Predicting known Vulnerabilities from Attack News: A Transformer-Based Approach

Refat Othman[1,*,†], Diaeddin Rimawi[1], Bruno Rossi[2] and Barbara Russo[1]

[1]Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy
[2]Masaryk University, Brno, Czech Republic

Abstract

Identifying the vulnerabilities exploited during cyberattacks is essential for enabling timely responses and effective mitigation in software security. This paper directly examines the process of predicting software vulnerabilities, specifically Common Vulnerabilities and Exposures (CVEs), from unstructured descriptions of attacks reported in cybersecurity news articles. We propose a semantic similarity-based approach utilizing the `multi-qa-mpnet-base-dot-v1` (MPNet) sentence transformer model to generate a ranked list of the most likely CVEs corresponding to each news report. To assess the accuracy of the predicted vulnerabilities, we implement four complementary validation methods: filtering predictions based on similarity thresholds, conducting manual validation, performing semantic comparisons with the first vulnerability explicitly mentioned in each report, and comparing against all CVEs referenced within the report. Experimental results, drawn from a dataset of 100 SecurityWeek news articles, demonstrate that the model attains a precision of 81% when employing threshold-based filtering. Manual evaluations report that 70% of the predictions are relevant, while comparisons with the initially mentioned CVEs reveal agreement rates of 80% with the first listed vulnerability and 78% across all referenced CVEs. In 57% of the news reports analyzed, at least one predicted vulnerability precisely matched a CVE-ID mentioned in the article. These findings underscore the model's potential to facilitate automated vulnerability identification from real-world cyberattack news reports.

Keywords

Vulnerability detection, LLMs, Transformer models, MITRE repositories, CVEs

1. Introduction

Timely response to cyberattacks is essential to limit damage and prevent further exploitation. News reports frequently provide early details about such incidents, often before technical reports are available. When a cyberattack is reported in the news, it is crucial to quickly identify and link the underlying vulnerabilities exploited. Delays in establishing this connection can leave systems exposed to ongoing threats, increasing both risk and remediation costs. Cybersecurity threats have become increasingly pervasive, with organizations experiencing an average of over 1,000 attacks per week [1]. Check Point Research reported a 28% increase in the frequency of such attacks during the first quarter of 2024 compared to the prior quarter [2]. This escalation can be attributed to the rapid advancement of technology, which has enabled cybercriminals to devise innovative methods for exploiting system vulnerabilities; in fact, the number of identified vulnerabilities has surged by 25% over the past two years [3]. In response to these growing threats, various initiatives have been developed to enhance organizational defenses, among which Cyber Threat Intelligence (CTI) has emerged as a critical strategy [4]. CTI encompasses the systematic process of collecting, analyzing, and disseminating information concerning potential cyber threats and vulnerabilities that may compromise an organization's security posture [5, 6].

A prominent resource utilized within CTI is the MITRE family of repositories [7], which provides publicly available information regarding contemporary attacks and software/hardware vulnerabilities. This family comprises several key components: (1) The Adversarial Tactics, Techniques, and Common

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT
rothman@unibz.it (R. Othman); diaeddin.rimawi@unibz.it (D. Rimawi); brossi@mail.muni.cz (B. Rossi); brusso@unibz.it (B. Russo)
0000-0003-1227-9734 (R. Othman); 0000-0003-3791-399X (D. Rimawi); 0000-0002-8659-1520 (B. Rossi); 0000-0003-3737-9264 (B. Russo)

Knowledge (ATT&CK) repository [7], which catalogues various attack tactics, techniques, and procedures. (2) Common Vulnerabilities and Exposures (CVE) [8], which details known vulnerabilities[1] along with the affected systems and products. (3) The Common Weakness Enumeration (CWE) [11], a collaboratively developed catalog of software weaknesses, coding errors, and security flaws. (4) The Common Attack Pattern Enumeration and Classification (CAPEC) [12], which offers a compilation of recognized attack patterns that exploit known weaknesses in systems and products [13, 14, 15].

These repositories provide a remarkable set of information for CTI. However, navigating and linking such a large amount of information poses challenges. This work aims to contribute to CTI by providing a methodology and an approach to automatically link attacks to vulnerabilities exposed by software systems. Specifically, we offer cybersecurity researchers and practitioners a CTI model derived from the textual information of the MITRE repositories that predicts known vulnerability descriptions from reported attacks. Linking attack-related news reports to specific CVEs can assist the cybersecurity community in understanding emerging threats and response priorities. To illustrate the need for such automation, consider a security analyst who is reviewing a news report describing a newly discovered breach. At this early stage, information vulnerability such as CVEs is often not available, yet rapid insights are essential for prioritizing response actions. Our approach addresses this gap, as soon as an article is published, the system can automatically suggest which known vulnerability is most likely being exploited, enabling analysts to quickly assess exposure, initiate patching, and guide incident response decisions. For instance, the model can be embedded within CTI platforms or news monitoring dashboards to automatically flag relevant CVEs whenever new attack reports appear, reducing the time and manual effort required to link emerging threats with known vulnerabilities. However, linking attacks manually with 295,604 CVE issues [8] is a non-trivial task requiring automated models to link attack information with vulnerabilities and weaknesses.

This paper presents a novel method for automating vulnerability detection in cyberattack news articles using the best-performing sentence-transformer identified in our prior studies [16, 10], `multi-qa-mpnet-base-dot-v1 (MPNet)`. Among all models evaluated in our earlier work [16, 10], MPNet achieved the highest accuracy due to its hybrid pre-training scheme, which combines masked and permuted language modeling. This training strategy equips MPNet with the ability to capture long-range dependencies and subtle semantic links between attack narratives and vulnerability descriptions. Moreover, its fine-tuning on large-scale question–answer datasets further enhances its semantic similarity capabilities, making it particularly effective for analyzing short, technical security text. While MPNet has previously shown strong performance in linking MITRE-curated attack descriptions with CVE records [16], its effectiveness on unstructured, real-world cybernews has not yet been investigated. To address this gap, we introduce the first dataset that annotates SecurityWeek articles [17] with their corresponding vulnerabilities, enabling systematic and reproducible evaluation in this domain. To support this task, we adopt a transfer-learning strategy in which MPNet is first fine-tuned on structured MITRE sources, where attack and CVE descriptions provide clean, well aligned semantic signals, and subsequently adapted to the very different linguistic characteristics of cyberattack news. Thus, we also release an initial prototype that applies the transfer-learned, fine-tuned MPNet model to news articles, enabling automated estimation of the vulnerabilities likely implicated in reported incidents. This prototype demonstrates how transfer learning, leveraging a model originally optimized for structured MITRE attack descriptions, can be effectively adapted to analyse unstructured real-world cyberattack news. Both our annotated dataset [18] and the accompanying implementation for model inference, transfer-learning fine-tuning, and validation are publicly available on GitHub [19] to support reproducibility and future work.

