IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 October 2024, accepted 2 November 2024, date of publication 11 November 2024,
date of current version 19 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3495663

RESEARCH ARTICLE

V2TSA: Analysis of Vulnerability to Attack Techniques Using a Semantic Approach

DO-YEON KIM, SEONG-SU YOON, AND IECK-CHAE EUOM, (Member, IEEE)
System Security Research Center, Chonnam National University, Gwangju 61186, South Korea

Corresponding author: Ieck-Chae Euom (iceuom@jnu.ac.kr)

This work was supported in part by Korea Internet & Security Agency (KISA)—Information Security Specialized University Support Project (50%); and in part by the Nuclear Safety Research Program through Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC), Republic of Korea (50%) under Grant 2106061.

ABSTRACT In recent years, vulnerabilities in industrial control systems have increased substantially. The operational environment's availability constraints hinder penetration testing from the attacker's perspective as a viable vulnerability management method, thereby limiting the ability to map attack flows fully. To address this, research has been focused on understanding attack techniques by analyzing vulnerability descriptions that detail the attack flow of these vulnerabilities. However, existing research faces the challenge of not fully capturing the overall meaning of sentences, as it relies on word embedding-based learning for vulnerability information. This study proposes the V2TSA model, which uses a semantic approach to extract attack technique information from vulnerability descriptions. Additionally, the study seeks to identify the most efficient attack techniques by applying a threshold of at least 10% for the similarity probability between vulnerability and attack technique descriptions. Compared to expert analysis, the proposed model effectively identifies specific attack paths associated with vulnerabilities. Moreover, the vulnerability attack information can be leveraged to implement appropriate detection and mitigation strategies.

INDEX TERMS Vulnerability analysis, MITRE ATT&CK, operational technology, adversary-centric, attack detection, attack mitigation.

I. INTRODUCTION

The number of vulnerabilities identified in industrial control systems has been increasing at an average annual growth rate of 25.32%.

Organizations implement mitigation policies from a defender's perspective by utilizing existing resources such as Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), and Common Vulnerability Scoring System (CVSS) [1].

However, due to the emphasis on availability in operational environments, vulnerability management from an attacker's perspective, such as penetration testing, is not utilized [2]. This limits understanding of the complete attack flow, including the stages of exploitation and subsequent impact [3].

Applying existing defense mechanisms without a thorough understanding of attack flows can lead to significant damage, especially in operational technology (OT) environments such as critical national infrastructure.

Recognizing this risk, 96% of decision-makers in most organizations consider it important to identify potential cyber attackers.

Consequently, vulnerability management from an attacker's perspective is becoming increasingly crucial. In particular, identifying exploit paths has emerged as a key challenge.

To address this, organizations employ methods to identify attackers within their operational environments based on vulnerability information. Recent research has focused on extracting attack-related information from vulnerability descriptions.

Some studies have focused on extracting attacker-related information by leveraging existing data. As illustrated

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

in Fig. 1, these approaches combine knowledge-based graphs with expert analysis techniques to achieve this extraction.

However, this approach has limitations in terms of time consumption when processing large volumes of vulnerability information.

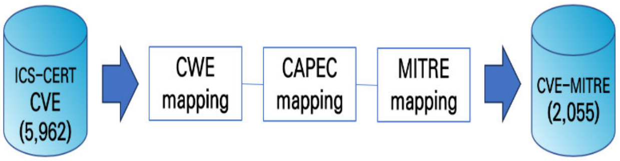


FIGURE 1. **Utilizing existing data for mapping vulnerabilities with attack techniques.**

To address time constraints, some studies have used AI algorithms. However, these face challenges due to training data limitations, failing to capture the complexity of vulnerability descriptions. These limitations have resulted in two main problems. First, there is an inability to accurately identify sentence components and their roles within vulnerability descriptions. Second, this leads to ineffective associations with specific attack techniques.

To overcome this challenge, this study proposes the Vulnerability to Attack Techniques using a Semantic Approach model. This model identifies MITRE ATT&CK framework techniques by performing semantic analysis on CVE description information, which provides detailed vulnerability data. The proposed model employs a semantic role labeling (SRL) algorithm in conjunction with a deep learning model to achieve this analysis.

The contributions of this study are as follows:

- **Semantic Analysis for Attack Technique Identification**: The proposed model uses SRL to identify attack techniques from CVE descriptions, accurately determining attack paths.
- **Integration with MITRE ATT&CK Framework**: By Linking CVE data with MITRE ATT&CK techniques enhances security detection by providing detection and response strategies.
- **Semantic Structure Analysis of Vulnerability Descriptions**: This model enhances analysis of attack techniques by understanding semantic structures beyond keyword detection.
- **Identification of Attack Paths and Response Strategies**: By analyzing similarities between vulnerabilities and attack techniques, the model identifies optimal attack paths, enhancing detection and response strategies.

II. BACKGROUND

This study details the vulnerability and attacker-related information to be utilized. Additionally, it explains the methods used to link vulnerability and attack technique information, highlighting the benefits that can be gained from this connection.

A. COMMON VULNERABILITY INFORMATION BASED ON TEXT

The Common Vulnerabilities and Exposures (CVE) list is a public repository of computer security vulnerabilities. It is managed by the MITRE Corporation under the supervision of the Cybersecurity and Infrastructure Security Agency (CISA) of the U.S. Department of Homeland Security. CVE data is provided in database format through sources such as the National Vulnerability Database (NVD) and the Computer Emergency Response Team/Coordination Center (CERT/CC).

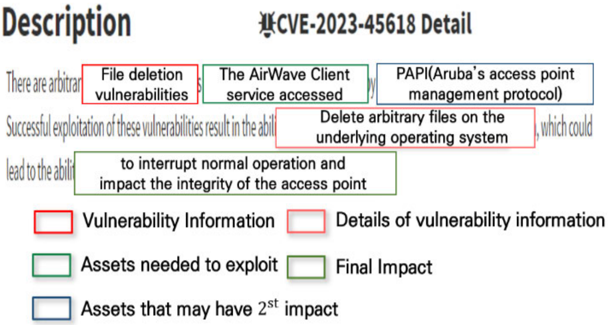

Description
CVE-2023-45618 Detail
There are arbitrar
File deletion vulnerabilities
The AirWave Client service accessed
PAPI(Aruba's access point management protocol)
Successful exploitation of these vulnerabilities result in the abil
Delete arbitrary files on the underlying operating system
, which could
lead to the abil
to interrupt normal operation and impact the integrity of the access point
Vulnerability Information
Details of vulnerability information
Assets needed to exploit
Final Impact
Assets that may have 2st impact

FIGURE 2. **Additional information in the CVE description.**

The NVD's CVE details include publication and modification dates, a description of the vulnerability, information on the CVSS, reference data, and details about the weakness and related assets. The vulnerability description, as unstructured data, covers aspects like attacker types, impacts, and attack vectors. As shown in Fig. 2, this description serves as a basis for integrating additional data [4].

This information is essential for identifying the required pre-attack and post-exploitation processes, uniquely linking vulnerabilities to specific versions of software systems, applications, or components. Analyzing unstructured data in this context involves a detailed understanding of each component's role within a sentence.

For instance, if a keyword like "File Deletion" is used without considering the role of sentence components, one might overlook crucial vulnerability-related details, such as "delete arbitrary files on -" and erroneously identify attack techniques based solely on the keyword.

Relying on keyword-based linkage can be misinterpreted as an evasion tactic when analyzing attack techniques. However, identifying the components related to the vulnerability's specifics makes it possible to assess the attack technique's impact on system availability more accurately.

B. TACTICS, TECHNIQUES AND PROCEDURES OF ATTACKER

The MITRE ATT&CK framework categorizes global cyber attacks, detailing tactics, techniques, procedures (TTPs), associated groups, software, and detection methods [5].

This information can reveal additional attack vectors, provide mitigation strategies, and offer detection insights when attack techniques are identified from existing vulnerability data. For example, Fig. 3 illustrates documents and projects associated with the ATT&CK framework.

One notable initiative is the "CVE to Attack" project [6], available on GitHub. In this project, experts analyze CVE descriptions to identify associated techniques.

The U.S. National Security Agency has also released the "Technique Cyber Threat Framework" [7], which helps prioritize mitigation efforts across 14 different tactics.

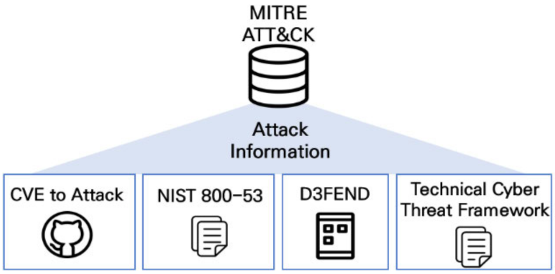


FIGURE 3. Related documents and projects based on the MITRE ATT&CK framework.

Furthermore, studies such as D3FEND [8], also developed by the MITRE Corporation, detail defensive cyber techniques using over 500 examples from the U.S. Patent Office. This catalog systematically maps defensive techniques to corresponding attack methods. MITRE also offers CSV files that link TTPs with the NIST 800-53 standard, making these resources publicly available.

The adoption of MITRE ATT&CK-based documents and projects is steadily increasing. Notably, identifying TTPs based on vulnerability information, traditionally performed manually through established methodologies, is increasingly automated through artificial intelligence.

C. NEED FOR IDENTIFYING TECHNIQUES BASED ON VULNERABILITY INFORMATION

From a defender's perspective, vulnerability reports often provide limited information because they fail to capture the attacker's detailed attack vectors and potential impacts.

Therefore, by identifying the techniques utilized by attackers based on descriptions provided in standard vulnerability information, defenders can gain several critical benefits:

- *Providing Vulnerability Detection Methods [9]:* As attack tactics diversify, scenarios become virtually infinite. The MITRE ATT&CK framework provides data for detecting specific tactics. Linking this with vulnerabilities helps identify related attack techniques and data sources, aiding in precise attack path identification. This process assists in detecting attacks and identifying alternative paths that exploit vulnerabilities, potentially revealing new vulnerabilities.
- *Providing Vulnerability Mitigation Methods [10]:* After vulnerability detection, implementing appropriate mitigation measures is crucial. The MITRE framework offers mitigation strategies for attack techniques linked to vulnerabilities. For example, network segmentation (M1039) mitigates the defense impairment technique T1562. While various mitigation measures exist, considering mitigation prohibitions is equally important to protect unintended targets. This approach enables effective vulnerability mitigation strategies.

D. SEMANTIC ANALYSIS OF TEXT INFORMATION

SRL is a natural language processing task that assigns roles to words and phrases within a sentence to elucidate their fundamental semantics and syntactic structure [11].

This process considers the complex relationship between sentence structure and meaning. Fig. 4 [12] illustrates SRL analyzing the sentence 'The San Francisco Examiner issued a special edition around noon yesterday.'

