 are listed. This field has a total of 61 CVE-IDs. Although three are duplicates, they were recognized as different CVE-IDs in the experiment. The experiment assumed that the link from CVE to CAPEC was many-to-one and all 61 CVE-IDs were used. The experiment aimed to verify that the CVE-ID listed in the example instance field can be traced to the corresponding CAPEC-ID.

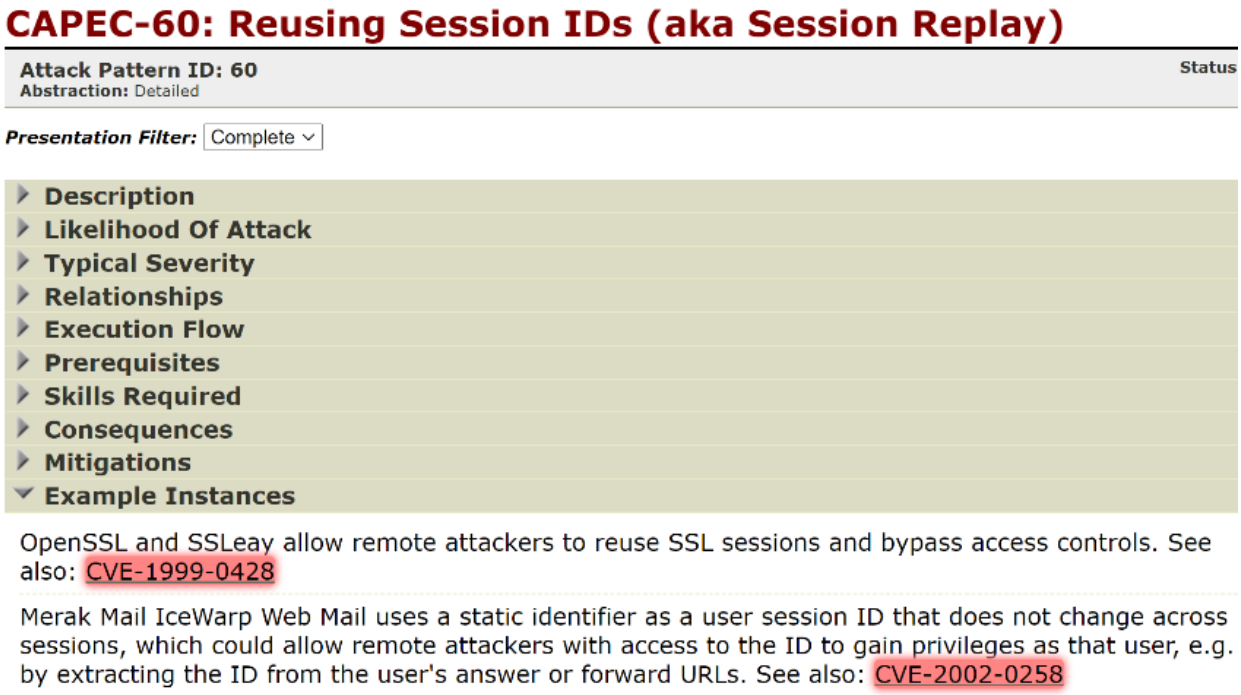


Figure 7. CAPEC-60 web page.

4.2. Experimental Patterns

The experiment employed nine experimental patterns: three algorithms combined with three measurement methods (Table 1). Because the BERT input is limited to 512 tokens, it cannot be used for the pattern shown in Figure 1 (document). Hence, the experiment only considered eight patterns.

Table 1. Experiment pattern.

	Document	Per Section	Section Average
TF-IDF	○	○	○
USE	○	○	○
SBERT	×	○	○

4.3. Metrics

The mean reciprocal rank (MRR) and Recall@n were used to evaluate the experimental results. MRR and Recall@n are popular evaluation metrics used in citation recommendation systems. Recall@n indicates the proportion of relevant items found in the top N recommendations. That is, it denotes the percentage of the 61 CVEs that are successfully traced. It is given as

$$Recall@N = \frac{|a \cap p_N|}{|a|} \quad (2)$$

where N is the number of top rankings to consider. a is the set of correct answer data. p_N is the top N recommendation list.

MRR represents the average value of the reciprocal rank of the correct entity for each prediction task. It is expressed as

$$MRR = \frac{1}{|N|} \sum_{i=1}^{|N|} \frac{1}{rank_i} \tag{3}$$

where N is the number of test data. $rank_i$ is the rank position of the first relevant document for the i-th query.

4.4. Results

Table 2 and Figure 8 show the results for the eight experimental patterns.

Table 2. Results for the eight patterns using the proposed approach.

	Recall@10	MRR
TF-IDF (document)	0.787	0.591
SBERT (section average)	0.803	0.368
TF-IDF (per section)	0.770	0.596
TF-IDF (section average)	0.754	0.470
USE (document)	0.705	0.491
USE (per section)	0.672	0.420
USE (section average)	0.590	0.474
SBERT (per section)	0.557	0.337