Thus, we intend to answer the following research questions:

> **RQ1:** *To what extent can our approach predict software vulnerabilities from free-form textual descriptions of cyberattacks news?*

[1] A *vulnerability* is defined as a flaw in software code that has been exploited, potentially impacting the availability, confidentiality, or integrity of an organization's assets [9, 10]

To answer this question, we conduct a manual evaluation of the CVEs predicted by our model using a dataset of 100 cybersecurity news reports from SecurityWeek. Each report is processed using a sentence transformer to generate a top-K list of likely CVEs based on semantic similarity. We manually reviewed each predicted CVE by comparing its description to the content of the corresponding news report, determining whether the predicted vulnerability was relevant to the attack described.

RQ2: *How effective are our oracle-based validation methods in assessing the accuracy of CVE predictions from attack descriptions?*
To answer this question, we applied three automated validation methods based on cosine similarity. These methods respectively assess: (i) the semantic relevance of predicted CVEs above a similarity threshold, (ii) alignment with the first CVE mentioned in each report, and (iii) consistency with all CVEs cited in the news report.

Overall, the major contributions of our work are the following:

- A new application of an MPNet-based sentence-transformer model to real-world cyberattack news data;
- A multi-method evaluation framework combining expert manual analysis with three oracle-based validation techniques;
- An empirical study on 100 SecurityWeek articles, evaluated through both manual expert review and automated validation;
- A practical contribution to CTI automation by demonstrating how transformer-based semantic similarity can support early vulnerability identification from unstructured text;

The paper is structured as follows: Section 2 briefly summarises the background, key concepts, and prior studies on software vulnerabilities. Section 3 discusses our methodology, including the tool and dataset. Section 4 shows the results of our approach. Section 5 identifies the limitations and summarizes the threats to validity. Section 6 discusses the related work, and the paper concludes in Section 7.

2. Background

In this section, we review the core concepts of our study: vulnerability knowledge bases, cyberattack news for threat intelligence, and the MPNet transformer model.

2.1. Vulnerability Knowledge Bases

According to the NIST National Vulnerability Database [3], a vulnerability is a flaw in the computational logic of software or hardware that, if exploited, can compromise confidentiality, integrity, or availability. Such vulnerabilities are core to the cybersecurity threat landscape, as they are commonly exploited to gain unauthorized access, execute malicious code, or disrupt services [9, 20, 21, 22]. Given the unpredictability of when and how a vulnerability will be exploited, it is essential to equip stakeholders with effective tools to detect and mitigate them [23, 24, 25, 26, 27]. The CVE repository provides a centralized catalog of publicly disclosed software vulnerabilities [20, 28, 29], assigning a unique CVE identifier to each entry for consistent cross-referencing across systems. Each CVE typically includes a brief description, identification number, and external references. Many entries also reference CWE categories [11], which classify underlying weakness types. Additional metadata may include patch details, severity scores based on the Common Vulnerability Scoring System (CVSS) [30], and impact assessments. Many vulnerabilities remain undetected until exploited, as traditional methods struggle with growing threat complexity. Analyzing unstructured sources like cyber news offers a proactive path to improve defense and reduce mitigation efforts [31].

2.2. Cyberattack News

Cyberattack reports in the form of news reports, incident disclosures, blogs, and advisories have emerged as rich sources of real-time threat intelligence. These texts typically describe adversarial behavior, targeted systems, exploited vectors, and observed outcomes. Unlike structured databases, news-based reports provide contextual and chronological information that may reveal unknown or unlinked vulnerabilities. As such, they represent an underutilized yet valuable asset for proactive security analysis [32].

Extracting actionable intelligence from these texts, however, is non-trivial. The descriptions are often ambiguous, vary in terminology, and lack explicit references to known vulnerabilities [33]. For example, a news report may mention that attackers gained access by exploiting a weakness in a widely used library without naming the corresponding CVE. This lack of specificity significantly limits the effectiveness of automated extraction systems, particularly those dependent on strict keyword matching or predefined rule-based approaches. To address this challenge, natural language processing (NLP) techniques extract meaningful information from unstructured text [34]. Combining NLP methods with structured cybersecurity repositories makes it possible to uncover links between textual attack descriptions and known vulnerabilities even without explicit identifiers. Moreover, the effectiveness of these methods is strongly influenced by the model's capability to understand complex semantic relationships within the text.

2.3. NLP and MPNet Sentence Transformer

Sentence transformer models are pre-trained models that produce sentence embeddings for various natural language processing tasks, such as semantic search, paraphrasing, and clustering. However, these models usually contain hundreds of millions of parameters, which brings challenges for fine-tuning and online serving in real-life applications for latency and capacity constraints. In this work, we use the `multi-qa-mpnet-base-dot-v1` model [35], which is based on the MPNet architecture [36]. MPNet is an improved Transformer-based model that extends BERT [37] by integrating both masked language modeling and permuted language modeling, allowing the model to capture dependencies between tokens better and generate richer semantic embeddings. The `multi-qa-mpnet-base-dot-v1` model has been fine-tuned on over 200 million question-answer pairs from diverse domains [35], making it particularly effective for semantic search and question-answer retrieval. The MPNet model outputs 768-dimensional vectors and can encode a sentence's meaning with high precision. In this study, we use pre-trained transformer models to generate embeddings for attack and vulnerability descriptions and then apply a similarity layer based on cosine similarity to determine whether an attack is linked to a vulnerability (Section 3.3).

3. Methodology

In this section, we describe our research methodology for detecting software vulnerabilities based on real-world cyberattack news, as illustrated in Fig. 1 [38]. The process starts with the collection of a dataset of cybersecurity news reports (Section 3.1). This is followed by a comprehensive text pre-processing step to clean and standardize the textual data (Section 3.2). Next, we employ the fine-tuned MPNet Sentence Transformer to generate high-dimensional embeddings for both the processed news reports and the known CVE vulnerability descriptions (Section 3.3). A similarity calculation is then performed between each news embedding and all CVE embeddings using cosine similarity, discussing sensitivity analysis (Section 3.4). Finally, the performance of the detection method is evaluated against a ground truth using the validation strategies M1-M4 (Section 3.5).