In SRL, part-of-speech (POS) tagging identifies and classifies the grammatical roles of words within a sentence. It assigns grammatical labels to each word using tags such as 'DT' (determiner) or 'NNP' (proper noun). This tagging system enables the application of various linguistic analysis rules, laying the foundation for deeper semantic analysis.

Through such semantic analysis, we can make judgments on complex issues like "who did what, to whom, and how it was changed." Furthermore, by accurately extracting causation and roles in this process, we can significantly enhance contextual understanding in natural language processing.

However, while SRL effectively analyzes individual sentences, it has limitations in recognizing relationships between sentences. To maximize SRL's analytical potential and achieve a broader contextual understanding, integration with additional language models is necessary.

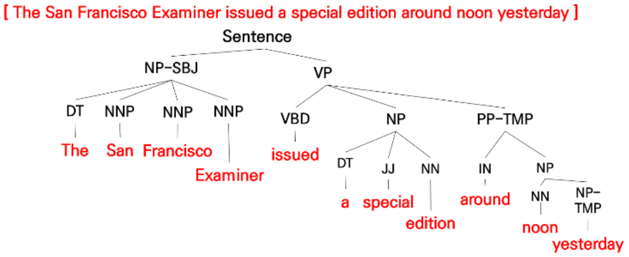


FIGURE 4. Example of semantic role labeling analysis.

III. RELATED WORK

Various studies utilize different formats, enumerations, and knowledge-based graphs to identify attack techniques based on vulnerability data. These existing studies have been classified and analyzed, as depicted in Fig. 5.

The analysis of related research in the third stage reveals three primary limitations: shortcomings in semantic analysis, difficulties in assessing the causes of outcomes, and the absence of criteria for identifying optimal attack techniques.

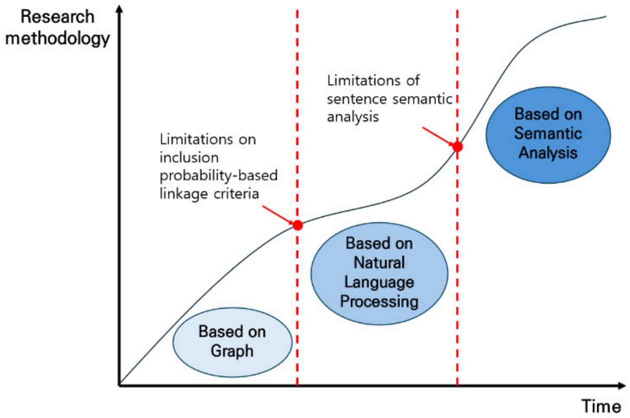

Research methodology
Limitations of sentence semantic analysis
Limitations on inclusion probability-based linkage criteria
Based on Semantic Analysis
Based on Natural Language Processing
Based on Graph
Time

FIGURE 5. Analysis of related research development.

These identified limitations underscore the need for the proposed study.

For each of these three limitations, we describe the background of the study from which the limitations were derived and the proposed study.

A. IDENTIFICATION OF ATTACK TECHNIQUE BASED ON GRAPH

The first step in research on linking vulnerabilities and attack techniques was done manually, not automated.

The following studies manually generated graphs using existing connections between vulnerabilities, weaknesses, attack patterns, and attack techniques.

Hemberg et al. [13] introduced a bidirectional aggregate data graph that facilitates relational path tracing between the enumerations and categorizations of CVE, CWE, CAPEC, and MITRE ATT&CK enterprise tactics and techniques.

The implementation of this graph database is based on data extracted from the existing CVE-MITRE information available on GitHub. The BRON graph enables tracing the path from an attacker's tactical goals to the target information involved in the attack and vice versa.

Varkas and Llias [14] proposed a knowledge-based graph that utilizes a interconnected knowledge base graph of CVE-CWE-CAPEC-MITRE ATT&CK to achieve this.

Lukas et al. [15] proposed a study to address the limitation of insufficient data when connecting CVE to MITRE using publicly available information.

This approach is achieved through exploratory linking, where the association between CAPEC and MITRE ATT&CK is clarified by referring to threat reports to analyze the relationship.

Like these studies, This graph-based method connects textual information using pre-existing data or manual analysis instead of semantic analysis. Utilizing resources like CWE and CAPEC to integrate CVE information into the MITRE framework facilitates a causal analysis of the relationships between attack techniques.

B. IDENTIFICATION OF ATTACK TECHNIQUE BASED ON NATURAL LANGUAGE PROCESSING

Several studies have been conducted to identify MITRE ATT&CK techniques by applying artificial intelligence to vulnerability-related information.

These studies utilized CVE descriptions and related data, such as vulnerability reports, threat analyses, and datasets provided by the European Union Agency for Cybersecurity (ENISA).

Their findings consistently align with the tactical and technical information outlined in the MITRE ATT&CK framework.

Lakhdhar et al. [16] leverages explicitly vulnerability-related information, including CVSS scores, vulnerability integrity, confidentiality, impact on availability, CWE, and affected products, among other factors. The AI model was trained using the ENISA dataset.

This study leverages vulnerability-related information to identify tactics from the MITRE ATT&CK framework for the enterprise.

Gionanidis et al. [17] identifies attackers' techniques based on CVE description information. The study addresses the issue of insufficient labeled training data by employing transfer learning through universal language model fine-tuning (ULMFiT).

Similarly, Grigorescu et al. [18] conducted a study that identified attack techniques based on CVE description information. This study utilized a dataset provided by the "mapping MITRE ATT&CK to CVEs for impact methodology" and 993 manually labeled CVEs.

Machine learning classifiers such as NB and SVC were employed, while deep learning models include a Convolutional neural network (CNN) with Word2Vec and two versions of BERT.

Additionally, the study employed the local interpretable model-agnostic explanations (LIME) technique to elucidate the extracted values, addressing the black-box limitations of AI models.

Domschot et al. [19] is another study that identifies attacker tactics using ransomware threat reports and data from the Reports Classification by Adversarial Tactics and Techniques (rcATT) GitHub repository.

The first approach involves selecting features based on the information content of specific phrases or words for a given class. It also considers the information gained from other variables when certain variables are known.

Kuppa et al. [20] conducted a study that utilized 690 cybersecurity articles, 63,720 vulnerability reports, and 37,000 threat reports.

Initially, the CVEs were contextualized using BiLSTM and subsequently embedded with Word2Vec. Additionally, techniques were employed to label characters and word tokens within the CVE information. Finally, feature extraction was performed to account for the complex relationships and sequences in the text data.

In the second layer, vector conversion of CVE text data was performed, and encoders containing information related to each ATT&CK technique, CVE mitigation details, and attack scenario information were incorporated.

Branescu et al. [21] identified MITRE ATT&CK tactic information from CVE description data. This study involved learning the connection between existing CVE IDs and Tactics, followed by determining the similarity of the corresponding CVE descriptions. The process utilized a transformer encoder architecture and large language models (LLMs) to perform natural language processing on textual information. Due to the existing training data imbalance, fine-tuning was performed using the F1-Score as the evaluation metric.

These studies utilized natural language processing methods and artificial intelligence models to extract relevant attack information from CVE data inputs. Various performance measures and countermeasures were employed to achieve this goal.

C. IDENTIFICATION OF ATTACK TECHNIQUE BASED ON SEMANTIC ANALYSIS

The next phase of research involved adding a new approach to analyzing the sentences in the studies mentioned above related to automatic identification of attack techniques.

These studies are based on semantic analysis, which identifies the components and roles of sentences, rather than the usual similarity checks.

MITRE Engenuity [22] employs the Delphi method on CVE descriptions to identify MITRE ATT&CK for enterprise attack techniques corresponding to three impact and technique classifications: "vulnerability type," "function," and "exploitation technique."

This process involves security experts analyzing the meaning of CVE information to identify appropriate attack techniques. Consequently, experts semantically analyze all the sentences and identify specific attack techniques when certain keywords are present. This project advances by linking specific words to specific attack techniques.

Aghaei et al. [23] explored the relationship between vulnerability information from CVE descriptions, threat actions, and techniques described in the ATT&CK TTP framework.

They utilized SRL techniques to extract subject-verb-object (SVO) structures from CVE descriptions and defined feature classes based on these available in MITRE Engenuity. These feature classes were used to create a learning model of feature-relevant sentence component pairs. The features identified in the model were subsequently associated with the MITRE ATT&CK TTP.

Soltani et al. [24] proposed a method to compare the performance of how well LLMs understand the cyber-attack domain by associating vulnerabilities with attack techniques.

Through expert analysis, they collected relationship data labeled with CVE, MITRE ATT&CK TTP, and vulnerability-related information. They then used language models to extract embedding values and analyzed cosine similarity. The method's effectiveness was verified by comparing the final identified values with those identified from the existing expert analysis.

Abdeen et al. [25] conducted a study that connects the textual components of CVE descriptions with MITRE technique information by performing semantic analysis and linking them based on sentence similarity. This was done by training a model on sentences related to an attack to derive embedding values from the CVE descriptions.

The probability values for all techniques derived from the CVE were then calculated, and only the top three techniques were identified as the result.

The study also mentioned that a causal analysis of the results was possible but did not perform such an analysis.

All four studies identified MITRE ATT&CK techniques by performing semantic analysis using expert methods or SRL techniques based on CVE information. While this approach addresses the limitations of traditional keyword-based identification, it also has drawbacks.

D. LIMITATION OF EXISTING STUDIES ON THE IDENTIFICATION OF ATTACK TECHNIQUE

Table 1 presents a comparative analysis of studies that identified attack techniques based on the three approaches discussed earlier.

All three of the perspectives of research link attack techniques to specific vulnerabilities.

The graph-based research conducted in the first phase was able to link vulnerability information and attack pattern information together, so it was possible to analyze the cause of attack technique identification in most cases.

However, there are limitations in identifying attack techniques for new vulnerabilities because the research was conducted using only existing data.

This means that automatic linking is not possible, and there are limitations in finding the optimal linking criteria through semantic analysis and inclusion probability values between sentences in natural language processing.

Then, in the second step, a natural language processing-based study that utilizes artificial intelligence to perform automatic attack technique identification is able to identify the corresponding attack technique when a new vulnerability is entered.

However, it suffers from the black box problem, which is a fundamental limitation of AI, and the limitation of sentence semantic analysis due to the general preprocessing.

It also has the limitation that the probability between vulnerability and attack technique is derived, but there is no optimal threshold to identify the attack technique.

Finally, the study performs semantic analysis, which is a limitation mentioned in the previous two related studies. This study performs semantic analysis to identify attack techniques more accurately when analyzing sentences using AI.

TABLE 1. Existing study analysis on the identification of attack technique.