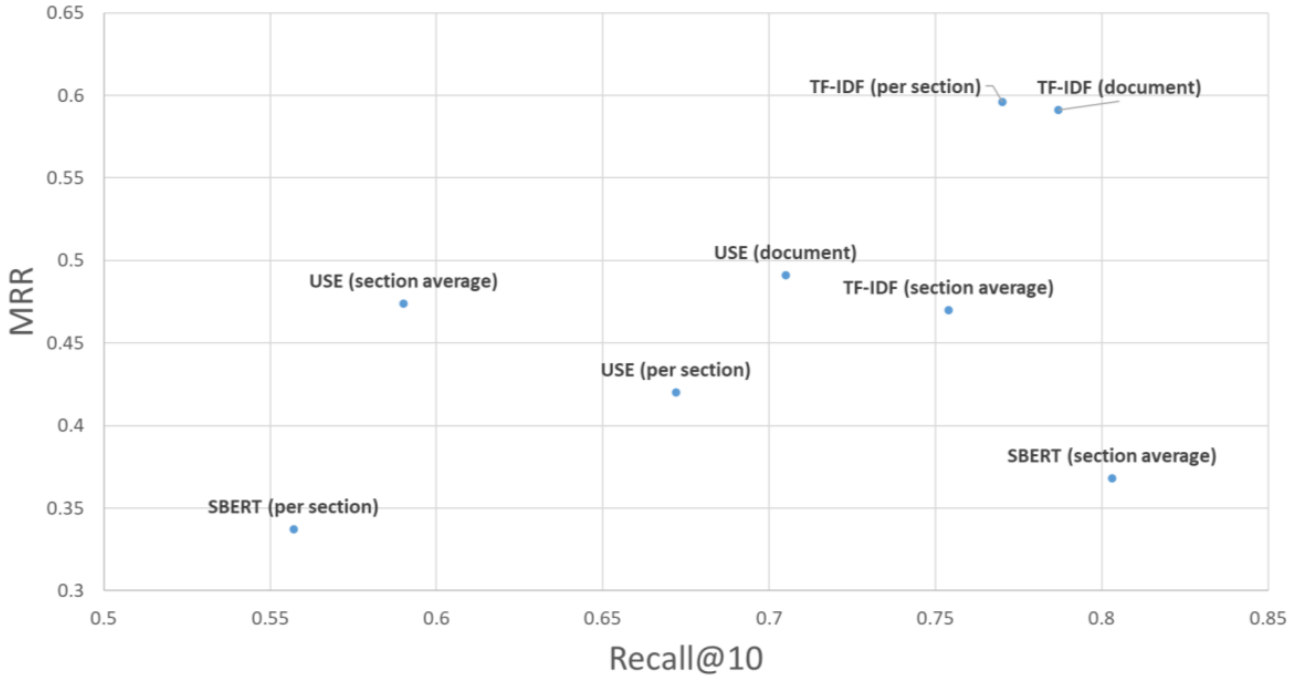


Figure 8. Scatter plots of the results for the eight patterns using the proposed approach.

4.5. RQ1 How Accurately Can a CVE-ID Be Traced to Its Associated CAPEC-ID following a Link between Cybersecurity Databases?

Of the 61 CVE-IDs, only 4 were successfully traced. This low accuracy is due to the CVE-CWE link. First, some of the identified CVE-IDs are not linked to the CWE. Figure 9 shows the percentage of CVE-IDs that are linked to the CWE by year. Although 80% have been linked since 2008, 20% have not. This 20% cannot be traced to the CWE.

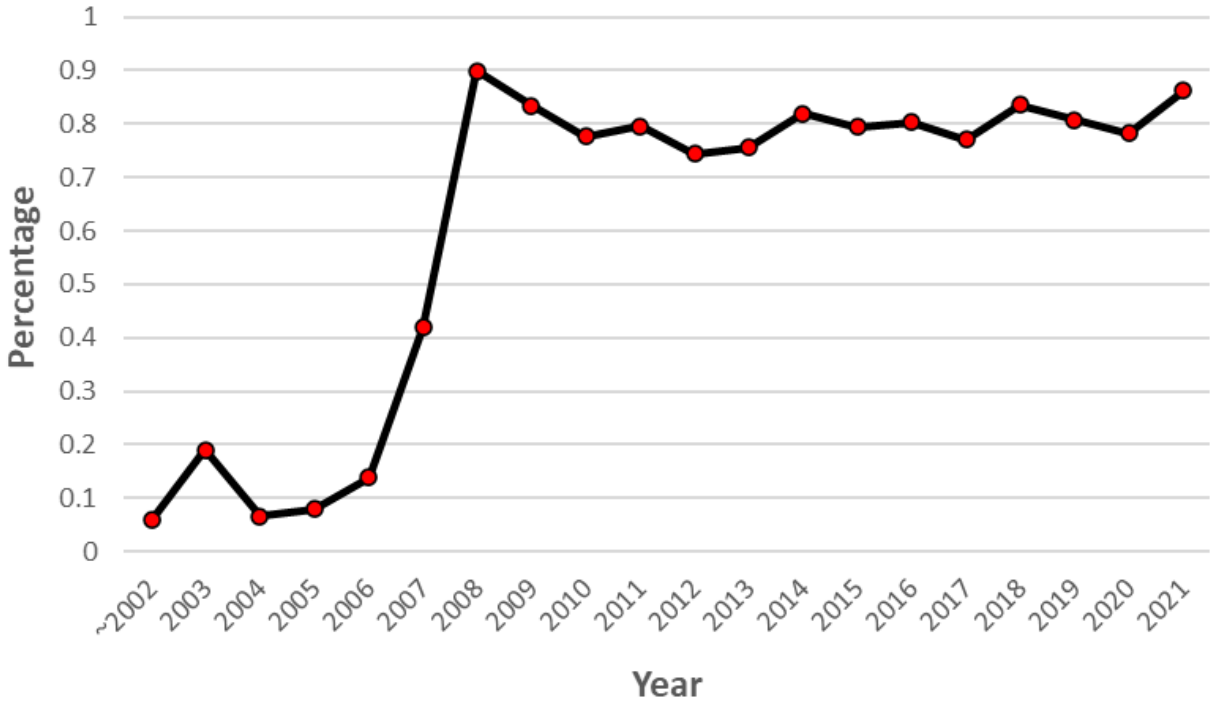


Figure 9. Changes in the percentage of CVE-ID linked to CWE-ID by year.