3.1. Data Collection

MITRE source (Attacks-vulnerabilities Mapping, $\mathcal{M}$): The MITRE-based dataset used to fine-tune

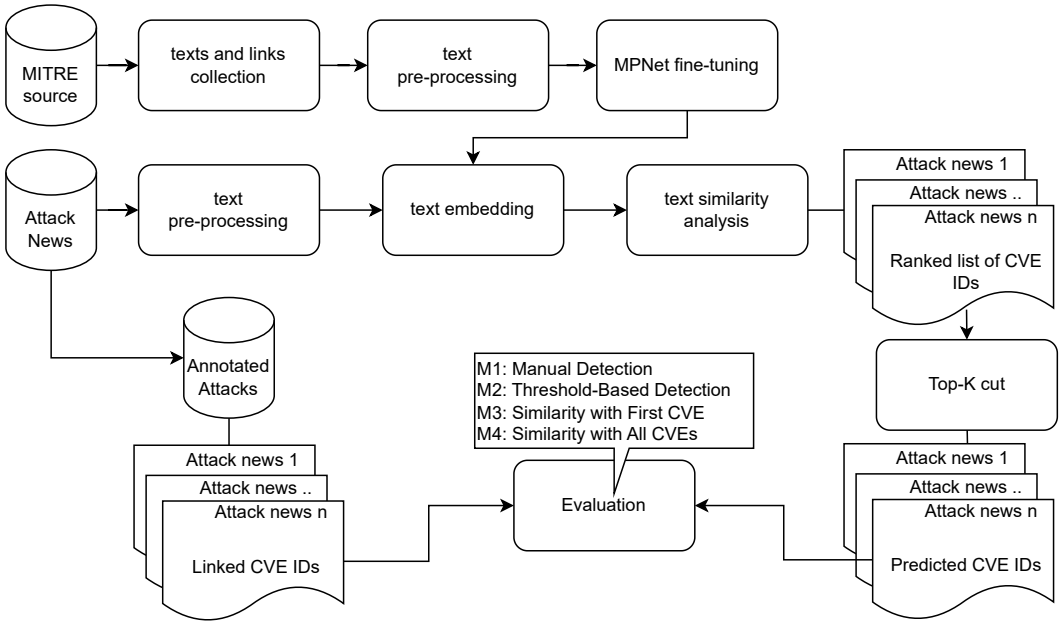

Figure 1: Overview of methodology.

our MPNet model for vulnerability prediction. Specifically, it provides the ground-truth associations between attack descriptions and the CVE reports. In what follows, we describe the construction of the mapping $\mathcal{M}$ that links attack descriptions to CVE entries. These connections are drawn from the explicit relationships documented across the ATT&CK, CAPEC, CWE, and CVE repositories.

$$\mathcal{M} : A \rightarrow C$$
$$\mathcal{M}(a) = \{c \in \mathcal{C} \mid \exists\, a \rightarrow c\}$$

Here, A denotes the set of attack descriptions in ATT&CK and CAPEC, and C represents the set of all CVE records. Each attack description a is mapped to a corresponding set of CVEs through the explicit cross-repository links ($\rightarrow$) documented in ATT&CK, CAPEC, CWE, and CVE. To construct $\mathcal{M}$, we systematically reviewed the official repository pages and extracted every explicitly defined relationship.

Table 1
Attack descriptions linked and not linked to CVE reports.

	Tactic	Technique	Procedure	Attack Pattern
Linked	11	100	721	86
Not linked	3	525	88	473
Total	14	625	809	559

Tables 1 summarize the collected items and their link coverage across the different repository layers. In addition, the table shows the number of attack descriptions linked to CVE reports; most Tactics and Procedures contain such links, whereas the majority of techniques and Attack Patterns do not. An example of the mapping structure is shown in Table 2, illustrating how CAPEC-38 relates to the ATT&CK Technique T1574 and to multiple CVE entries, such as CVE-2022-4826 and CVE-2020-26284.

We use the final mapping $\mathcal{M}$ to construct an annotated dataset linking attack descriptions to their associated weaknesses and vulnerabilities. This dataset [18] serves as the ground truth for fine-tuning our MPNet model and for evaluating the accuracy of our vulnerability prediction approach.

SecurityWeek News Dataset: Following the construction of the MITRE-based fine-tuning dataset, we assembled a second, real-world evaluation dataset consisting of 100 cybersecurity news articles obtained

Table 2
Example of a CAPEC pattern and its associated technique and CVE reports.

CAPEC ID: Description	Technique ID: Description	CVE ID: Description
CAPEC-38: This pattern describes how an adversary inserts a malicious component into a program's trusted execution path so that the malicious code is executed instead of the legitimate resource. The attack may involve altering search paths or manipulating dependent libraries.	**T1574.007:** Adversaries may execute their own malicious payloads by hijacking environment variables used to load libraries. The PATH environment variable contains a list of directories (User and System) that the OS searches sequentially through in search of the binary that was called from a script or the command line.	**CVE-2022-4826:** A stored XSS vulnerability in the Simple Tooltips WordPress plugin before version 2.1.4 due to improper handling of shortcode attributes.
		CVE-2020-26284: A command execution vulnerability in Hugo for Windows caused by loading malicious files from the working directory via PATH hijacking.

from the *Vulnerabilities* section of SecurityWeek [17]. SecurityWeek is a reputable cybersecurity news outlet known for reporting newly discovered vulnerabilities, active exploitation campaigns, emergency patches, and vendor advisories. The articles were collected chronologically to reflect the latest publicly disclosed incidents. Each article typically contains a mixture of high-level narrative, technical descriptions of the vulnerability or exploit behavior, details about affected software or vendors, references to exploitation status in the wild. We collected the latest 100 reports as they appeared on the website and subsequently extracted any CVE identifiers mentioned within them. For each article, we retrieved the full text along with title and applied pattern-based extraction to identify CVE references, followed by manual verification. We also recorded the vendors and products mentioned in the text using keyword extraction and manual confirmation. Out of the 100 articles, 97 explicitly reference at least one CVE ID, while the remaining 3 discuss security flaws without naming specific vulnerabilities, providing useful test cases for implicit prediction scenarios. The dataset also captures a wide range of attack categories (e.g., remote code execution, privilege escalation, command injection, authentication bypass). This diversity ensures that the dataset covers variations in writing style, technical depth, and vulnerability classes, offering a realistic and challenging benchmark for evaluating the model's ability to infer relevant CVEs from unstructured natural-language descriptions. The manually verified CVE annotations, together with the raw article content, form the basis for assessing the real-world applicability and generalization capability of our vulnerability prediction approach.

3.2. Text pre-processing

We applied a light pre-processing pipeline designed to eliminate extraneous content while keeping the core semantics intact. We first standardized all text to lowercase and removed citations, embedded URLs, and other non-informative symbols. We intentionally avoided conventional NLP transformations such as stemming, lemmatization, and stop-word elimination. Prior research [39, 40] indicates that transformer models operate on subword tokenization and derive much of their strength from contextual and syntactic information, which can be distorted by these techniques. Function words often provide structural cues that guide attention mechanisms, and morphological variations may carry subtle meaning differences that transformers can exploit. By applying only minimal cleaning while preserving linguistic

and grammatical features, we ensure that the sentence transformer receives rich, contextually consistent input suitable for modeling relationships between attack descriptions and CVE entries.

3.3. Approach Architecture

We implemented our approach using a sentence transformer model, augmented with a semantic similarity computation layer. Specifically, the model calculates a similarity score (Equation 1) between a given attack description and each CVE issue. For this work, we fine-tuned a single sentence transformer model, `multi-qa-mpnet-base-dot-v1`, which was identified in our previous study as the strongest-performing architecture for cybersecurity text matching [16]. The model produces fixed-size sentence embeddings by encoding input texts through multiple transformer layers followed by mean pooling. Fine-tuning was performed on the ATT&CK-to-CVE mapping dataset to encourage high cosine similarity between embeddings of linked attack–CVE pairs and low similarity for non-linked pairs. We used an 80/10/10 split for training, validation, and testing, respectively, following recommendations for reproducibility and stable comparison [41]. CosineSimilarityLoss was employed during training, with four epochs, 100 warmup steps, and evaluation every 500 steps, consistent with established practices in sentence-transformer training [42]. This fine-tuning process adapts MPNet to the specific linguistic and semantic characteristics of cybersecurity attack descriptions and vulnerability reports.