Study	Identification Method	SEMANTIC ANALYSIS	Cause Analysis	Criteria for Optimal Attack Technique Identification
Erik[13]	Graph	○	●	○
Llias[14]	Graph	○	●	○
Lukas[15]	Graph	○	○	○
Erik[13]	Natural Language Processing	○	○	○
Emmanouil[17]	Natural Language Processing	○	○	○
Octavian[18]	Natural Language Processing	○	●	○
EVA[19]	Natural Language Processing	○	○	○
Aditya[20]	Natural Language Processing	○	◐	○
Ioana[21]	Natural Language Processing	○	○	○
Engenuity[22]	Semantic	●	○	○
Ehsan[23]	Semantic	●	○	○
Arian[24]	Semantic	●	○	○
Basel[25]	Semantic	●	◐	○

(○: Not applicable, ◐: Mentioned only, ●: Applicable)

However, the fundamental black box limitation of AI is not solved, and there is a limitation that the identification threshold for the probability value of the attack technique is not clear at all stages.

In this study, we aim to conduct semantic analysis to identify accurate attack techniques based on vulnerability description information, derive a clear basis for the resulting values by solving the black box problem, and set a threshold for the probability values derived for optimal attack technique identification.

The existing study proposes a method that employs SRL for the semantic analysis of sentences. It extracts the probability of each attack technique and the associated vulnerability description information to perform a causal analysis of the final results, thereby addressing these limitations.

IV. STUDY ON ALGORITHM MODELING FOR IDENTIFYING ATTACK TECHNIQUES BASED ON SEMANTIC ANALYSIS

In this study, we propose an algorithm to model the identification of MITRE ATT&CK for enterprise techniques based on textual vulnerability information. We have named this method the Vulnerability to Attack Techniques using a Semantic Approach (V2TSA) model. The summary process is outlined in Fig. 6, and the detailed process is depicted in Fig. 7.

A. DATA GENERATION BASED ON SEMANTIC ANALYSIS

1) DATA COLLECTION

In this study, we utilize three types of data: CVE information, MITRE ATT&CK for Enterprise Techniques & Procedures.

First, to collect vulnerabilities related to industrial control systems (ICS), we performed web scraping of ICS-CERT advisories from 2016 to February 2024, resulting in 5,962 data entries [26]. However, since ICS-CERT advisories do not provide CVE description information, we used the National Vulnerability Database (NVD) data feed [27] provided by NIST to obtain CVE descriptions.

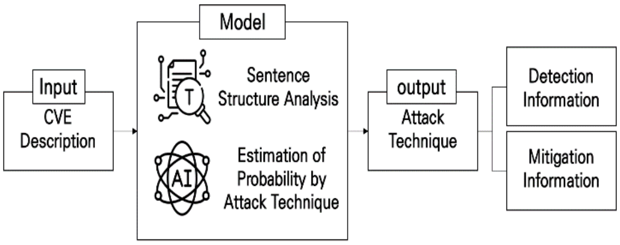


FIGURE 6. Process of identifying attack technique based on vulnerability information.

The second type of data involves techniques from MITRE ATT&CK for Enterprise. These data are available in Excel format from the MITRE Corporation's 'ATT&CK Data & Tools' page [28].

The current version, v15.1 [29], includes details such as technique ID, technique name, sub-technique ID and name, detection information, related URLs, associated tactics, last modification date, and platforms. The data used in this study focus on the technique ID, name, description, and tactic information, totaling 202 attack technique entries.

Finally, we collected procedure information, which links actual attack instances from campaigns, software, and groups to technique information. Similar to the method used for collecting technical information, these data are also provided in Excel from the "ATT&CK Data & Tools" page on the MITRE Corporation website.

The Enterprise dataset includes 25 campaigns, 632 software, and 132 groups, excluding sub-technique information. 248, 4,383, and 1,214 pieces of procedure information

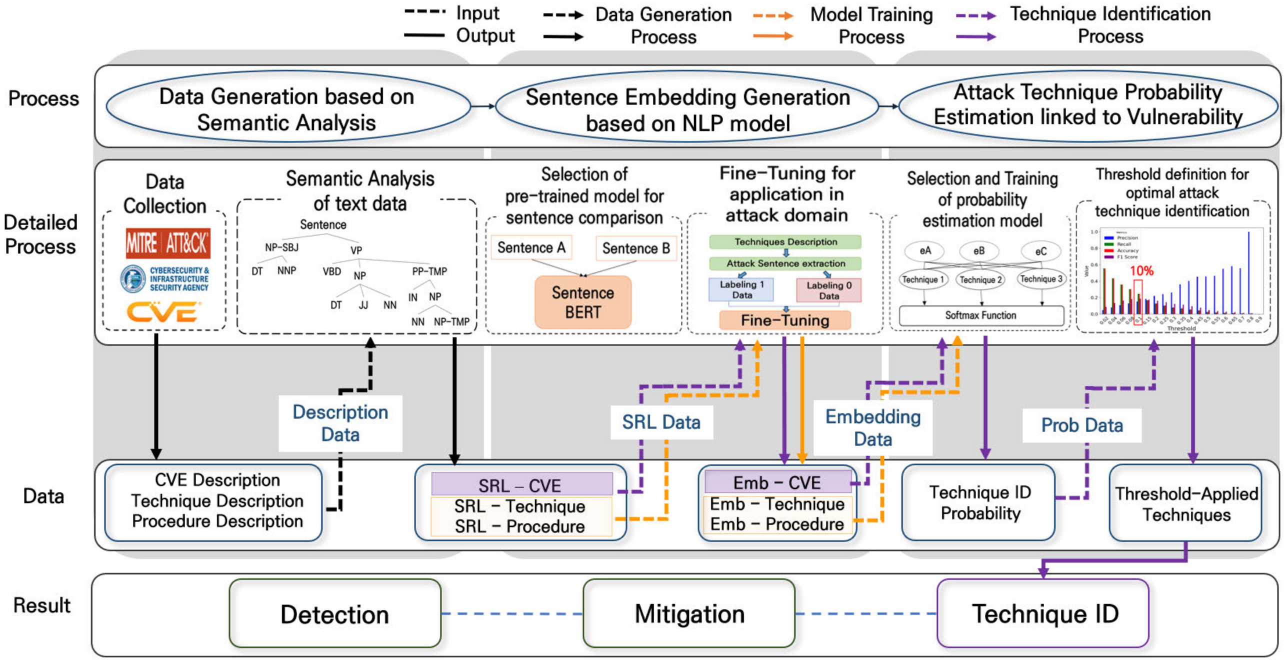


FIGURE 7. Algorithm modeling for identifying techniques based on semantic analysis.

TABLE 2. Features and detailed information of training data.

Type of Data	Feature	Description
CVE description	CVE ID	ID for ICS-CERT vulnerabilities
	CVE Description	Information on the vulnerability is provided in text format
Technique	Technique ID	ID for technique
	Technique name	Name for technique
	Technique Description	Information on the technique is provided in text format
	Tactic name	Name for tactic involving technique
Procedure	Target ID	ID for actual attack case data
	Target name	Name for actual attack case data
	Description	Information on the actual attack case data is provided in text format

were extracted from these, totaling 5,845 actual attack-based technique procedure data entries.

Table 2 summarizes these three types of learning data, detailing the column names and the information contained within.

2) SEMANTIC ANALYSIS OF TEXT DATA

Semantic analysis is performed as a preprocessing step for the collected learning data. This process involves extracting attack-related sentences using the SRL model from the input CVE descriptions. It also includes output technique descriptions and actual attack technique procedure descriptions.

The primary library used in the preprocessing process is spaCy [30], an open-source natural language processing software in Python. Additionally, semantic analysis of the sentences is conducted using the AllenNLP Model v2.10.1 SRL predictor [31].

The steps for semantic analysis of each description in this study are outlined in Algorithm 1.

The first step is "Sentence Separation," which allows each sentence to be analyzed independently by dividing the text into individual sentences using the spaCy library's sentencizer.

The second step is "SRL Extraction," which performs semantic analysis on each sentence to identify verbs and their semantic roles, assigning these roles to related elements. This step utilizes the SRL model of AllenNLP for semantic analysis.

The third step is "Verb Identifier Addition," where unique identifiers are added to each occurrence of a verb if it appears multiple times within the same sentence to distinguish them. For example, in the sentence "The user logs in and logs the error," the verb "logs" would be differentiated as "logs_1" and "logs_2."

The fourth step is "Dictionary Conversion," which involves converting the analyzed SRL results into a dictionary format.

The fifth step is "ARG0 Addition." Here, ARG0 refers to the actor related to the verb in the sentence, which helps accurately understand the sentence's meaning. For example,

Algorithm 1 Text Processing

```
function Semantic Analysis(inputText)
    sentences ← Sentence Separation(inputText)
    sentenceResults ← initialize empty dictionary
    for sentence in sentences do
        srlResults ← SRL extraction(sentence)
        srlResults ← Verb Identifier Addition(srlResults)
        srlDictionary ← Dictionary conversion(srlResults)
        srlDictionary ← ARG0 Addition(srlDictionary)
        sentenceResults[sentence] ← srlDictionary
    end for
    attackSentences ← Attack Sentence extraction(sentenceResults)
    return attackSentences
end function
function Sentence Separation(text)
    Load spaCy language model
    sentences ← use sentencizer to divide text
    return sentences
end function
function SRL extraction(sentence)
    Load AllenNLP SRL model
    srlResults ← Tagging sentence components and their role
return srlResults
end function
function Verb Identifier Addition(srlResults)
    for verb, index in srlResults do
        if verb occurs more than once then
            update verb with unique identifier (e.g., verb index)
        end if
    end for
    return srlResults
end function
function Dictionary conversion(srlResults)
    dictionaryF ormat ← converts SRL tags to the dictionary
    return srlDictionary
end function
function ARG0 Addition(srlDictionary)
    for ARG0 in srlDictionary do
        if ARG0 is missing then
            infer ARG0 based on context and add to srlDictionary
        end if
    end for
    return srlDictionary
end function
function Attack Sentence extraction(srlDictionary)
    Attacksentences ← initialize empty list
    for sentence, ARG0, verbs in srlDictionary do
        if ARG0, verbs meet attack criteria then
            add sentence to Attacksentences
        end if
    end for
    return Attacksentences
end function
```

if the verb "gain" in "The attacker uses a tool to exploit a vulnerability and gain control." does not explicitly show its subject, the context indicates "The attacker" as the subject, which is then added to ARG0.

The final step uses the analyzed results to extract attack sentences related to exploitation. This step involves filtering based on the subject, object, and verbs, using criteria frequently found in the data collected during the vulnerability and attack technique data collection stage.

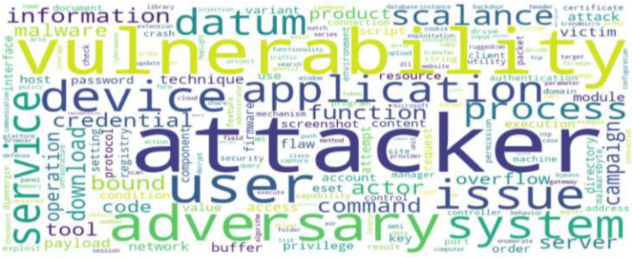


FIGURE 8. WordCloud using ARG0(subjects, objects) from information related to vulnerability and attack technique.