Second, many CVE-IDs are linked to CWE-IDs with a high level of abstraction, such as CWE-20 and CWE-200. Figure 10 shows the frequency that highly abstract CWE-IDs are linked to each other. CWE-20 and CWE-200 are linked with high frequency. Highly abstract CWE-IDs generate two issues. First, the underlying weaknesses become invisible, and the chain and complex relationships are hidden. CWE-20 has the following description:

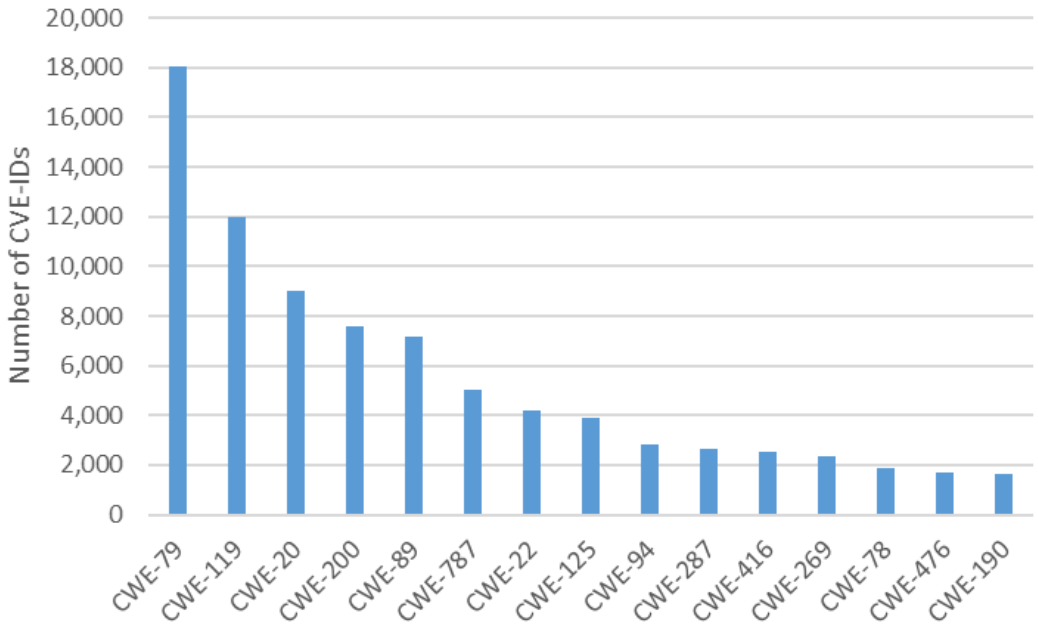


Figure 10. Vulnerability distribution by CWE-ID.

The "input validation" term is extremely common, but it is used in many different ways. In some cases, its usage can obscure the real underlying weakness or otherwise hide chaining and composite relationships [42].

Based on the above description, a highly abstract CWE-ID may not identify the root cause or trace the attack. Second, one CWE-ID is associated with many CAPEC-IDs. CWE-20 is linked to 51 CAPEC-IDs and CWE-200 is linked to 58 CAPEC-IDs. Hence, it is difficult to identify the correct CAPEC-ID. The relationship between CVE-ID and CWE-ID is not always considered as a characteristic of vulnerability usage by attackers.

RQ1. Answer

Only 4 out of the 61 CVE-IDs were traced to the associated CAPEC-ID. CAPEC was not taken into account in the link between CVE and CWE. Therefore, following CVE to the CAPEC attack information may not provide useful information.

4.6. RQ2 How Accurate Is the Tracing of CVE-IDs to the Associated CAPEC-IDs When Using the Proposed Approach?

Table 2 shows that the "document" pattern is better suited for TF-IDF and USE. The "section average" pattern is most suitable for SBERT. Figures 11–13 show the experimental results for these three patterns. Figure 14 shows a box-and-whisker diagram of average precision. All algorithms traced more than 70% of the CVE-IDs. The experiment involved only one ground truth because the link between CVE and CAPEC is a many-to-one link. This is the reason for the low precision. Changing the criteria of the ground truth impacts the value of precision. The number of words in the CVE description and CAPEC document are not correlated with the success rate of the proposed approach. A characteristic of CVE-IDs that failed was an insufficient CVE description. This is due to missing cybersecurity words important to trace or hidden fundamental weaknesses. Figure 15 shows the results of a Kruskal–Wallis test. The *p*-value less than 0.05 indicates a significant difference.