Following standard practice in embedding-based NLP [43, 44], we use cosine similarity as the standard metric for comparing sentence embeddings. Prior work has shown its effectiveness for semantic matching tasks: Reimers and Gurevych [43] demonstrated strong gains in Sentence-BERT through cosine-based retrieval, while Muennighoff et al. [44] employed it as the default metric across more than 50 models in the MTEB benchmark. In our setting, cosine similarity (Equation. 1) is computed between the normalized embedding of an attack description p and each CVE embedding q, producing values in the range $[-1, 1]$ (practically $[0, 1]$ for our normalized vectors). For consistency with our threshold analysis, these values are reported on a 0–100 scale. We then rank all CVE embeddings according to their similarity to the attack vector and retain those above a threshold ρ:

$$\mathcal{L}_\rho(a) = \{c \in C \mid Sim(\vec{a}, \vec{c}) > \rho\}.$$

Here, C denotes the set of all CVE IDs, and $\mathcal{L}_\rho(a)$ represents the predicted vulnerabilities for attack a. A classification decision is positive when $\mathcal{L}_\rho(a)$ is non-empty and negative otherwise.

After fine-tuning, we used the model to encode both attack descriptions and CVE issues into a shared semantic embedding space (see Section 2.3). Cosine similarity is then applied to the normalized embedding vectors to quantify their semantic closeness. The resulting similarity score ranges from 0 (no similarity) to 1 (maximum similarity):

$$\text{Sim}(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{|\vec{p}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \tag{1}$$

Given an input attack text a, the model computes similarity scores between a and all CVE descriptions. The CVEs are then ranked in descending order of similarity, forming a list $\mathcal{L}$. We evaluate the model's performance based on varying values of k, where k denotes the number of top-ranked CVEs considered. Accordingly, our approach links each attack description with the top-k most semantically similar vulnerability reports. All experiments were executed on a GPU cluster consisting of six nodes, each equipped with an NVIDIA A100 GPU (80 GB), 192 GB of system memory, and a 16-core Intel Xeon 4208 processor.

3.4. Evaluation metrics and threshold/top-k sensitivity analysis

To evaluate our model, we leveraged the ground truth mappings between attack texts and CVE issues derived from MITRE repositories. These mappings served for sensitivity analysis to determine the optimal value of k, which controls how many top-ranked CVEs are associated with each attack description.

Specifically, for a given attack text a, our model outputs a ranked list $\mathcal{L}(a)$ of CVE issues based on cosine similarity. We assess how well this list overlaps with the true set of CVEs $\mathcal{M}(a)$ from MITRE, where a correct prediction satisfies

$$\mathcal{L}(a) \bigcap \mathcal{M}(a) \neq \emptyset$$

We compute the performance metrics by means of the cardinality of the sets in Table 3. We consider each attack as an instance, where the model predicts a positive if $\mathcal{L}(a)$ is non-empty and a negative otherwise. Using this setup, we calculate standard classification metrics: Precision, Recall, and their harmonic mean (F1 score). These are computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Table 3
Positives and Negatives.

Type	Description
Positives	$\{a \in \mathcal{A} : \exists\, c \in \mathcal{M}_a\}$
Negatives	$\{a \in \mathcal{A} : \nexists\, c \in \mathcal{M}_a\}$
True Positives (TP)	$\{a \in \mathcal{A} : \mathcal{L}_a \cap \mathcal{M}_a \neq \emptyset\}$
False Positives (FP)	$\{a \in \mathcal{A} : (\mathcal{L}_a \neq \emptyset \wedge \mathcal{M}_a = \emptyset)\}$
False Negatives (FN)	$\{a \in \mathcal{A} : (\mathcal{L}_a = \emptyset \wedge \mathcal{M}_a \neq \emptyset)\}$
True Negatives (TN)	$\{a \in \mathcal{A} : (\mathcal{L}_a = \emptyset \wedge \mathcal{M}_a = \emptyset)\}$

In similarity-based classification tasks, a decision threshold ρ is typically used to determine whether a predicted CVE is considered relevant to a given attack description. Although a default cutoff of 0.5 is commonly adopted, prior work has shown that such arbitrary thresholds often fail to yield optimal performance [45, 46]. To address this issue, we conducted a threshold-sensitivity analysis to identify the most effective value of ρ for our model. Similar to standard practices in binary decision calibration, we examined how Precision and Recall vary across different threshold values using the Precision–Recall (PR) curve [47]. High precision reflects a low false-positive rate, whereas high recall indicates fewer false negatives. Using the `multi-qa-mpnet-base-dot-v1` model and a balanced subset of MITRE technique descriptions (59 positive and 59 negative pairs), we plotted the PR curve across a range of similarity thresholds. As depicted in Fig. 2, the precision and recall curves intersect at $\rho = 0.58$, corresponding to the Equal Error Rate (EER) point. This threshold represents the most balanced trade-off between false positives and false negatives and is therefore adopted in our subsequent evaluations.

Beyond the decision threshold, the choice of the top-k predictions also plays a crucial role in ranking-based vulnerability identification. The value of k (i.e., the number of top-ranked CVE predictions to consider) directly impacts these metrics. Lower values of k may decrease the number of false positives but increase the number of false negatives. selection of relevant CVEs for each attack description. To identify the optimal value of k, we performed a sensitivity analysis using a balanced dataset from MITRE ATT&CK composed of 50 positive and 50 negative instances of technique descriptions. For each value of k, we computed Precision and Recall and visualized their variation using boxplots. As shown in Fig. 3, the boxplot illustrates how the average Precision and Recall scores change as k increases. At $k = 20$, the average values of Precision and Recall converge, resulting in the highest F1 score. This indicates that $k = 20$ offers the most balanced prediction performance, effectively guiding the selection of relevant CVEs for each attack description.