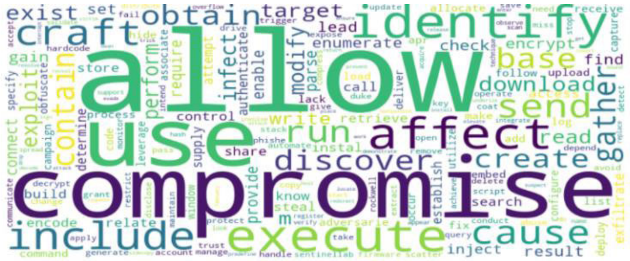


FIGURE 9. WordCloud using verbs from information related to vulnerability and attack technique.

Words such as "have", "be", "that", and "[", and "]" which do not contribute to the meaning are treated as stopwords before identifying frequently used subjects, objects, and verbs. Subsequently as shown in Fig. 8 and 9, a WordCloud [32] is used to visualize word frequencies.

The top sentences that include words like 'attacker,' 'adversary,' 'user,' 'vulnerability,' and 'device' are selected as attack vectors. Additionally, verbs such as 'allow,' 'compromise,' 'use,' 'identify,' and 'execute' are considered.

Based on these findings, Table 3 outlines the purpose and necessity of each step.

Fig. 10. shows the application of this process to an example: "Heap-based buffer overflow in FManagerService.exe in Schneider Electric Accutech Manager 2.00.1 and earlier allows remote attackers to execute arbitrary code via a crafted HTTP request.".

B. SENTENCE EMBEDDING GENERATION BASED ON NLP MODEL

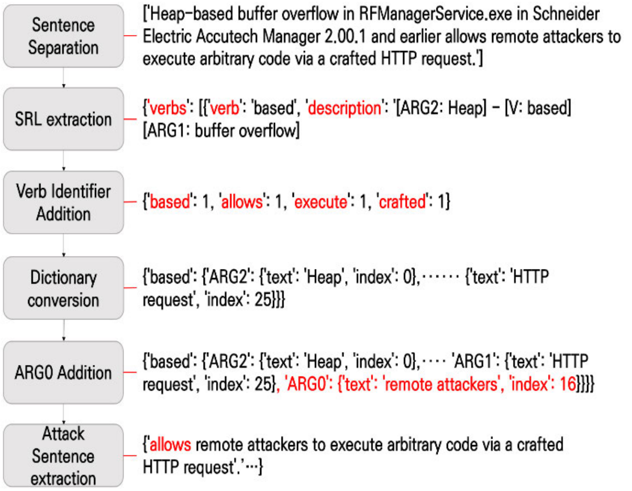
1) SELECTION OF PRE-TRAINED MODEL FOR SENTENCE COMPARISION

Bidirectional encoder representations from transformers (BERT) have demonstrated high performance in various sentence classification and pair regression tasks [33]. A cross-encoder architecture, which applies full attention to both sentences for detailed comparison, is commonly used. However, this approach is slow, making it impractical for real-time applications [34].

To address this limitation, the bi-encoder architecture is utilized. This architecture substantially reduces processing time; for example, clustering 10,000 sentences using a bi-encoder can save approximately 65 h compared to a

TABLE 3. The purpose and significance of the data preprocessing step.

Step	Purpose	Significance
Sentence Separation	Separating into individual sentences	Preventing the incorrect understanding of relationships and characteristics between sentences
SRL extraction	Perform semantic analysis	Preventing the incorrect understanding of verbs and related components and related semantic relationships in sentences
Verb Identifier Addition	Distinguishing the meaning of duplicate verbs	Preventing the incorrect understanding of information by linking companies with different actors or objects
Dictionary Conversion	ease of follow-up process	Prevent inefficient data processing due to unstructured data formats
ARG0 Addition	Explicit representation of the subject	Preventing the incorrect understanding of the subject in context about the verb
Attack Sentence extraction	Attack Sentence extraction based on phrase and clause expressions close to vulnerability information	Avoid simple keyword matching and reliance on superficial data analytics



```
['Heap-based buffer overflow in RFManagerService.exe in Schneider
Electric Accutech Manager 2.00.1 and earlier allows remote attackers to
execute arbitrary code via a crafted HTTP request.']

{'verbs': [{'verb': 'based', 'description': '[ARG2: Heap] - [V: based]
[ARG1: buffer overflow]

{'based': 1, 'allows': 1, 'execute': 1, 'crafted': 1}

{'based': {'ARG2': {'text': 'Heap', 'index': 0},······ {'text': 'HTTP
request', 'index': 25}}}

{'based': {'ARG2': {'text': 'Heap', 'index': 0},···· 'ARG1': {'text': 'HTTP
request', 'index': 25}, 'ARG0': {'text': 'remote attackers', 'index': 16}}}}

{'allows remote attackers to execute arbitrary code via a crafted
HTTP request.'···}
```

FIGURE 10. Example of data generation based on semantic analysis.

traditional cross-encoder. This substantial reduction in time makes the bi-encoder approach more suitable for practical applications where speed is a critical factor.

In this study, we apply Sentence-BERT, a model utilizing a bi-encoder structure, for training. Sentence-BERT excels in tasks requiring semantic textual similarity, such as document clustering and information retrieval.

2) FINE-TUNING FOR APPLICATION IN THE ATTACK DOMAIN

Sentence-BERT is generally trained on everyday language, limiting its ability to effectively capture the nuances and specifics of cybersecurity contexts [35]. To address this issue, we fine-tune the model using enterprise MITRE ATT&CK data. The fine-tuning process proceeds as shown in Fig. 11.

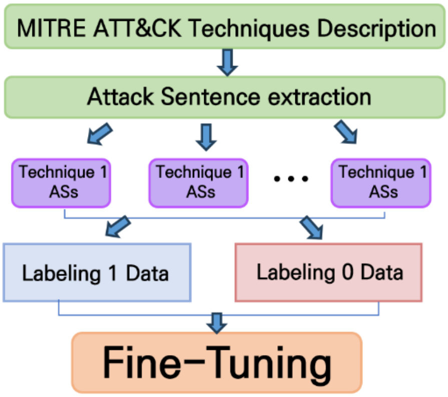


FIGURE 11. Process of Fine-Tuning for application in the attack domain.

This fine-tuning process tailors the model to better understand and embed sentences related to cyberattacks. It ensures the model is fast, efficient, and accurately responsive to the specialized needs of cybersecurity.

Before data labeling, attack sentences extracted from various attack techniques and procedures are consolidated to create a fine-tuning dataset. Two types of labeled datasets are subsequently utilized:

First, label '1' data comprises attack sentences extracted from the same technique ID (TID). Since each sentence in this set originates from the same TID, it is assumed that it shares a common attack objective and is thus labeled '1'.

The criteria for determining the appropriate set size for "label '1'" data are based on each technique's minimum and median number of attack sentences. The median is used in addition to the minimum because relying solely on the minimum (typically two sentences) would result in insufficient comparisons, hindering effective evaluation.

The range of attack sentences extracted from each technique varies from 2 to 894, with 89 different sentence lengths observed. Based on the frequency distribution, a median value of 27 sentences is calculated.

Consequently, when the number of attack sentences extracted from a single attack technique exceeds 27, they are randomly grouped into subsets of 27 sentences each.

However, attack sentences that result in fewer than 27 sentences are considered positive data. To ensure sufficient data, pairs of sentences are randomly formed, with a minimum group size of two sentences. This process is illustrated in Fig. 12.

Conversely, data labeled as '0', representing negative data, consists of attack sentences extracted from different TIDs. Each set contains 27 sentences, each originating from a different TID. Because these sentences come from distinct TIDs, they are presumed to have different attack objectives and are labeled '0'.

In this way, the MITRE ATT&CK for enterprise dataset is constructed; for fine-tuning, the dataset is balanced with a

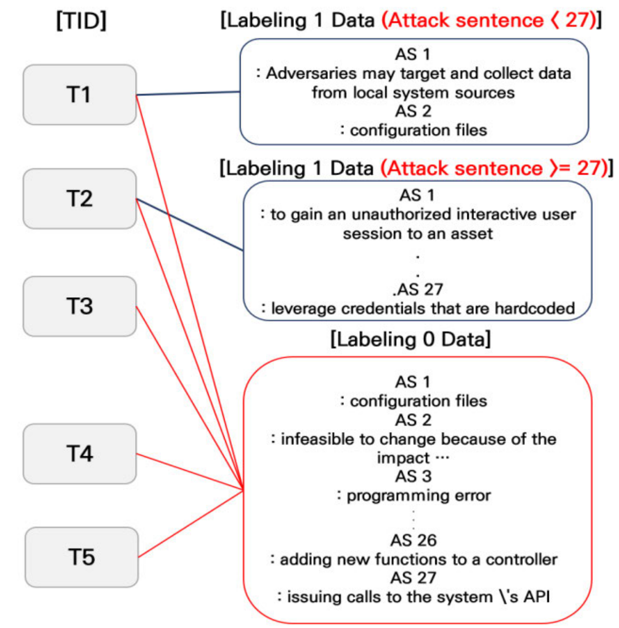

[TID]
T1
T2
T3
T4
T5
[Labeling 1 Data (Attack sentence ⟨ 27)]
AS 1
: Adversaries may target and collect data from local system sources
AS 2
: configuration files
[Labeling 1 Data (Attack sentence ⟩= 27)]
AS 1
: to gain an unauthorized interactive user session to an asset
AS 27
: leverage credentials that are hardcoded
[Labeling 0 Data]
AS 1
: configuration files
AS 2
: infeasible to change because of the impact …
AS 3
: programming error
AS 26
: adding new functions to a controller
AS 27
: issuing calls to the system \'s API

FIGURE 12. Process for generating train data of fine-tuning.

1:1 ratio, resulting in 11,775 positive data points and 11,775 negative data points from the enterprise perspective.

C. ATTACK TECHNIQUE PROBABILITY ESTIMATION LINKED TO VULNERABILITY

1) SELECTION OF PROBABILITY ESTIMATION MODEL

Using sentence embeddings extracted from the proposed BERT model, we quantitatively assess the likelihood that each embedding corresponds to one of the 202 enterprise techniques. While reviewing various models for measuring similarity, we found limitations. Traditional models generate only linear boundaries or require substantial resources to capture nonlinear relationships, leading to significant time constraints.

To address this, we employ the multinomial LR model [36] to determine the inclusion probabilities. This model is particularly suitable when the dependent variable is nominal with no inherent order or ranking. It is also ideal when more than two categories are involved.

The model calculates probability based on the following equation [37].

$$P\left(Y=k\,|\,X=x\right)=\frac{e^{\beta_{0k}}+e_k^T x}{\sum_{l=1}^{K} e^{\beta_{0l}}+e_l^T x} \qquad (1)$$

By applying this equation to the proposed probability extraction method, the algorithm generates probabilities. These correspond to the 202 attack techniques associated with vulnerabilities, as outlined in Algorithm 2.

This study proposes calculating these probabilities by deriving the inclusion probabilities of the 202 ATT&CK techniques from vulnerability description data. Table 4 briefly explains the functionality of Algorithm 2.