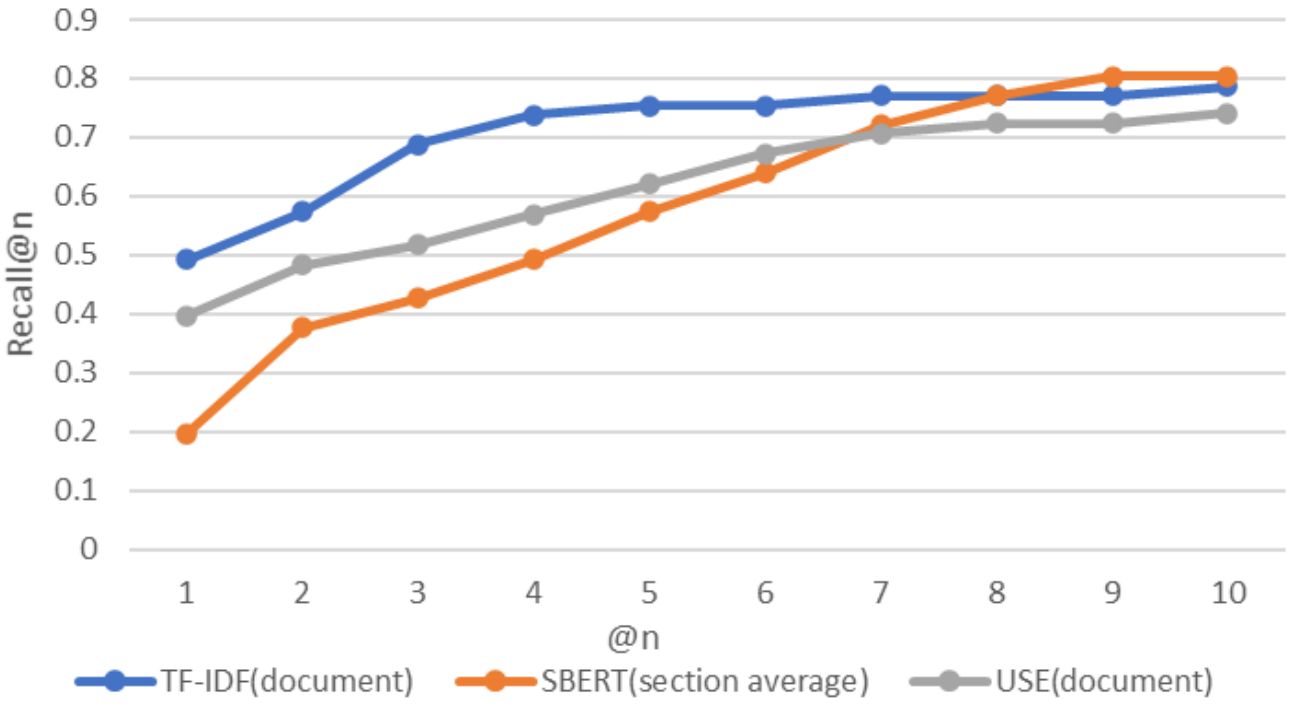


Figure 11. Recall for each approach.

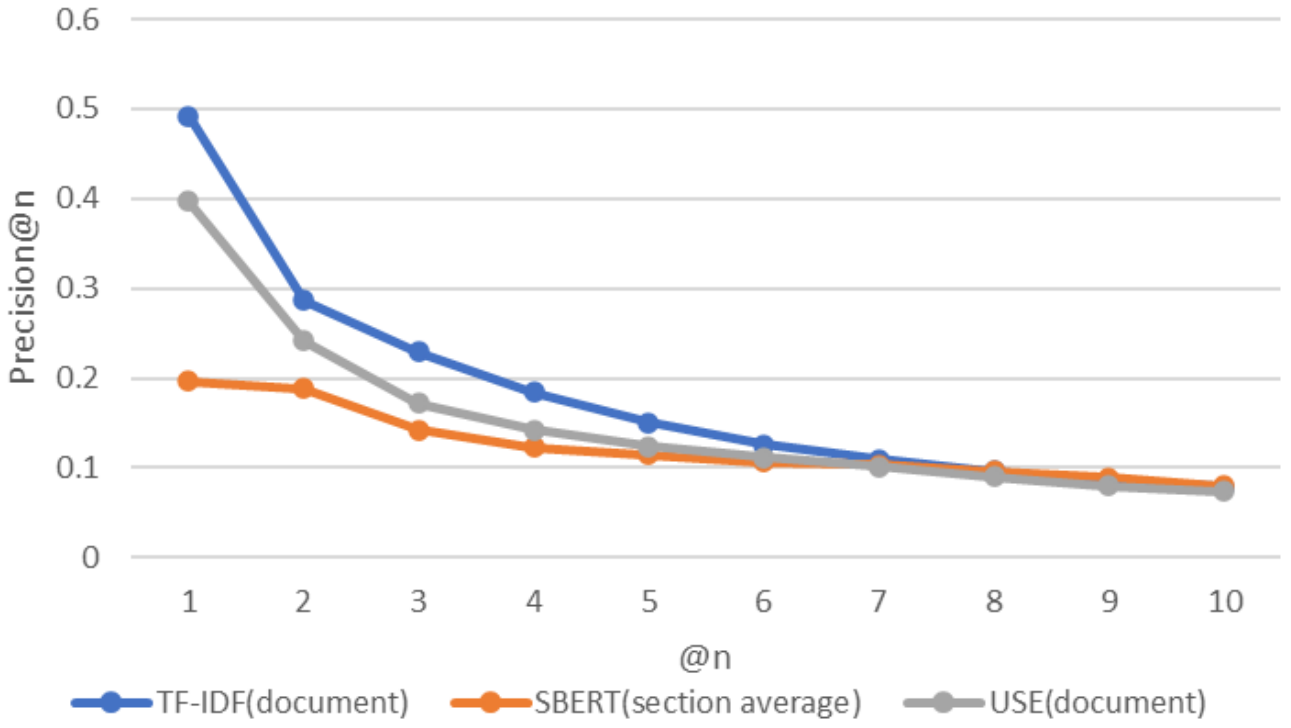


Figure 12. Precision for each approach.