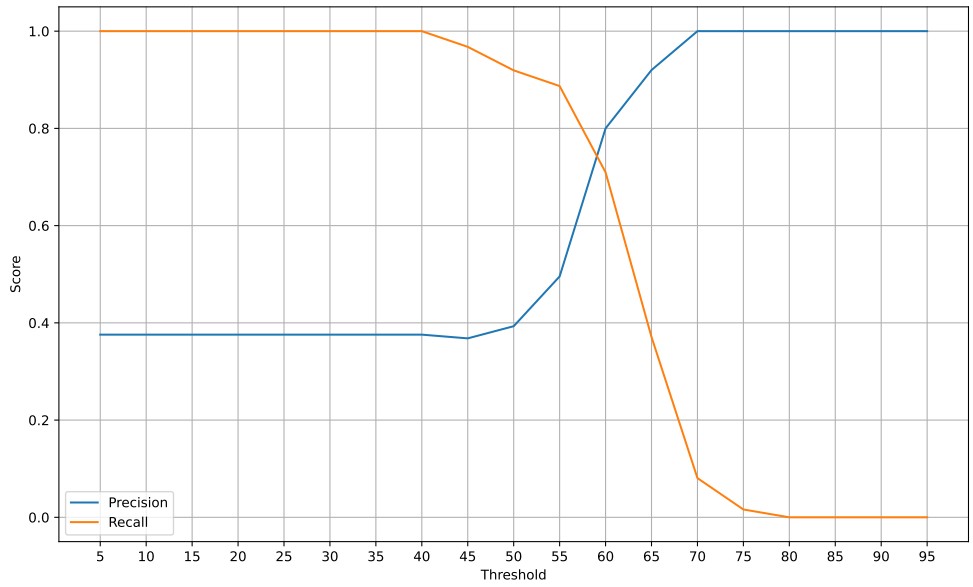

Figure 2: Precision-Recall curve for varying similarity threshold ρ.

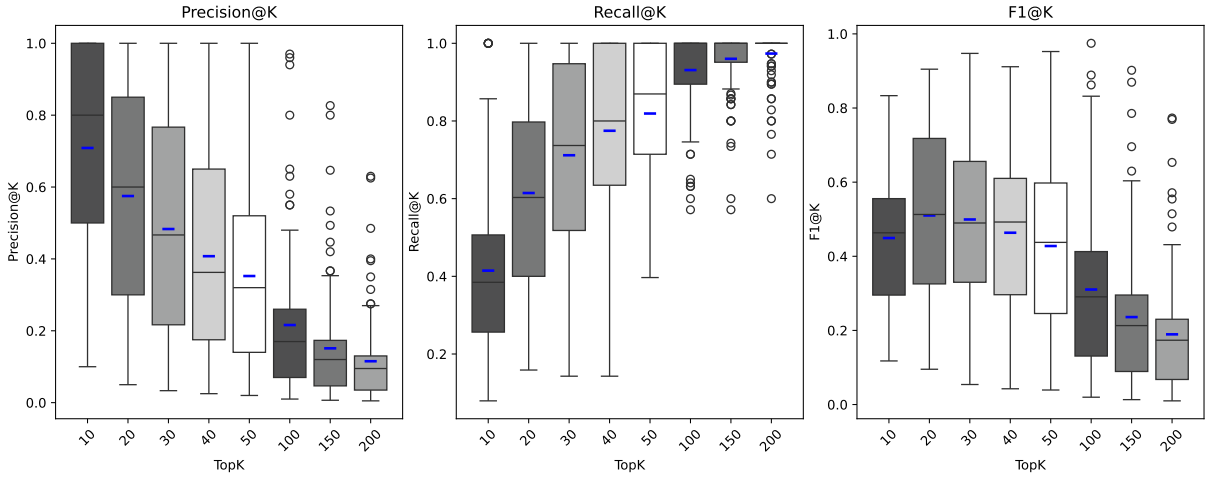

Figure 3: Performance measures of our approach over different k Values in top-k.

3.5. Model Evaluation and Validation

To answer our RQs, we evaluate the effectiveness of our model in predicting software vulnerabilities from attack descriptions. The output of our model consists of a ranked list of top-k vulnerability candidates for each attack description based on cosine similarity scores. We designed four evaluation strategies to validate the model's predictions: Three automated (oracle-based) methods and one manual. The methods target a distinct aspect of semantic similarity and practical performance. Let a denote an attack description, and let $C = \{c_1, c_2, ..., c_n\}$ be the set of vulnerability descriptions. Let $\text{sim}(a, c)$ represent the cosine similarity between the embeddings of a and c, and let ρ denote a fixed similarity threshold.

M1: Manual Detection. In this method, the study's two authors manually validated the predictions generated by the model. For each attack description, the top-k CVEs predicted by the model were reviewed to determine whether they were semantically and contextually related to the attack text:

$$\text{Predicted}_{\text{M1}}(a) = \{c_i \in C \mid c_i \text{ is judged related to } a \text{ by manual inspection}\}$$

M2: Threshold-Based Detection. In this method, we consider as relevant all predicted vulnerabilities from the model's top-k list whose similarity score exceeds the threshold ρ. Since the optimal value of ρ was already determined through our threshold-sensitivity analysis using a balanced subset of MITRE mappings (see Section 3.4), we directly apply the resulting value $\rho = 0.58$ in this evaluation. This value corresponds to the Equal Error Rate (EER), where precision and recall intersect on the Precision–Recall curve (Fig. 2). Using this calibrated threshold, the prediction set for M2 is defined as:

$$\text{Predicted}_{\text{M2}}(a) = \{\, c \in \text{top-}k(C) \mid \text{sim}(a, c) \geq \rho \,\}.$$

M3: Similarity with First CVE. This evaluation method investigates whether the model's predictions align semantically with the most vulnerable description mentioned in an attack news report. Instead of directly comparing the attack description to the CVE corpus, we extract the textual description of the first CVE explicitly referenced within the attack report. This CVE is assumed to represent the most central vulnerability discussed.

The embedding of this first CVE description, denoted as c_1^{attack}, is then compared to each predicted CVE from the model's output using cosine similarity. If the similarity score between the first CVE embedding and a predicted CVE exceeds the threshold ρ, the predicted CVE is considered a valid match:

$$\text{Predicted}_{\text{M3}}(a) = \{c \in C \mid \text{sim}(c_1^{\text{attack}}, c) \geq \rho\}$$

M4: Similarity with All CVEs. This evaluation method examines the model's ability to retrieve vulnerabilities semantically aligned with all CVEs mentioned in the attack report. Unlike M3, which uses a single CVE description, this method concatenates all CVE descriptions referenced in the attack report into a single text. The concatenated description is then embedded using the same sentence transformer model.

The resulting single embedding, which captures the combined semantics of all referenced vulnerabilities, is compared to the top-k predictions generated by the model. If a predicted CVE has a similarity score above the threshold ρ with the aggregated embedding, it is considered a correct prediction:

$$\text{Predicted}_{\text{M4}}(a) = \{c \in \text{top-}k(C) \mid \text{sim}(\text{concat}(C')^{\text{attack}}, c) \geq \rho\}$$

where $C' \subseteq C$ is the set of all CVEs explicitly mentioned in the attack report, and $\text{concat}(C')^{\text{attack}}$ represents the embedded concatenation of their descriptions.

These evaluation methods collectively validate the accuracy and reliability of the model's predictions. By combining automated and manual assessment techniques, we ensure that the predicted vulnerabilities are quantitatively strong (high precision) and contextually meaningful about the original attack descriptions. This multi-validation process enhances the credibility of the evaluation and provides a rigorous foundation for assessing the model's effectiveness in practical cybersecurity scenarios.

4. Results and Discussion

In this section, we present the results of our experiments and the outcomes for the RQs.

4.1. RQ1: To what extent can our approach predict software vulnerabilities from free-form textual descriptions of cyberattacks news?