Algorithm 2 Predict Probability of Attack Techniques

```
function PredictTechniques(Techniqueembedding, coefficients)
    maxProbabilities ← array of length 202 initialized to 0
    for SentenceEmb in Techniqueembedding do
        sentenceProbabilities ← CalcProbs(SentenceEmb, coefficients)
        for i ← 1 to 202 do
            if sentenceProbabilities[i] > maxProbabilities[i] then
                maxProbabilities[i] ← sentenceProbabilities[i]
            end if
        end for
    end for
    return maxProbabilities
end function
function CalcProbs(SentenceEmb, coefficients)
    expValues ← empty list
    for beta in coefficients do
        dotProduct ← DotProduct(beta, SentenceEmb)
        expValue ← exp(dotProduct)
        Append(expValues, expValue)
    end for
    totalExp ← Sum(expValues)
    probabilities ← empty list
    for expValue in expValues do
        probability ← expValue/totalExp
        Append(probabilities, probability)
    end for
    return probabilities
end function
```

TABLE 4. Descriptions of functions used in the pseudocode.

Function	Input – Output	Description
Predicttechniques	attack_sentence & coefficients – max_probabilities	- Extract the maximum probability for each attack sentence from the CalcProbs function. - Extract the vulnerability inclusion probability for each attack technique
CalProbs	Sentence & coefficients – probabilities	- Use the multinomial LR model formula. - Calculate probabilities for each attack sentence for an attack technique.

In the CalcProbs function, equation (1) from the multinomial LR model is employed, with the total number of classes incorporated into the PredictTechnique function.

Table 5 presents the variables referenced in equation (1), detailing their usage and significance within the pseudocode.

TABLE 5. Descriptions of variables used in formulas and pseudocode.

Variables (Formula)	Variables (Pseudocode)	Description
$P(Y = k \mid X = x)$	Probabilities	Probability that output Y belongs to class k given input X=x
β_{0k}	beta in coefficients	LR model coefficients for class k
β_k	-	LR model coefficients for class k, with β_k as the slope vector.
x	SentenceEmb	Input feature vector
e	-	Natural constant e
X	-	Total number of output classes (202)
$\beta_k^T x$	dotProduct	Dot product of input and slope vectors
$e^{\beta_k^T x}$	expValue	Exponential function calculation
$\sum_{l=1}^{K} e^{\beta_{0l}+\beta_l^T x}$	totalExp	Sum of exponential function values

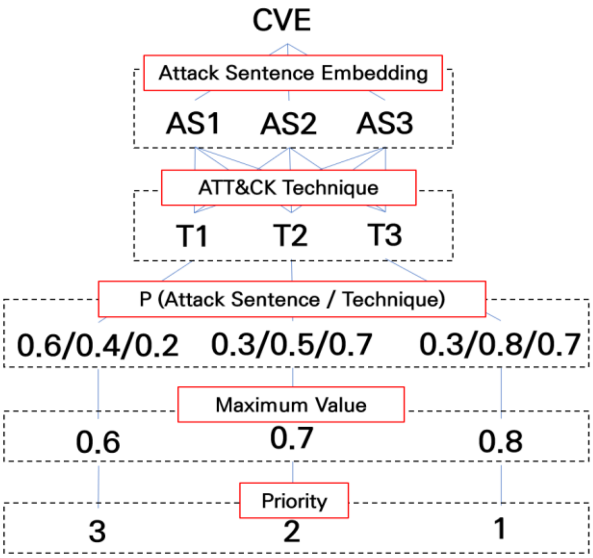


FIGURE 13. Generating attack technique probabilities based on the multinomial logistic regression model.

Based on these conditions, the probabilities derived for each attack technique ultimately determine the ranking of the 202 attack techniques, as illustrated in the process shown in Fig. 13.

2) TRAINING OF PROBABILITY ESTIMATION MODEL

The model's final output comprises probability values that indicate the likelihood of vulnerabilities associated with each of the 202 attack techniques.

Embedding training related to these techniques is essential for accurately classifying and predicting these probabilities using the embedding values derived from the sentence-BERT model.

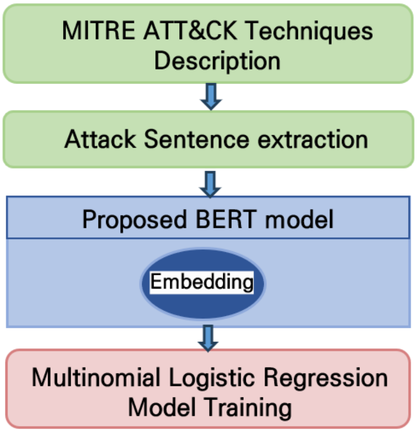

MITRE ATT&CK Techniques Description
Attack Sentence extraction
Proposed BERT model
Embedding
Multinomial Logistic Regression Model Training

FIGURE 14. Multinomial logistic regression model training process.

As illustrated in Fig. 14., this process involves adjusting the weights for each attack technique or class to account for the varying sizes of their corresponding attack sentences.

The most common method for weight adjustment is applied using equation (2):

$$weight = \frac{total\_samples}{num\_classes * class\_samples} \tag{2}$$

This formula calculates the weight by dividing the total number of samples by the product of the number of classes and the number of samples for a particular class. This weight adjustment is implemented in the source code by setting the class weight parameter.

Applying these weights during the training process considers the significance of each sample when calculating the loss function. This effectively addresses the issue of data imbalance among different classes. This approach ensures the model appropriately recognizes each attack technique's varying importance. It also responds effectively to their representation within the training data.

3) THRESHOLD DEFINITION FOR OPTIMAL ATTACK TECHNIQUE IDENTIFICATION

When embeddings from the proposed BERT model are input into the Multinomial Logistic Regression(MLR) model, values are generated. These values indicate the probability that each of the 202 attack techniques is associated with a given vulnerability. Techniques with very low probabilities are excluded from consideration.

We evaluate performance using test data to determine the most appropriate threshold for identifying relevant attack techniques. This test data comprises 835 CVE-MITRE records [38] from the MITRE Engenuity Center for Threat-Informed Defense (CTID) attack_to_cve project. Fig. 15 shows the probability distributions of the attack techniques for each of the 835 data points.

To set the threshold, we examine the probability distribution. The mean probability is 0.086, the median is 0.027, and the standard deviation is 0.224. The top 25% of data have probabilities of 0.093 or higher.

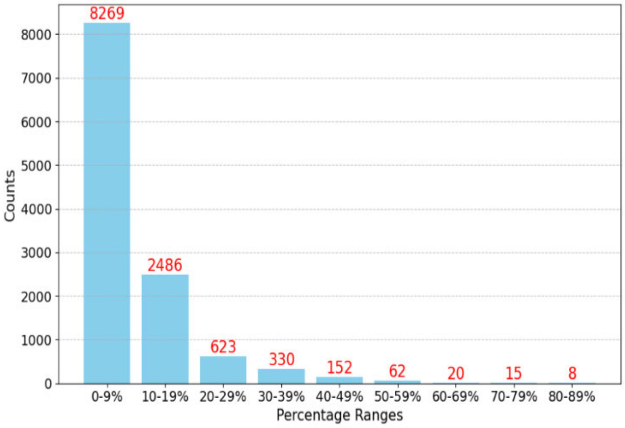


FIGURE 15. Probability distribution of attack techniques for 835 vulnerabilities.

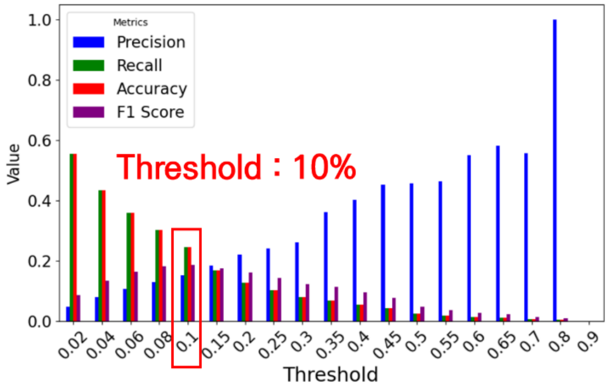


FIGURE 16. Performance comparison based on different thresholds.

Using this information, we define the range for the threshold. For probabilities below 0.1, intervals of 0.02 are used, while for probabilities above 0.1, intervals of 0.05 are applied. Precision, recall, accuracy, and F1-Score are calculated for each threshold, as shown in Fig. 16.

Based on these results, three key thresholds are considered: 0.02, 0.1, and 0.8. A threshold of 0.02 achieves the highest accuracy and recall, indicating a near perfect prediction true positives but with many false positives. A threshold of 0.1 provides the highest F1-Score, balancing the prediction of positive cases with prediction accuracy. Finally, threshold of 0.8 results the highest precision, meaning most predicted positives are true positives, though many actual positives may be missed.

Therefore, this study sets the threshold at 0.1, or 10%, to ensure a balanced outcome in predicting relevant attack techniques.

V. PERFORMANCE EVALUATION

A. METHODS FOR MODEL PERFORMANCE EVALUATION

The performance of the proposed model is assessed using two distinct approaches.

The first approach focuses on evaluating the model's performance through semantic analysis. This method determines the effectiveness of the semantic analysis applied in this study. To assess accuracy, we compare the results with the MITRE Engenuity CTID attack_to_cve project, which employs the Delphi method to analyze descriptions and identify technique information based on expert opinions.

The second approach evaluates the explainability of the SRL model's results. This approach mitigates artificial intelligence's black-box nature, a limitation frequently noted in previous studies. By analyzing the explainability of the model's outcomes, we aim to ensure that the results are both understandable and verifiable.

B. SEMANTIC ANALYSIS-BASED MODEL PERFORMANCE EVALUATION

To assess the effectiveness of the semantic analysis approach proposed in this study, we leverage the results from the MITRE Engenuity CTID attack_to_cve project. This project, which was also utilized during threshold setting for performance evaluation, employs the Delphi method to identify MITRE ATT&CK for enterprise attack techniques based on CVE descriptions.

The classification is conducted across three categories of influence: "vulnerability type," "function," and "exploitation technique" [39].

Using this project for performance evaluation allows us to verify whether the semantic model proposed in this study accurately interprets sentences, given that it is grounded in expert opinions that assess the general meaning of the sentences.

The evaluation utilizes 835 CVEs collected during the data collection phase, facilitating a comparative analysis with the CVEs employed in the project.

When assessing the overall accuracy across all CVEs, the model achieved a result of 52.83%, indicating suboptimal performance.

We adopted a different approach to refine the evaluation and focused on more critical CVEs. Specifically, we utilized CVSS scores, defining scores of 9+ as critical and scores in the 7+ range, also considered critical in OT environments, as our evaluation range [40].

The data were then divided into four categories: CVSS scores of 9+ and below 9 and CVSS scores of 7+ and below 7. The performance evaluation results for these categories are presented in Table 6.