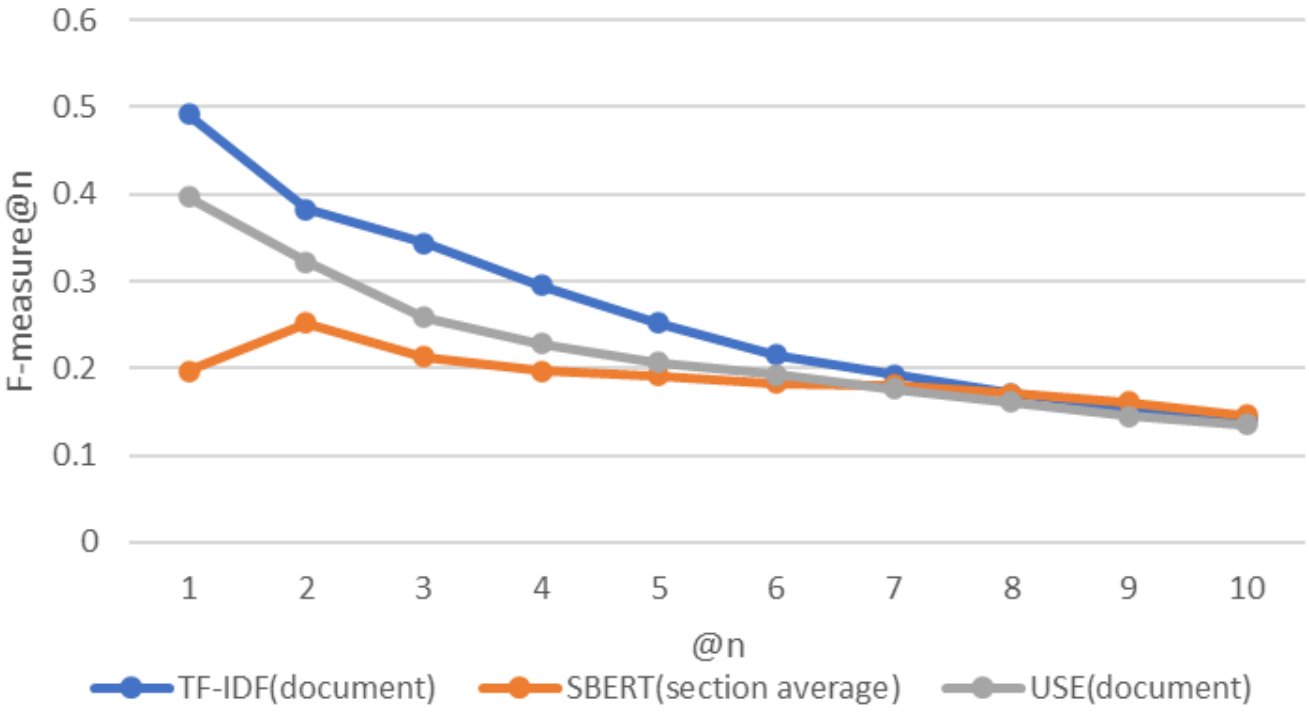
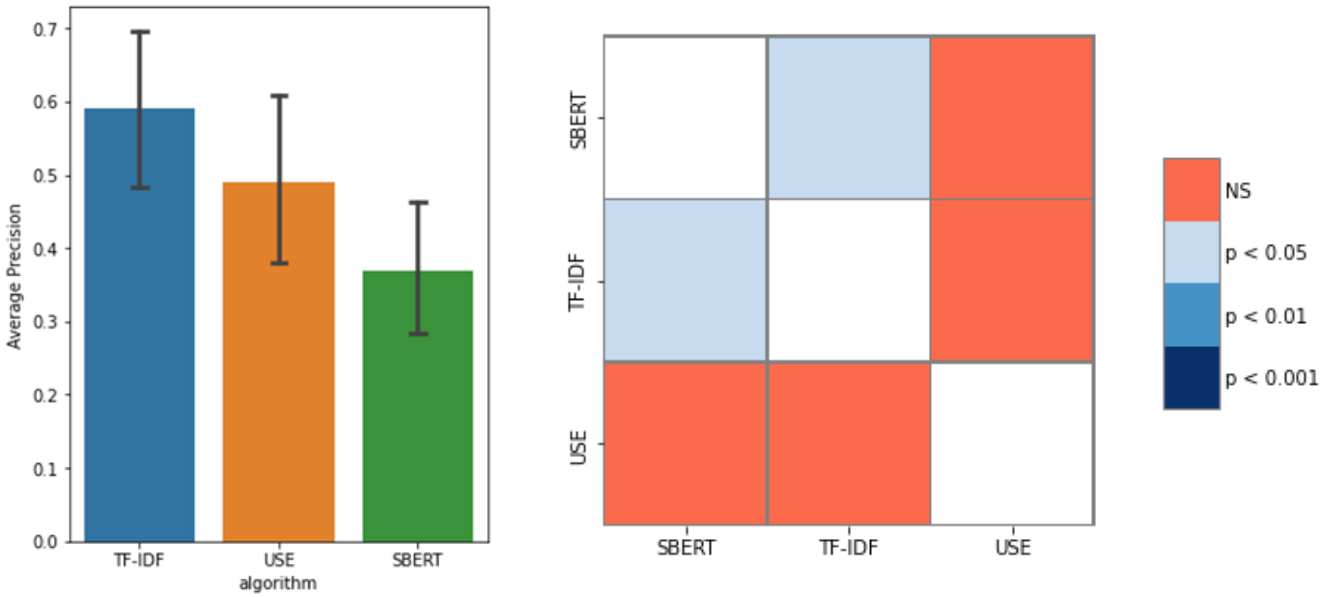


Figure 13. F-measure for each approach.

Figure 14. Box plot of average precision.



(**a**) (**b**)

Figure 15. (**a**) Plotting average precision data; (**b**) heatmap of the results of the Kruskal–Wallis test.

RQ 2. Answer

All three algorithms were successful in tracing. All algorithms were 70% successful. In order of success rate, SBERT successfully traced 49 CVE-IDs, TF-IDF traced 48 CVE-IDs, and USE traced 43 CVE-IDs.