To answer RQ1, we evaluated the performance of our approach in predicting CVE issues from news descriptions about cyberattacks through manual validation (M1: Manual Detection) (see Section 3.5). We evaluate the model's accuracy across 2000 predictions. To assess their validity, two authors manually reviewed each predicted CVE by comparing its description to the content of the corresponding news report, determining whether the prediction was contextually and semantically relevant.

Table 4
Validation outcomes for each method (M1–M4).

	Retrieved	Validated	
	Total	Relevant	Not Relevant
M1: Manual Detection	2000	1418 (70%)	582 (30%)
M2: Threshold-Based Detection	2000	1625 (81%)	375 (19%)
M3: Similarity with First CVE	2000	1607 (80%)	393 (20%)
M4: Similarity with All CVEs	2000	1575 (78%)	425 (22%)

This manual validation served as the ground truth for evaluating model accuracy. Among the 2000 predicted CVEs, 1418 (70%) were validated as relevant, demonstrating the model's strong potential to support human analysts in identifying vulnerabilities from natural language threat reports. The detailed results of this validation process, including a comparison across all evaluation methods, are summarized in Table 4. Moreover, our model's predictions overlapped with actual CVE mentions in 57 out of 100 news articles. In other words, the model predicts at least one exact CVE-ID truly mentioned, demonstrating a strong ability to pinpoint known vulnerabilities. The remaining cases include 40% with only unseen (but possibly relevant) predictions and 3% with no CVE mentions to match. Our approach can suggest plausible CVEs even when no explicit vulnerability identifiers (e.g., CVE-IDs) are mentioned in the news article, demonstrating its capability to infer hidden or implied vulnerabilities from unstructured text. However, this result is influenced by the choice of the top-k parameter, which was set to $k = 20$ in our study. A larger k could increase recall by including more potentially relevant CVEs in the prediction set, thereby improving the match rate. On the other hand, increasing k may also reduce precision and introduce noise, which would require more effort in filtering or post-validation. In future work, we plan to validate the effect of varying k values by comparing the model's performance across multiple cybersecurity news and threat reports datasets. This comparison will allow us to identify patterns in prediction behavior and compute an average optimal k value that balances precision and recall.

4.2. RQ2: How effective are our oracle-based validation methods in assessing the accuracy of CVE predictions from attack descriptions?

To answer RQ2, we investigated the effectiveness of our automated oracle-based validation methods in assessing the semantic accuracy of CVE predictions generated by our approach from unstructured cyberattack news reports. These methods are designed to represent ground truth when manual annotation is unavailable or impractical. Our evaluation focuses on three distinct oracle methods: M2: Threshold-based detection, M3: Similarity with First CVE, and M4: Similarity with All CVEs.

The M2 method evaluates model predictions by applying a similarity threshold ρ to the top-k ranked CVEs produced for each news report. By applying $\rho = 0.58$ to the model's top-k predictions, M2 yielded a precision of 81%, indicating that the majority of CVEs retrieved at or above this similarity level were contextually relevant to the attack descriptions, as summarized in Table 4.

The M3 method assesses whether the predicted vulnerabilities fall within the semantic proximity of the attack's core vulnerability. Thus, we filtered the predictions using the similarity threshold ($\rho = 0.58$) to retain only those with a similarity above the threshold. Table 4 shows that the M3 achieved an 80% agreement with the first-mentioned CVE, suggesting that the model consistently retrieves CVEs that are semantically coherent with the primary vulnerability highlighted in the news report. These results demonstrate the model's capacity to capture subtle contextual relationships and reflect its potential utility in prioritizing the most relevant threats in real-world reporting.

The M4 method extends M3 by evaluating the semantic similarity between the model's predicted CVEs and the combined set of all CVEs mentioned in the attack report. The results of M4 yielded a match rate of 78%, indicating that the model can identify vulnerabilities that align with the overall threat context described in multi-CVE issues. This further highlights the model's robustness in handling more complex and information-rich inputs, where multiple vulnerabilities contribute to the characterization of the attack. The three automated validation methods (M2, M3, M4) all show roughly similar success

rates (78–81%), providing confidence that our model's predictions align well with known vulnerabilities whether we compare them to a threshold or actual CVEs mentioned in reports.

5. Threats to Validity

In this section, we illustrate the threats to the validity of our study.

Construct validity. Construct validity refers to the degree to which theoretical claims or hypotheses articulated at a conceptual level are supported by empirical evidence obtained from operational measures [48]. In the context of our study, construct validity explicitly addresses the efficacy of our evaluation in accurately capturing the concept of predicting pertinent vulnerabilities from attack descriptions. Two main threats may impact this. First, the cosine similarity threshold ρ is not fixed; while we selected a value balancing false positives and false negatives, its tuning may vary by use case. A higher ρ favors precision, while a lower one favors recall. Second, in M3 and M4, we assume that the first-mentioned or aggregated CVEs in a report reflect ground truth. However, some reports may reference irrelevant or no CVEs. We mitigate these risks through consistent thresholds and leveraging real-world datasets from MITRE.

Internal validity. Internal validity reflects how confidently the results can be attributed to our model and evaluation design [49]. A potential threat lies in dataset selection; while the 100 SecurityWeek reports are credible, they may not reflect the full diversity of writing styles in cybersecurity reporting. Some texts include structured technical content or CVE references that differ from natural language patterns. Additionally, using a fixed top-k value ($k = 20$) ensures consistency but may not suit all cases equally. To mitigate these concerns, we applied uniform pre-processing, used a similarity threshold, and validated performance across multiple independent methods.

External validity. External validity concerns the generalizability of our findings beyond the current dataset and setup [49]. Our evaluation used 100 cybersecurity news reports from SecurityWeek, a reputable but single source, which may limit applicability to other news sources with different styles and terminology. Additionally, the method was tested only on English-language reports, excluding multilingual or alternative sources like social media or technical blogs. Additionally, our evaluation set of 100 news reports can be considered limited; expanding this number in future work would increase confidence in the generalizability of the results and validate the approach on broader and more diverse datasets. Additionally, our approach uses only the MPNet model, which may limit generalizability. In future work, we aim to test other transformer models and domain-adapted variants to improve robustness.

6. Related Work

In this section, we provide an overview of related work on vulnerability-attack models. This section summarizes recent advancements in this domain and contrasts them with our proposed approach. Kuppa et al. [50] proposed a multi-head deep embedding model to link CVE issues with MITRE ATT&CK techniques. Their method involved extracting relevant information from CVE metadata using regular expressions and comparing it with ATT&CK vectors via cosine similarity. However, this approach was limited to a small subset of ATT&CK techniques. Similarly, Sun et al. [51] employed a BERT-based model [37] to enrich textual CVE descriptions, aiding downstream information extraction and linkage tasks. Other studies approached the problem from a multi-label classification perspective. Lakhdhar et al. [52] experimented with various traditional and deep learning algorithms to map CVEs to ATT&CK tactics. Grigorescu et al. [53] introduced CVE2ATT&CK, a model that annotated CVE issues with relevant ATT&CK tactics using both BERT-based and classical machine learning models. Ampel et al. [54] proposed CVE Transformer (CVET), which incorporates a self-distillation mechanism for fine-tuning RoBERTa [55] to associate CVEs with one of ten ATT&CK tactics. Several studies have focused on linking attack patterns (e.g., CAPEC) to vulnerabilities (CVE) rather than leveraging Tactics, Techniques, and Procedures (TTPs). Kanakogi et al. [56, 57] and Hemberg et al. [58] used NLP

techniques, including RoBERTa, to compute semantic similarity between CAPEC and CVE descriptions. TF-IDF [59] and Doc2Vec [60] were also evaluated, with TF-IDF showing the most effective matching in ranking top-N relevant CAPEC documents for each CVE.