When comparing performance across these ranges, we observed that the results for the more critical CVEs (9+ and 7+) were superior to those for the less critical CVEs (below 9 and 7).

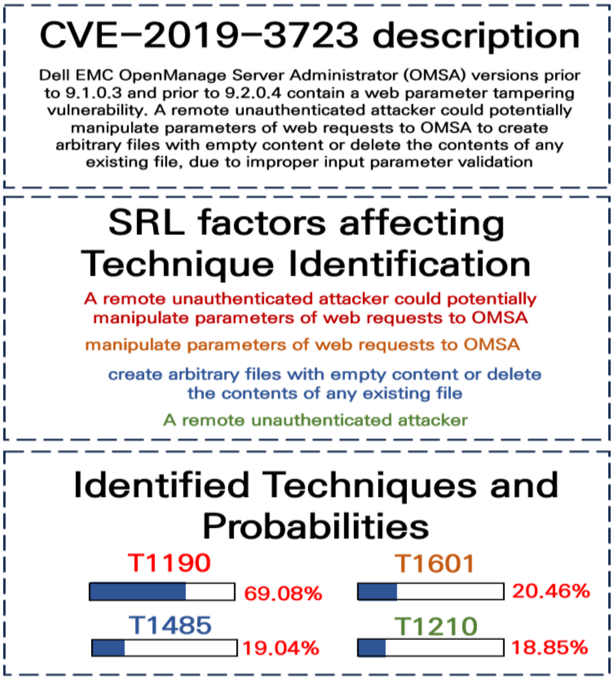
This indicates that, while the proposed model's overall performance could be much higher, it demonstrates greater effectiveness in analyzing and identifying attack techniques associated with more critical vulnerabilities.

C. ANALYZING THE CAUSE OF THE RESULT

A key challenge in using AI models for predictions is the 'black-box' problem. Previous studies have also highlighted the difficulty in identifying clear causes behind the results.

TABLE 6. Performance evaluation results by CVSS range.

CVSS Scope	Precision	Recall	Accuracy	F1-Score
CVSS ≥ 9	45.83%	28.86%	28.85%	33.68%
CVSS < 9	40.62%	27.45%	27.44%	31.29%
CVSS ≥ 7	44.97%	30.48%	30.48%	34.65%
CVSS < 7	35.41%	21.93%	21.93%	25.88%



CVE-2019-3723 description

Dell EMC OpenManage Server Administrator (OMSA) versions prior to 9.1.0.3 and prior to 9.2.0.4 contain a web parameter tampering vulnerability. A remote unauthenticated attacker could potentially manipulate parameters of web requests to OMSA to create arbitrary files with empty content or delete the contents of any existing file, due to improper input parameter validation

SRL factors affecting Technique Identification

A remote unauthenticated attacker could potentially manipulate parameters of web requests to OMSA

manipulate parameters of web requests to OMSA

create arbitrary files with empty content or delete the contents of any existing file

A remote unauthenticated attacker

FIGURE 17. Results of CVE-2019-3723 and cause analysis through SRL.

To address this, our study employs SRL analysis to investigate the underlying causes of the results. Through SRL analysis, attack sentences are extracted and then probabilistically compared with each attack technique to determine the final attack techniques.

In other words, employing SRL enables us to pinpoint the attack sentence, which is the sentence component that most substantially influences the identification of each attack technique. For instance, the cause is identified in the analysis of the results for CVE-2019-3723, as illustrated in Fig. 17.

For each CVE, the Technique information is output along with the index values of the list of Attack Sentences. By using the extracted index values, we can extract the corresponding Attack Sentences from the list, thereby performing cause analysis.

This method enables a transparent and explainable approach to understanding how each attack technique is identified, addressing the 'black box' issue and providing clear insights into the factors influencing the model's predictions.

VI. CASE STUDY

The case study compares the results obtained from the MITRE Engenuity CTID attack_to_cve project, conducted under the "Semantic Analysis-Based Model Performance Evaluation," with vulnerabilities that occurred in an operational environment.

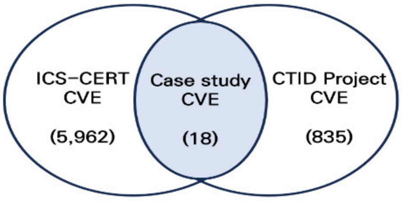


FIGURE 18. Selecting case study data.

This comparison is not based on performance metrics used in previous evaluations but focuses on manually analyzing whether the attack techniques identified from the vulnerability information are appropriate.

This manual analysis aims to determine the relevance and accuracy of the attack techniques identified by the proposed model in real-world scenarios, ensuring that the identified attack techniques are valid and applicable to the specific vulnerabilities analyzed.

A. IDENTIFY TECHNIQUE FOR OPERATIONAL TECHNOLOGY VULNERABILITY

Based on the Delphi method, we identify 18 vulnerabilities as shown in the following Fig. 18.

The comparison of the results from the existing CTID project and the proposed model for these identified vulnerabilities is presented in Table 7.

B. CONDUCTING COMPARATIVE ANALYSIS TO EVALUATE THE EFFECTIVENESS OF SEMANTIC ANALYSIS

Using the identified attack technique information, we analyzed the results of the proposed model and the existing project based on specific vulnerability descriptions. The comparative analysis employed the LLM model, ChatGPT, alongside a passive heuristic method.

This approach helps determine which model or algorithm is more effective at identifying attack techniques based on vulnerability description information. A summary of the vulnerability descriptions and their relationship to each attack vector is presented in Table 8.

In the case of CVE-2016-1409, the vulnerability description mentions a denial-of-service attack. This prompted the model proposed in this study to identify T1499 and T1498, which are related to denial-of-service attacks. Additionally, T1557 was identified by analyzing the description's meaning. This was done by comparing the probability with attack description information, even though it wasn't directly mentioned in the vulnerability description. It was determined that interception or modification of network traffic through the neighbor discovery protocol (NDP).

In the existing CTID project, T1189 is not directly related to the vulnerability description, as it pertains to accessing a system when a user visits a website.

Similarly, in the case of T1203, the vulnerability description does not explicitly mention exploiting a software vulnerability in a client application, making it challenging to observe a direct connection.

However, the existing CTID project can derive attack vectors for potential future impacts not directly outlined in the vulnerability description. This is achieved by extrapolating possible paths or impacts based on the information associated with the vulnerability.

In this context, T1189 and T1203 can be inferred to assume that a Cisco device has a web interface and that a user visits a vulnerable web page, which automatically triggers an NDP message.

As another example, the vulnerability description for CVE-2018-8835 mentions remote code execution. Our model identifies T1190, T1203, and T1210 as related techniques. Additionally, it identifies T1105 and T1036 by analyzing the description's meaning and comparing it with attack technique descriptions. T1105 is linked to processing malicious tools after remote code execution, while T1036 involves manipulating an '.mp3' file to appear legitimate.

In contrast, existing CTID projects identified T1204 and T1574. T1204 is interpreted as direct user execution of malware, which differs from the remote code execution mentioned in the vulnerability description. Additionally, T1574, described as a remote attack disrupting local system execution, may not be directly related to this vulnerability.

However, similar to the previous example, we can derive attack vectors related to subsequent impacts that are not explicitly mentioned in the vulnerability description.

From this perspective, T1204 is relevant because the same vulnerability could be exploited by a user rather than a remote attacker. Moreover, T1574 could represent an exploit technique using privileges gained from a remote attack. This technique might be used to control the execution flow in the local system. Such control could lead to subsequent impact scenarios.

In the case of CVE-2018-14819, the vulnerability description mentions remote code execution. The model proposed in this study identifies T1203 and T1210 as related techniques.

In contrast, existing CTID projects identify T1574, which involves disrupting the execution flow in the local system. This differs from the remote code execution described in the vulnerability description.

However, as with previous examples and considering the characteristics of CTID projects, T1574 could represent an attack vector for subsequent impacts where a remote attacker gains privileges to control the flow of the local system.

Finally, in CVE-2018-7499, T1210 and T1190 were identified as exploiting software vulnerabilities in remote services, as described in the vulnerability information. Additionally, attack techniques such as T1059 were also identified, as attackers can leverage these remote services to execute arbitrary code.

In contrast, existing CTID projects identified T1574 and T1499. T1574 appears unrelated as it's interpreted as disrupting execution flow in local systems. This differs from the remote code execution mentioned in the vulnerability description. Additionally, DOS attacks like T1499 are not mentioned in the vulnerability description, making it difficult to identify this attack technique from the description alone.

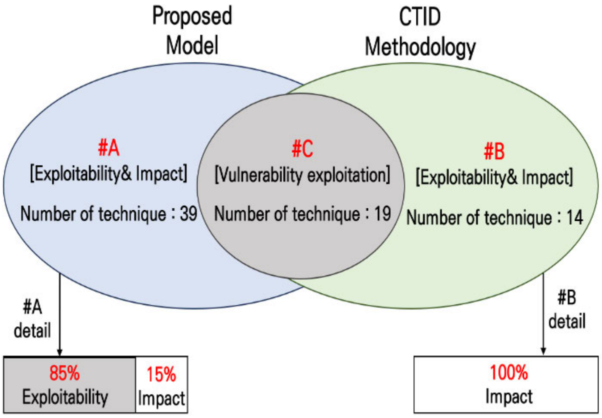


FIGURE 19. Results of CVE-2019-3723 and cause analysis through SRL.

However, T1574 could be used as an attack vector for subsequent impacts not detailed in the vulnerability description. It represents scenarios where a remote attacker gains the ability to control the local system's execution flow. Similarly, T1499 could be utilized to derive information about the vulnerability's potential impacts.

When all 18 vulnerabilities are analyzed in this way, the following summary can be seen in Fig.19.

#A represents the part of the proposed model that excludes the standard attack techniques identified in the proposed model. In contrast, #B represents the part of the CTID methodology that excludes the standard attack techniques identified in the CTID methodology. #C denotes the common attack technique identified in both models.

In this context, #C corresponds to the attack techniques explicitly mentioned in the vulnerability description and can exploit the vulnerability.

For #A and #B, in addition to the standard techniques, attack techniques that are either mentioned in the vulnerability description and can exploit the vulnerability or techniques that are not mentioned but could be applied in other ways based on a later understanding of the vulnerability's impact, are identified.

In the case of #A, which includes the attack technique identified by the proposed model, the advantage lies in identifying techniques based on the vulnerability description that can exploit the vulnerability, with a success rate of approximately 85%.

Conversely, in the case of #B, all the techniques identified in #C are recognized as impact information, except for those identified in #A. This is due to the CTID project's focus on identifying attack techniques based on the vulnerability's impact information.

This approach can determine the impact after the vulnerability is exploited and identify additional attack paths that

TABLE 7. Comparative analysis of techniques targeting OT vulnerabilities.