4.7. RQ3 Which Algorithm Provides the Most Suitable Tracing?

The first factor to find candidate attack patterns that may arise from vulnerabilities is not to miss associated attacks. The second is the ranking quality. Thus, the most important metric is recall followed by MRR. SBERT had the best recall, as 49 of the 61 CVE-IDs were successfully traced (Figure 11). Similarly, TF-IDF successfully traced 48 CVE-IDs, which is not a significant difference. In MRR, TF-IDF showed the best performance. Therefore,

judging from the two evaluation indices comprehensively, TF-IDF is the most suitable. Below is a detailed analysis of TF-IDF and SBERT.

4.7.1. TF-IDF

The advantage of TF-IDF is that it can make decisions based on the importance of words. This is a well-utilized tactic in similarity judgment without missing security and system terminologies, which are important for linking CVE and CAPEC. On the other hand, factors that could not be identified by TF-IDF are due to the inability to understand the context. An example is CVE-2020-0601. CVE-2020-0601 is associated with CAPEC-475. CAPEC-475 is signature spoofing by improper validation. The description of CVE-2020-0601 is as follows:

A spoofing vulnerability exists in the way that Windows CryptoAPI (Crypt32.dll) validates elliptic curve cryptography (ECC) certificates. An attacker could exploit the vulnerability using a spoofed codesigning certificate to sign a malicious executable, making it appear the file was from a trusted, legitimate source, aka "Windows CryptoAPI Spoofing Vulnerability" [43].

Both CVE-2020-0601 and CAPEC-475 contain many words related to cryptography, such as "cryptography," "cryptographic," "cryptographically," and "cryptanalysis." TF-IDF cannot measure a high similarity because it does not understand the context. Context-sensitive algorithms such as SBERT can compensate for this shortcoming.

4.7.2. SBERT

The advantage of SBERT is that it understands context. On the other hand, factors that could not be identified by SBERT are based on other words that are not important in the link. An example is CVE-2004-0629. CVE-2004-0629 is associated with CAPEC52. CAPEC52 is Embedding NULL Bytes. The description of CVE-2004-0629 is as follows:

Buffer overflow in the ActiveX component (pdf.ocx) for Adobe Acrobat 5.0.5 and Acrobat Reader, and possibly other versions, allows remote attackers to execute arbitrary code via a URI for a PDF file with a null terminator (%00) followed by a long string [44].

When CVE-2004-0629 is entered, SBERT recommended many of the CAPEC-IDs related to "flash." This may be based on the word "Adobe." In addition, since it is written as buffer overflow, there were many CAPEC-IDs recommended for buffer overflow. However, the important part of this CVE description is "a null terminator (%00)," and it cannot be identified unless this word is well utilized. Incidentally, TF-IDF was successful in this linking. One possible solution is to increase the amount of training data. In this study, the training data consisted of about 150 sentences. However, merely increasing the number is inadequate. Learning the relationship between CVE and CWE decreased the accuracy. Additionally, this study used CosineSimilarityLoss, but TripletLoss may be more appropriate. Therefore, finetuning the data collected by the cybersecurity group with TripletLoss should improve the accuracy.

RQ3. Answer

TF-IDF was the best algorithm overall based on the two values of recall and MRR because it can identify the security terms that are important to the link and measure.

4.8. Findings

Our approach has different uses for vulnerability reporters and system administrators.

For vulnerability reporters, it is useful to determine if there is an insufficient description in the report. For example, TF-IDF traced CVE-2004-0629 to CAPEC-52. CAPEC-52 is Embedding NULL Bytes. The description of CVE-2004-0629 is shown above. Although this might be considered buffer overflow, CVE-2004-0629 was traced to CAPEC-52. This indicates that the quality of the description of CVE-2004-0629 has been ensured.

On the other hand, CVE-2006-4705 was not traced to CAPEC-54. CAPEC-54 is a Query System for Information. The description of CVE-2006-4705 is as follows:

SQL injection vulnerability in login.php in dwayner79 and Dominic Gamble Timesheet (aka "Timesheet.php") 1.2.1 allows remote attackers to execute arbitrary SQL commands via the username parameter [45].

In the above description, SQL injection should have been described as blind SQL injection to improve the quality of vulnerability reporting.

For system administrators, it is useful to understand the scenario-based attacks that may occur in the future. For example, CVE-2006-2786 is an HTTP Response Smuggling vulnerability. CVE-2006-2786 was traced to CAPEC-273 (HTTP Response Smuggling). The CAPEC273 page shows that it leads to XSS and cache poisoning. System administrators can consider their environment and anticipate subsequent attacks for vulnerability management. In this way, the CAPEC description of the trace destination can be used to predict future events, which should be useful for more accurate vulnerability management.

4.9. Threats to Validity

The ground truth was based on the link set up by MITRE. Although the correctness is ensured, MITRE may have omitted some links. This is a threat to the internal validity. In the future, the usefulness of other link candidates should be examined.

A threat to external validity is that the effectiveness of the proposed approach for all CVEs was not verified. In the future, the effectiveness of the approach should be validated by randomly selecting CVE-IDs.

5. Conclusions and Future Works

Herein, one approach to trace associated CAPEC-IDs directly from CVE-IDs is proposed and different patterns are evaluated. The conventional method follows the links between each cybersecurity database, which requires manual linking. This leads to accuracy issues. By contrast, the tracing approach in this study uses a similarity algorithm. Different patterns (three similarity measures and three algorithms) using the proposed approach were experimentally evaluated. Although each algorithm has its own merits, TF-IDF is the most suitable overall.