Prior work has explored reversing the typical mapping direction by predicting CVEs from textual descriptions of ATT&CK techniques [10]. One study introduced an automated tool leveraging nine sentence transformer models to perform this mapping, offering fine-grained linkage based on real-world procedures. Another comparative study [14] evaluated five feature extraction techniques (TF-IDF, LSI, BERT, MiniLM, RoBERTa) for linking CAPEC attack patterns to CVEs, resulting in a comprehensive mapping dataset that connects 133 CAPEC patterns to 685 CVEs through shared weaknesses.

While most studies focus on enhancing CVE data or linking known vulnerabilities to attacks using structured sources such as MITRE's CVE, CAPEC, or ATT&CK, a different approach has been proposed: predicting CVEs directly from real-world cyberattack news. This task is particularly challenging due to the unstructured nature of news text and the frequent absence of explicit technical details. Recent work explores the automatic linking of attack news reports to CVE-IDs using sentence transformer models, offering a novel direction to improve vulnerability awareness and incident response.

7. Conclusion and Future Work

In this study, we introduced an innovative approach for predicting software vulnerabilities directly from unstructured cyberattack news reports using the MPNet sentence transformer model. Our proposed method generates a ranked list of top-K CVE predictions based on the semantic similarity between attack descriptions and vulnerability reports. To ensure a robust evaluation, we developed four validation strategies: threshold-based detection, manual expert review, similarity with the first mentioned CVE, and similarity with all mentioned CVEs. Our results highlight the model's effectiveness, achieving validation accuracy of up to 81% depending on the strategy employed, confirming the practical feasibility of mapping attack narratives to vulnerability databases such as the CVE repository. Additionally, we assessed our model using a dataset of 100 cybersecurity news articles, analyzing each to determine the relevance of the predicted CVEs. This evaluation methodology, which combined threshold analysis, semantic similarity, and manual validation, facilitated a comprehensive assessment of the model's predictive accuracy. Our approach quantified performance and revealed the model's strengths in identifying relevant vulnerabilities from unstructured textual sources.

For future research, we intend to explore the influence of the top-K parameter on prediction accuracy. While this study utilized a fixed value (e.g., $k = 20$), employing a dynamic or optimized K-value could enhance the balance between precision and recall, especially across various attack types and news formats. Additionally, our current evaluation is based exclusively on reports from SecurityWeek. To improve generalizability, we plan to expand our evaluation to include additional datasets and test our approach on varied sources beyond SecurityWeek. Moreover, we aim to establish connections between attack reports and their corresponding vulnerability codes (e.g., CWE IDs). This approach would facilitate more accurate and explainable linkages between textual descriptions of attacks and formal vulnerability databases, ultimately enhancing the validation and traceability of predicted CVEs.

Acknowledgements

This research was supported by the European Social Fund Plus (ESF+), Project ESF2f30005, CUP: B56F24000100001. The authors also gratefully acknowledge the support of the Cybersecurity Laboratory (CSLab) at the Free University of Bozen-Bolzano funded by the EFRE-FESR 2021–2027 program, project EFRE1039, CUP: I53C23001690009.

Declaration on Generative AI

During the preparation of this manuscript, the authors used Writefull for grammar and spelling checks and to improve writing style. After using this tool, the authors carefully reviewed and edited the content as necessary and take full responsibility for the final content of the publication.

References

[1] C. Point, 38% increase in 2022 global cyberattacks, 2023. https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/.

[2] C. Point, Shifting attack landscapes and sectors in q1 2024 with a 28% increase in cyber attacks globally, 2024. https://blog.checkpoint.com/research/shifting-attack-landscapes-and-sectors-in-q1-2024-with-a-28-increase-in-cyber-attacks-globally/.

[3] NVD, Nvd vulnerabilities, 2025. https://nvd.nist.gov/vuln.

[4] J. Liu, R. Zhang, W. Liu, Y. Zhang, D. Gu, M. Tong, X. Wang, J. Xue, H. Wang, Context2vector: Accelerating security event triage via context representation learning, Information and Software Technology 146 (2022) 106856.

[5] Gartner, Threat intelligence, 2025. https://www.gartner.com/en/documents/2487216.

[6] S. Contributor, What is threat intelligence?, 2025. https://www.dnsstuff.com/what-is-threatintelligence.

[7] MITRE, Attack, 2025. https://attack.mitre.org/.

[8] MITRE, Cve, 2025. https://cve.mitre.org/.

[9] S. Elder, N. Zahan, R. Shu, M. Metro, V. Kozarev, T. Menzies, L. Williams, Do i really need all this work to find vulnerabilities? an empirical case study comparing vulnerability detection techniques on a java application, Empirical Software Engineering 27 (2022) 154.

[10] R. Othman, B. Rossi, B. Russo, Cybersecurity defenses: Exploration of cve types through attack descriptions, in: 2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2024, pp. 415–418.

[11] MITRE, Cwe dataset, 2025. https://cwe.mitre.org/.

[12] MITRE, Capec, 2025. https://capec.mitre.org/.

[13] F. Lonetti, A. Bertolino, F. Di Giandomenico, Model-based security testing in iot systems: A rapid review, Information and Software Technology (2023) 107326.

[14] R. Othman, B. Rossi, B. Russo, A comparison of vulnerability feature extraction methods from textual attack patterns, in: 2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2024, pp. 419–422.

[15] R. T. Othman, Vulnerability detection for software-intensive system, in: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, 2024, pp. 510–515.

[16] R. Othman, D. Rimawi, B. Rossi, B. Russo, From attack descriptions to vulnerabilities: A sentence transformer-based approach, Journal of Systems and Software (2025) 112615.

[17] SecurityWeek, News vulnerabilities, 2025. https://www.securityweek.com/category/vulnerabilities/.

[18] R. Othman, Vuldat- vulnerability dataset, 2025. Accessed: Feb 2, 2025. figshare.Dataset.https://doi.org/10.6084/m9.figshare.25828102.v1.

[19] R. Othman, Att&ck2vul - automated vulnerability detection from cyberattack text, 2025. Accessed: Feb 2, 2025. https://github.com/ref3t/Attack2VUL/tree/main.

[20] T. Armerding, Cve definitions, 2025. https://www.csoonline.com/article/3204884/what-is-cve-its-definition-and-purpose.html.