CVE-ID	Proposed model	CTID Methodology	Common technique
CVE-2019-13922	T1210, T1003, T1552, T1110	T1552	T1552
CVE-2020-11023	T1059, T1185, T1553	T1059, T1204, T1557	T1059
CVE-2016-5645	T1542, T1601, T1195	T1078, T1542	T1542
CVE-2016-1409	**T1499, T1498, T1557**	**T1189, T1203**	**-**
CVE-2020-12014	T1190, T1059	T1059	T1059
CVE-2020-6986	T1110, T1499, T1205	T1499	T1499
CVE-2018-8835	**T1190, T1203, T1210, T1105, T1036**	**T1204, T1574**	**-**
CVE-2018-17908	T1548, T1562, T1068	T1068, T1562	T1068, T1562
CVE-2015-7931	T1071, T1040, T1557	T1557	T1557
CVE-2017-3881	T1203, T1210, T1071, T1059, T1190	T1190	T1190
CVE-2018-10611	T1059, T1210, T1505	T1059, T1190	T1059
CVE-2018-10633	T1556, T1078	T1078	T1078
CVE-2018-14819	**T1203, T1210**	**T1574**	**-**
CVE-2020-14508	T1499, T1498, T1190, T1059	T1059, T1190, T1499	T1059, T1190, T1499
CVE-2018-7499	**T1210, T1190, T1059**	**T1499, T1574**	**-**
CVE-2015-7912	T1203, T1059, T1190	T1059, T1190	T1059, T1190
CVE-2015-0984	T1005, T1078, T1190	T1005, T1190, T1552	T1005, T1190
CVE-2020-11896	T1190, T1572, T1090, T1210	T1190, T1203, T1499	T1190

TABLE 8. Comparative analysis of vulnerabilities with no commonality to existing research.

CVE-ID	CVE description	Proposed model	CTID Methodology
CVE-2016-1409	The Neighbor Discovery Protocol (NDP) implementation in the IPv6 stack in Cisco IOS XE 2.1 through 3.17S, IOS XR 2.0.0 through 5.3.2, and NX-OS allows remote attackers to cause a denial of service (packet-processing outage) via crafted ND messages, aka Bug ID CSCuz66542, as exploited in the wild in May 2016.	T1499 - Endpoint Denial of Service T1498 - Network Denial of Service T1557 - Adversary-in-the-Middle	T1189 - Drive-by Compromise T1203 - Exploitation for Client Execution
CVE-2018-8835	Double free vulnerabilities in Advantech WebAccess HMI Designer 2.1.7.32 and prior caused by processing specially crafted .pm3 files may allow remote code execution.	T1190 - Exploit Public-Facing Application T1203 - Exploitation for Client Execution T1210 - Exploitation of Remote Services T1105 - Ingress Tool Transfer T1036 - Masquerading	T1204 - User Execution T1574 - Hijack Execution Flow
CVE-2018-14819	Fuji Electric V-Server 4.0.3.0 and prior, An out-of-bounds read vulnerability has been identified, which may allow remote code execution.	T1203 - Exploitation for Client Execution T1210 - Exploitation of Remote Services	T1574 - Hijack Execution Flow
CVE-2018-7499	In Advantech WebAccess versions V8.2_20170817 and prior, WebAccess versions V8.3.0 and prior, WebAccess Dashboard versions V.2.0.15 and prior, WebAccess Scada Node versions prior to 8.3.1, and WebAccess/NMS 2.0.3 and prior, several stack-based buffer overflow vulnerabilities have been identified, which may allow an attacker to execute arbitrary code.	T1210 - Exploitation of Remote Services T1190 - Exploit Public-Facing Application T1059 - Command and Scripting Interpreter	T1499 - Endpoint Denial of Service T1574 - Hijack Execution Flow

could be revealed based on the impact rather than just the techniques used to exploit the vulnerability.

In conclusion, the model proposed in this study can identify more accurate attack paths based on vulnerability information.

However, the likelihood of identifying impact information or additional attack vectors not mentioned in the vulnerability description is lower than the existing CTID project, highlighting a limitation of this study.

TABLE 9. Comparison of vulnerability detection methods based on identified attack techniques.

CVE-ID	Detection (Proposed model)	Detection (CTID Methodology)
CVE-2016-1409	Detects DoS attacks through network and log monitoring	Detects malware and abnormal behavior via URL inspection and intrusion detection
CVE-2018-8835	Detects anomalous behavior across the network and system	Detects file and environment configuration changes
CVE-2018-14819	Detect abnormal system behavior	Detect file system changes
CVE-2018-7499	Detect changes in network traffic and service availability	Detect changes in files and processes.

VII. DISCUSSION

Based on the attack technique information identified in the proposed model and the CTID methodology, we will assess how each vulnerability can be detected and mitigated.

This assessment aims to determine whether the appropriate attack technique has been identified and whether the detection and mitigation measures are suitable.

A. ANALYZING ATTACK DETECTION METHODS FOR VULNERABILITIES IN OPERATIONAL TECHNOLOGY

Using the attack techniques identified from vulnerabilities in the operational environment in the case study, we applied the detection information provided by the MITRE ATT&CK framework.

This analysis was conducted on the four vulnerabilities in the case study to evaluate how accurately and efficiently the exploits identified based on the vulnerability description detect the vulnerabilities, as shown in Table 9.

CVE-2016-1409 affects the NDP implementation in the IPv6 stack of certain Cisco IOS versions. Exploiting this vulnerability, remote attackers can cause a DoS by sending crafted NDP messages.

The detection scheme for the attack techniques identified in the proposed model focuses on detecting DoS attacks through network monitoring and server logging. Real-time network traffic monitoring can quickly identify fabricated NDP messages. Additionally, server log analysis helps detect suspicious activity related to this vulnerability.

In contrast, the CTID methodology identifies specific attack techniques through various methods. These include URL inspection, network intrusion detection for identifying malicious code, and monitoring endpoints for anomalous behavior.

However, URL inspection is unsuitable for network-level attacks that exploit the NDP, and malware detection methods cannot identify manipulated NDP messages. Additionally, more than monitoring individual endpoints is required to detect NDP attacks that affect the entire network.

In conclusion, the proposed model's detection scheme effectively identifies DoS attacks exploiting the NDP. It achieves this through network traffic analysis and server log monitoring.

CVE-2018-8835 is a double-free vulnerability caused by a memory management issue in the Advantech WebAccess HMI designer when handling '.mp3' files. It allows remote code execution.

The proposed model's detection scheme identifies various attack techniques, including abnormal DLL loading, irregular network traffic, packet anomalies, and unusual process behaviors. By comprehensively monitoring these aspects, it serves as an effective early warning system in complex attack scenarios.

In contrast, The CTID methodology identifies specific attack techniques and applies detection methods focused on monitoring changes in executables, environment variables, and identifying malicious files. These methods effectively identify tampering with specific files or environment settings.

However, they are limited in detecting memory management issues or changes in DLL loading, as noted in the vulnerability description. Moreover, the lack of network traffic analysis hinders detection of remote code execution attempts.

Therefore, the detection scheme identified through the proposed model is more suitable for detecting and responding to double-free vulnerabilities.

CVE-2018-14819 is an out-of-bounds read vulnerability when a program reads data beyond its allocated memory range.

The proposed model's detection techniques offer a broad security scheme. This approach can indirectly detect memory-related vulnerabilities by monitoring anomalous behavior, unusual DLL loading, and suspicious network traffic.

By monitoring these indicators, attack attempts exploiting such vulnerabilities can be detected early.

In contrast, the CTID methodology's detection methods focus primarily on system changes. However, they are somewhat limited in directly addressing memory vulnerabilities, such as out-of-bound reads.

This approach effectively monitors filesystem changes, including binary hashing and registry modifications. However, it doesn't directly detect changes in memory access patterns or anomalous behavior.

The proposed model's detection approach is more relevant for detecting such vulnerabilities. Still, it also has limitations because its broader scope of monitoring does not specifically target memory manipulation.

CVE-2018-7499 is a stack-based buffer overflow vulnerability in multiple versions of Advantech WebAccess that allows an attacker to execute arbitrary code.

Our proposed model's detection approach can identify early signs of buffer overflow attacks. It does this by analyzing network traffic in real-time, detecting abnormal packet sizes and unexpected use of communication protocols.

TABLE 10. Comparison of vulnerability mitigation methods based on identified attack techniques.

CVE-ID	Mitigation (Proposed model)	Mitigation (CTID Methodology)
CVE-2016-1409	Apply Cisco Software Updates, ND Filtering and Limitation, IP Security, Strengthen Network Device Security Settings	Network Segmentation
CVE-2018-8835	Update and Patch, Access Controls and Permissions, Network Security Measures, Regular Security Audits and Code Reviews, User Education and Awareness, Application Isolation and Sandboxing	Update and Patch, Access Controls and Permissions, Network Security Measures, User Education and Awareness, Application Isolation and Sandboxing
CVE-2018-14819	Software Updates, Memory Protection Techniques, Input Validation Enhancements, Intrusion Detection Systems (IDS) and Firewall Configurations, Principle of Least Privilege, Education and Awareness	Software Updates, Memory Protection Techniques, Input Validation Enhancements, Intrusion Detection Systems (IDS) and Firewall Configurations, Principle of Least Privilege
CVE-2018-7499	Patch and Update Management, Network Security Enhancements, Security Applications and Protective Tools, Application and Process Isolation, User Privilege Management, Threat Intelligence Development, Configuration and Access Controls	Patch and Update Management, Network Security Enhancements, Security Applications and Protective Tools, User Privilege Management, Threat Intelligence Development, Configuration and Access Controls

Additionally, by external monitoring, we can respond quickly to service availability before the buffer overflow vulnerability is fully exploited and impacts the service.

In contrast, the CTID methodology's attack detection techniques focus on monitoring changes to files and processes within the system.

While this approach can be useful as a follow-up after an attack, it has limitations in detecting buffer overflow vulnerabilities before they are exploited.

Methods that analyze changes within a system are unsuitable for proactive detection, as they can only identify issues after an attack.

Therefore, the proposed model's detection approach focuses on real-time network-level analysis and monitoring of external services. This method is more effective in early detection and response to buffer overflow vulnerabilities.

The detection methods for the attack techniques identified in the proposed model offer both specific and broad information to identify vulnerabilities in advance.

The CTID methodology's detection methods are not closely related to the specific vulnerability. However, they provide useful follow-up information after the vulnerability has been exploited.

B. ANALYZING ATTACK MITIGATION METHODS FOR VULNERABILITIES IN OPERATIONAL TECHNOLOGY

We identified attack techniques from vulnerabilities in the operational environment of our case study. Using this information, we analyzed defensive measures based on the Mitigation information provided by the MITRE ATT&CK Framework.

This analysis was conducted for the four vulnerabilities in our case study. As shown in Table 10, this allowed us to evaluate the accuracy and efficiency of the identified exploits in mitigating these vulnerabilities.

Mitigations for CVE-2016-1409 include Cisco software updates and NDP filtering. Additional measures involve router settings, network segmentation, IP security, monitoring, and device hardening.

Six of the seven mitigation categories are addressed based on the mitigation information provided by the attack techniques identified in our proposed model. In contrast, the CTID methodology's identified attack techniques provide mitigation information for only one of these categories.