In the future, employing ensemble learning [46] may improve accuracy. Ensemble learning merges multiple models that have been trained individually. Specifically, it combines the predictions of various similarity algorithms in a process, for example, "taking an average".

This study has limitations in tracing perfectly the best attack pattern. However, the impact of the traced attack patterns on the actual vulnerability management is evaluated. This study used CAPEC attack pattern information, but attack information is available elsewhere. Examples are the Pattern Language [47] and ATT&CK. In the future, the candidates of the tracing attack information should be expanded. Additionally, studies should focus on expanding the proposed approach into a comprehensive and proactive cyber threat intelligence (CTI) [48,49] by collecting and analyzing data from many cyber security databases.

Author Contributions: Conceptualization, methodology, and writing—original draft preparation, K.K.; funding acquisition, H.W.; formal analysis and writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the SCAT Research Grant, the MEXT enPiT-Pro Smart SE: Smart Systems and Services innovative professional Education program, and the JST-Mirai Program grant number JPMJMI20B8.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Common Vulnerabilities and Exploits. Available online: https://cve.mitre.org/ (accessed on 16 June 2021).
2. Common Attack Pattern Enumeration and Classification. Available online: https://capec.mitre.org/ (accessed on 16 June 2021).
3. Common Weakness Enumeration. Available online: https://cwe.mitre.org/ (accessed on 16 June 2021).
4. National Vulnerability Database. Available online: https://nvd.nist.gov/ (accessed on 10 February 2022).
5. Kanakogi, K.; Washizaki, H.; Fukazawa, Y.; Ogata, S.; Okubo, T.; Kato, T.; Kanuka, H.; Hazeyama, A.; Yoshioka, N. Tracing CAPEC Attack Patterns from CVE Vulnerability Information using Natural Language Processing Technique. In Proceedings of the 54th Hawaii International Conference on System Sciences, Kauai, HI, USA, 4–8 January 2021; pp. 6996–7004.
6. Kanakogi, K.; Washizaki, H.; Fukazawa, Y.; Ogata, S.; Okubo, T.; Kato, T.; Kanuka, H.; Hazeyama, A.; Yoshioka, N. Tracing CVE Vulnerability Information to CAPEC Attack Patterns Using Natural Language Processing Techniques. *J. Inf.* **2021**, *12*, 298. [CrossRef]
7. Miller, D.; Leek, T.; Schwartz, R. A hidden Markov model information retrieval system. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 15–19 August 1999; pp. 214–221. [CrossRef]
8. Cer, D.; Yang, Y.; Kong, S.Y.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. Universal sentence encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 31 October–4 November 2018; pp. 169–174.
9. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
10. Dang, Q.; François, J. Utilizing attack enumerations to study SDN/NFV vulnerabilities. In Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 25–29 June 2018; pp. 356–361. [CrossRef]
11. Navarro, J.; Legrand, V.; Lagraa, S.; Francois, J.; Lahmadi, A.; Santis, G.D.; Festor, O.; Lammari, N.; Hamdi, F.; Deruyver, A.; et al. HuMa: A multi-layer framework for threat analysis in a heterogeneous log environment. In Proceedings of the 10th International Symposium on Foundations & Practice of Security, Nancy, France, 23–25 October 2017; pp. 144–159. [CrossRef]
12. Scarabeo, N.; Fung, B.C.M.; Khokhar, R.H. Mining known attack patterns from security-related events. *PeerJ Comput. Sci.* **2015**, *1*, e25. [CrossRef]
13. Ma, X.; Davoodi, E.; Kosseim, L.; Scarabeo, N. Semantic Mapping of Security Events to Known Attack Patterns. In Proceedings of the 23rd International Conference on Natural Language and Information Systems, Paris, France, 13–15 June 2018; Volume 10859, pp. 91–98.
14. MITRE ATT&CK. Available online: https://attack.mitre.org (accessed on 10 February 2022).
15. Aghaei, E.; Shaer, E.A. ThreatZoom: Neural Network for Automated Vulnerability Mitigation. In Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security, New York, NY, USA, 1–3 April 2019; pp. 1–3. [CrossRef]
16. Ampel, B.; Sagar Samtani, S.; Ullman, S.; Chen, H. Linking Common Vulnerabilities and Exposures to the MITRE ATT&CK Framework: A Self-Distillation Approach. In Proceedings of the 2021 ACM Conference Knowledge Discovery and Data Mining (KDD' 21) Workshop on AI-enabled Cybersecurity Analytics, Singapore, 14–18 August 2021; pp. 1–5.
17. Kuppa, A.; Aouad, L.; Le-Khac, N. Linking CVE's to MITRE ATT&CK Techniques. In Proceedings of the 16th International Conference on Availability, Reliability and Security, New York, NY, USA, 17–20 August 2021; pp. 1–12. [CrossRef]
18. Ouchn, J.N. Method and System for Automated Computer Vulnerability Tracking. U.S. Patent 9,871,815, 16 January 2018.
19. Adams, S.; Carter, B.; Fleming, C.; Beling, P.A. Selecting system specific cybersecurity attack patterns using topic modeling. In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, New York, NY, USA, 31 July 2018–3 August 2018; pp. 490–497. [CrossRef]
20. Bezdan, T.; Stoean, C.; Naamany, A.A.; Bacanin, N.; Rashid, T.A.; Zivkovic, M.; Venkatachalam, K. Hybrid Fruit-Fly Optimization Algorithm with K-Means for Text Document Clustering. *Mathematics* **2021**, *9*, 1929. [CrossRef]
21. Mounika, V.; Yuan, X.; Bandaru, K. Analyzing CVE Database Using Unsupervised Topic Modelling. In Proceedings of the 6th Annual Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 5–7 December 2019; pp. 72–77. [CrossRef]
22. Ou, S.; Kim, H. Unsupervised Citation Sentence Identification Based on Similarity Measurement. In Proceedings of the 13th International Conference on Transforming Digital Worlds, Sheffield, UK, 25–28 March 2018; pp. 384–394. [CrossRef]
23. Kim, D.; Seo, D.; Cho, S.; Kang, P. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *J. Inf. Sci.* **2019**, *477*, 15–29. [CrossRef]
24. Zhu, L.; Zhang, Z.; Xia, G.; Jiang, C. Research on Vulnerability Ontology Model. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, 24–26 May 2019; pp. 657–661. [CrossRef]
25. Gao, J.B.; Zhang, B.W.; Chen, X.H.; Luo, Z. Ontology-based model of network and computer attacks for security assessment. *J. Shanghai Jiaotong Univ. Sci.* **2013**, *18*, 554–562. [CrossRef]
26. Ansarinia, M.; Asghari, S.A.; Souzani, A.; Ghaznavi, A. Ontology-based modeling of DDoS attacks for attack plan detection. In Proceedings of the 6th International Symposium on Telecommunications, Tehran, Iran, 6–8 November 2012; pp. 993–998. [CrossRef]