[21] S. Alevizopoulou, P. Koloveas, C. Tryfonopoulos, P. Raftopoulou, Social media monitoring for iot cyber-threats, in: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), IEEE, 2021, pp. 436–441.

[22] H. Gasmi, J. Laval, A. Bouras, Information extraction of cybersecurity concepts: An lstm approach, Applied Sciences 9 (2019) 3945.
[23] D. Iorga, D.-G. Corlatescu, O. Grigorescu, C. Sandescu, M. Dascalu, R. Rughinis, Yggdrasil—early detection of cybernetic vulnerabilities from twitter, in: 2021 23rd International Conference on Control Systems and Computer Science (CSCS), IEEE, 2021, pp. 463–468.
[24] A. L. Queiroz, S. Mckeever, B. Keegan, Eavesdropping hackers: Detecting software vulnerability communication on social media using text mining, in: The Fourth International Conference on Cyber-Technologies and Cyber-Systems, 2019, pp. 41–48.
[25] K. Baccar, Automated mapping of CVE vulnerabilties to MITRE ATT&CK Framework, Ph.D. thesis, Tekup, 2021.
[26] N. Dionísio, F. Alves, P. M. Ferreira, A. Bessani, Cyberthreat detection from twitter using deep neural networks, in: 2019 international joint conference on neural networks (IJCNN), IEEE, 2019, pp. 1–8.
[27] W. Tang, M. Tang, M. Ban, Z. Zhao, M. Feng, Csgvd: A deep learning approach combining sequence and graph embedding for source code vulnerability detection, Journal of Systems and Software 199 (2023) 111623.
[28] M. Catillo, A. Del Vecchio, A. Pecchia, U. Villano, A critique on the use of machine learning on public datasets for intrusion detection, in: Quality of Information and Communications Technology: 14th International Conference, QUATIC 2021, Algarve, Portugal, September 8–11, 2021, Proceedings 14, Springer, 2021, pp. 253–266.
[29] L. Regano, D. Canavese, L. Mannella, A privacy-preserving approach for vulnerability scanning detection, in: ITASEC 2024: 8th Italian Conference on Cyber Security, 2024.
[30] NVD, Cvss, 2025. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.
[31] R. Othman, B. Russo, Vuldat: Automated vulnerability detection from cyberattack text, in: Embedded Computer Systems: Architectures, Modeling, and Simulation: 23rd International Conference, SAMOS, 2023.
[32] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, X. Niu, Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources, in: Proceedings of the 33rd annual computer security applications conference, 2017, pp. 103–115.
[33] G. Wang, P. Liu, J. Huang, H. Bin, X. Wang, H. Zhu, Knowcti: Knowledge-based cyber threat intelligence entity and relation extraction, Computers & Security 141 (2024) 103824.
[34] H. Jo, Y. Lee, S. Shin, Vulcan: Automatic extraction and analysis of cyber threat intelligence from unstructured text, Computers & Security 120 (2022) 102763.
[35] H. Face, multi-qa-mpnet-base-dot-v1, 2024. Accessed: May 2, 2024. https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-dot-v1.
[36] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Advances in neural information processing systems 33 (2020) 16857–16867.
[37] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[38] R. Othman, Predicting known vulnerabilities from attack descriptions using sentence transformers, 2026. URL: https://arxiv.org/abs/2602.22433. arXiv:2602.22433.
[39] O. Okonkwo, A. Dridi, E. Vakaj, Leveraging word embeddings and transformers to extract semantics from building regulations text, in: Proceedings of the 11th Linked Data in Architecture and Construction Workshop, 2023.
[40] M. Siino, I. Tinnirello, M. La Cascia, Is text preprocessing still worth the time? a comparative survey on the influence of popular preprocessing methods on transformers and traditional classifiers, Information Systems 121 (2024) 102342.
[41] K. Hiniduma, S. Byna, J. L. Bez, Data readiness for ai: A 360-degree survey, ACM Comput. Surv. 57 (2025). URL: https://doi.org/10.1145/3722214. doi:10.1145/3722214.
[42] Semantic textual similarity, 2024. https://github.com/UKPLab/sentence-transformers/blob/master/examples/sentence_transformer/training/sts/README.md.
[43] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in:

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019.
[44] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, Mteb: Massive text embedding benchmark, arXiv preprint arXiv:2210.07316 (2022).
[45] A. Lobo, P. Oliveira, P. Sampaio, P. Novais, Cost-sensitive learning and threshold-moving approach to improve industrial lots release process on imbalanced datasets, in: International Symposium on Distributed Computing and Artificial Intelligence, Springer, 2022, pp. 280–290.
[46] V. S. Sheng, C. X. Ling, Thresholding for making classifiers cost-sensitive, in: Aaai, volume 6, 2006, pp. 476–481.
[47] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.
[48] D. I. Sjøberg, G. R. Bergersen, Construct validity in software engineering, IEEE Transactions on Software Engineering 49 (2022) 1374–1396.
[49] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, et al., Experimentation in software engineering, volume 236, Springer, 2012.
[50] A. Kuppa, L. Aouad, N.-A. Le-Khac, Linking cve's to mitre att&ck techniques, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, 2021, pp. 1–12.
[51] J. Sun, Z. Xing, H. Guo, D. Ye, X. Li, X. Xu, L. Zhu, Generating informative cve description from exploitdb posts by extractive summarization, arXiv preprint arXiv:2101.01431 (2021).
[52] Y. Lakhdhar, S. Rekhis, Machine learning based approach for the automated mapping of discovered vulnerabilities to adversial tactics, in: 2021 IEEE Security and Privacy Workshops (SPW), IEEE, 2021, pp. 309–317.
[53] O. Grigorescu, A. Nica, M. Dascalu, R. Rughinis, Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques, Algorithms 15 (2022) 314.
[54] B. Ampel, S. Samtani, S. Ullman, H. Chen, Linking common vulnerabilities and exposures to the mitre att&ck framework: A self-distillation approach, arXiv preprint arXiv:2108.01696 (2021).
[55] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[56] K. Kanakogi, H. Washizaki, Y. Fukazawa, S. Ogata, T. Okubo, T. Kato, H. Kanuka, A. Hazeyama, N. Yoshioka, Tracing cve vulnerability information to capec attack patterns using natural language processing techniques, Information 12 (2021) 298.
[57] K. Kanakogi, H. Washizaki, Y. Fukazawa, S. Ogata, T. Okubo, T. Kato, H. Kanuka, A. Hazeyama, N. Yoshioka, Comparative evaluation of nlp-based approaches for linking capec attack patterns from cve vulnerability information, Applied Sciences 12 (2022) 3400.
[58] E. Hemberg, A. Srinivasan, N. Rutar, U.-M. O'Reilly, Sourcing language models and text information for inferring cyber threat, vulnerability and mitigation relationships (2022).
[59] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the first instructional conference on machine learning, volume 242, Citeseer, 2003, pp. 29–48.
[60] J. H. Lau, T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, arXiv preprint arXiv:1607.05368 (2016).