27. Wang, J.A.; Wang, H.; Guo, M.; Zhou, L.; Camargo, J. Ranking attacks based on vulnerability analysis. In Proceedings of the 43rd Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2010; pp. 1–10. [CrossRef]
28. Wita, R.; Jiamnapanon, N.; Teng-Amnuay, Y. An ontology for vulnerability lifecycle. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, China, 2–4 April 2010; pp. 553–557. [CrossRef]
29. Lee, Y.; Woo, S.; Song, Y.; Lee, J.; Lee, D.H. Practical Vulnerability-Information-Sharing Architecture for Automotive Security-Risk Analysis. *IEEE Access* **2020**, *8*, 120009–120018. [CrossRef]
30. Stellios, I.; Kotzanikolaou, P.; Grigoriadis, C. Assessing IoT enabled cyber-physical attack paths against critical system. *J. Comput. Secur.* **2021**, *107*, 102316. [CrossRef]
31. Rostami, S.; Kleszcz, A.; Dimanov, D.; Katos, V. A Machine Learning Approach to Dataset Imputation for Software Vulnerabilities. In Proceedings of the 10th International Conference on Multimedia Communications, Services and Security, Krakow, Poland, 8–9 October 2020; pp. 25–36. [CrossRef]
32. Sion, L.; Tuma, K.; Scandariato, R.; Yskout, K.; Joosen, W. Towards Automated Security Design Flaw Detection. In Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshops, San Diego, CA, USA, 10–15 November 2019; pp. 49–56. [CrossRef]
33. Almorsy, M.; Grundy, J.; Ibrahim, A.S. Collaboration-based cloud computing security management framework. In Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, 4–9 July 2011; pp. 364–371. [CrossRef]
34. Kotenko, I.; Doynikova, E. The CAPEC based generator of attack scenarios for network security evaluation. In Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw, Poland, 24–26 September 2015; pp. 436–441. [CrossRef]
35. Xianghui, Z.; Yong, P.; Zan, Z.; Yi, J.; Yuangang, Y. Research on parallel vulnerabilities discovery based on open source database and text mining. In Proceedings of the 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Adelaide, Australia, 23–25 September 2015; pp. 327–332. [CrossRef]
36. Ruohonen, J.; Leppanen, V. Toward Validation of Textual Information Retrieval Techniques for Software Weaknesses. Commun. *Comput. Inf. Sci.* **2018**, *903*, 265–277. [CrossRef]
37. Guo, M.; Wang, J.A. An ontology-based approach to model common vulnerabilities and exposures in information security. In Proceedings of the ASEE 2009 Southest Section Conference, Marietta, GA, USA, 5–7 April 2009.
38. CVE. Available online: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-10108 (accessed on 10 February 2022).
39. Scikit-Learn. Available online: https://scikit-learn.org/stable/ (accessed on 10 February 2022).
40. Tensorflow Hub. Available online: https://tfhub.dev/ (accessed on 10 February 2022).
41. Sentence Transformers Documentation. Available online: https://www.sbert.net/ (accessed on 10 February 2022).
42. CWE. Available online: https://cwe.mitre.org/data/definitions/20.html (accessed on 10 February 2022).
43. CVE. Available online: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0601 (accessed on 10 February 2022).
44. CVE. Available online: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0629 (accessed on 10 February 2022).
45. CVE. Available online: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4705 (accessed on 10 February 2022).
46. Xia, P.; Zhang, L.; Li, F. Learning similarity with cosine similarity ensemble. *Inf. Sci.* **2015**, *307*, 39–52. [CrossRef]
47. Hafiz, M.; Adamczyk, P.; Johnson, R. Growing a pattern language (for security). In Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Tucson, AZ, USA, 19–26 October 2012; pp. 139–158. [CrossRef]
48. Biswas, B.; Mukhopadhyay, A.; Gupta, G. "Leadership in Action: How Top Hackers Behave" A Big-Data Approach with Text-Mining and Sentiment Analysis. In Proceedings of the 51st Hawaii International Conference on System Sciences, Honolulu, HI, USA, 2–6 January 2018; pp. 1752–1761. [CrossRef]
49. Samtani, S.; Chinn, R.; Chen, H.; Nunamaker, J.F., Jr. Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. *J. Manag. Inf. Syst.* **2017**, *34*, 1023–1053. [CrossRef]