This demonstrates that the proposed model provides sufficient information for identifying mitigations and existing detection methods.

Similarly, for CVE-2018-8835, the proposed model and CTID methodology show differences in mitigation strategies. The main distinctions are in approaches to regular security audits and code reviews. These audits help mitigate issues related to memory management, such as double-free vulnerabilities.

The proposed model identifies mitigations related to attack techniques. Specifically, it recommends regular system scans to automatically identify potentially vulnerable services.

On the other hand, in the case of CTID, there is a limitation in that it does not provide relevant mitigation information.

For the remaining two vulnerabilities, the first difference in the mitigation information provided for CVE-2018-14819 is related to Education and Awareness.

These are measures to provide security training to developers and system administrators to improve their understanding of the vulnerability and ensure they follow security best practices.

While this is not a technically feasible mitigation, it is considered important to prevent exploitation of the vulnerability through social attacks or human error.

The final difference in mitigation information identified in CVE-2018-7499 is Application and Process Isolation. This restricts certain processes from accessing other parts of the system and reduces the impact of malicious code on the system.

When applied to this vulnerability, it can be an important mitigation to ensure that even if the buffer overflow vulnerability is exploited, arbitrary code cannot be executed remotely.

Both the proposed model and CTID method provide most of these mitigations. However, the proposed model identifies more efficient mitigations by suggesting additional measures and offering more definitive solutions.

VIII. CONCLUSION

This study proposes a method for identifying MITRE ATT&CK techniques using vulnerability information that may arise in operational environments. We performed semantic analysis-based preprocessing of the training data to address the word embedding issues identified in previous research. We compared our performance evaluation method with the CTID project, which employs the Delphi method to link vulnerabilities to attack techniques.

The results indicate that our proposed model links specific attack techniques based on the detailed information in the vulnerability descriptions. However, unlike the CTID project, our model has limitations in identifying subsequent impacts that extend beyond the initial vulnerability description.

Future research will address these limitations by expanding the data range to include various cybersecurity resources such as CTI reports, cybersecurity articles, news, and vulnerability reports. This expansion will help achieve a more comprehensive and accurate linkage of attack techniques. Additionally, we plan to incorporate MITRE ATT&CK for ICS technique information in further training to understand the exact impacts in operational environments better.

Finally, to identify and mitigate vulnerabilities as outlined in our study, we will conduct further research to link appropriate detection and mitigation measures, making the model applicable to real-world attack scenarios.

REFERENCES

[1] (2024). *Understanding Vulnerability Detail Pages*. Accessed: Jul. 30, 2024. [Online]. Available: https://nvd.nist.gov/vuln/vulnerability-detail-pages

[2] A. Staves, A. Gouglidis, and D. Hutchison, "An analysis of adversary-centric security testing within information and operational technology environments," *Digit. Threats, Res. Pract.*, vol. 4, no. 1, pp. 1–29, Mar. 2023.

[3] (2023). *CVE To T&TS: Using CVE Attributes for MITRE ATT&CK Mapping*. Accessed: Jul. 30, 2024. [Online]. Available: https://l.vulcan.io/hubfs/Ebooks-and-White-Papers/Vulcan-Cyber-Mapping-CVEs-to-MITRE.pdf

[4] J. Sun, Z. Xing, H. Guo, D. Ye, X. Li, X. Xu, and L. Zhu, "Generating informative CVE description from ExploitDB posts by extractive summarization," 2021, *arXiv:2101.01431*.

[5] (2024). *MITRE ATT&CK? Matrix for Enterprise*. Accessed: Jul. 30, 2024. [Online]. Available: https://attack.mitre.org/matrices/enterprise/

[6] (2024). *Mapping MITRE ATT&CK? To CVEs for Impact*. Accessed: Jul. 30, 2024. [Online]. Available: https://github.com/center-for-threat-informed-defense/attack_to_cve

[7] *NSA/CSS Technical Cyber Threat Framework V2. Cybersecurity Operations The Cybersecurity Products and Sharing Division*, document PP-18-0844, 2018.

[8] (2024). *D3FEND: A Knowledge Graph of Cybersecurity Countermeasures*. Accessed: Jul. 30, 2024. [Online]. Available: https://d3fend.mitre.org/

[9] (2024). *Detections and Analytics*. Accessed: Jul. 30, 2024. [Online]. Available: https://attack.mitre.org/resources/get-started/detections-and-analytics/

[10] (2024). *Enterprise Mitigations*. Accessed: Jul. 30, 2024. [Online]. Available: https://attack.mitre.org/mitigations/enterprise/

[11] P. M. Stone, G. Daniel, and X. Nianwen, *Semantic Role Labeling*. San Rafael, CA, USA: Morgan & Claypool, 2010, ch. 1, sec. 3, pp. 7–20.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, vol. 1. Stanford, CA, USA: Univ. Stanford, 2023, ch. 4, pp. 441–460.

[13] E. Hemberg, J. Kelly, M. Shlapentokh-Rothman, B. Reinstadler, K. Xu, N. Rutar, and U.-M. O'Reilly, "Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting," 2020, *arXiv:2010.00533*.

[14] V. Ilias, "A security control ontology to support automated risk mitigation," M.S. thesis, Univ. Piraeus, Piraeus, Greece, 2023.

[15] L. Sadlek, P. Celeda, and D. Tovarnák, "Current challenges of cyber threat and vulnerability identification using public enumerations," in *Proc. 17th Int. Conf. Availability, Rel. Secur.*, Aug. 2022, pp. 1–8.

[16] Y. Lakhdhar and S. Rekhis, "Machine learning based approach for the automated mapping of discovered vulnerabilities to adversial tactics," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2021, pp. 309–317.

[17] E. Gionanidis, P. Karvelis, G. Georgoulas, K. Stamos, and P. Garg, "Evaluating text augmentation for boosting the automatic mapping of vulnerability information to adversary techniques," in *Proc. IEEE Secure Develop. Conf.*, Oct. 2022, pp. 23–29.

[18] O. Grigorescu, A. Nica, M. Dascalu, and R. Rughinis, "CVE2ATT&CK: BERT-based mapping of CVEs to MITRE ATT&CK techniques," *Algorithms*, vol. 15, no. 9, p. 314, Aug. 2022.

[19] E. Domschot, R. Ramyaa, and M. R. Smith, "Improving automated labeling for ATT&CK tactics in malware threat reports," *Digit. Threats, Res. Pract.*, vol. 5, no. 1, pp. 1–16, Mar. 2024.

[20] K. Aditya, L. Aouad, and Nhien-An Le-Khac, "Linking cve's to mitre attack techniques," in *Proc. 16th Int. Conf. Availability, Rel. Security.*, 2021, pp. 1–12.

[21] I. Branescu, O. Grigorescu, and M. Dascalu, "Automated mapping of common vulnerabilities and exposures to MITRE ATT&CK tactics," *Information*, vol. 15, no. 4, p. 214, Apr. 2024.

[22] (2024). *Mapping ATT&CK To CVEs for Impact*. Accessed: Jul. 30, 2024. [Online]. Available: https://mitre-engenuity.org/cybersecurity/center-for-threat-informed-defense/our-work/mapping-attck-to-cve-for-impact/

[23] E. Aghaei and E. Al-Shaer, "CVE-driven attack technique prediction with semantic information extraction and a domain-specific language model," 2023, *arXiv:2309.02785*.

[24] S. Arian, "Assessing language models for semantic textual similarity in cybersecurity," in *Proc. Int. Conf. Detection Intrusions Malware Vulnerability Assessment*, 2024, pp. 370–380.

[25] A. Basel, "Smet: Semantic mapping of CVE to att&ck and its application to cybersecurity," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, 2023, pp. 243–260.

[26] (2024). *ICS Advisories*. Accessed: Jul. 30, 2024. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories?f

[27] (2024). *NVD Data Feeds*. Accessed: Jul. 30, 2024. [Online]. Available: https://nvd.nist.gov/vuln/data-feeds

[28] (2024). *ATT&CK Data and Tools*. Accessed: Jul. 30, 2024. [Online]. Available: https://attack.mitre.org/resources/attack-data-and-tools/

[29] (2024). *Updates—April 2024*. Accessed: Jul. 30, 2024. [Online]. Available: https://attack.mitre.org/resources/updates/updates-april-2024/

[30] (2024). *SpaCy Industrial-strength Natural Language Processing in Python*. Accessed: Jul. 30, 2024. [Online]. Available: https://spacy.io/

[31] (2024). *Models*. Accessed: Jul. 30, 2024. [Online]. Available: https://docs.allennlp.org/models/main/
[32] (2024). *Word_cloud*. Accessed: Jul. 30, 2024. [Online]. Available: https://amueller.github.io/word_cloud/
[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[34] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," 2019, *arXiv:1908.10084*.
[35] (2024). *Sentence Transformers*. Accessed: Jul. 30, 2024. [Online]. Available: https://huggingface.co/sentence-transformers
[36] (2024). *Multinomial Logistic Regression*. Accessed: Jul. 30, 2024. [Online]. Available: https://www.ibm.com/docs/en/spss-statistics/29.0.0?topic=regression-multinomial-logistic
[37] C. Kwak and A. Clayton-Matthews, "Multinomial logistic regression," *Nursing Res.*, vol. 51, pp. 404–410, Nov. 2002.
[38] (2024). *Att&ck To CVE Mappings CSV*. Accessed: Jul. 30, 2024. [Online]. Available: https://github.com/center-for-threat-informed-defense/attack_to_cve/blob/master/Att
[39] (2024). *Methodology: Mapping ATT&CK To CVEs*. [Online]. Available: https://github.com/center-for-threat-informed-defense/attack_to_cve/blob/master/methodology.md
[40] S. Hollerer, W. Kastner, and T. Sauter, "Towards a threat modeling approach addressing security and safety in OT environments," in *Proc. 17th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Jun. 2021, pp. 37–40.

DO-YEON KIM received the B.S. degree in physics and in IoT artificial intelligence engineering from Chonnam National University, South Korea, in 2022, where she is currently pursuing the master's degree in convergence security. Her research interests include industrial control systems, vulnerability analysis, and artificial intelligence applied cyber threat intelligence problems. During her master's program, she participated in several research projects, including identifying and analyzing vulnerabilities in industrial control systems, the IoT device artifact collection schemes, and vulnerability analysis from an attacker's perspective. Her research interests include predict, detect, and mitigate attack vectors that can be exploited by attackers using explainable artificial intelligence.

SEONG-SU YOON received the B.S. degree in software engineering and the M.S. degree in convergence security from Chonnam National University, Republic of Korea, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree in convergence security. His research interests include machine learning, deep learning, natural language processing, vulnerability analysis, and cyber threat intelligence. During the Ph.D. degree, he has been actively involved in several research projects, including the development of systems for assessing the cross-impact of safety and security regulations on digital assets and investigating the use of machine learning for correlating intrusion activities across diverse log data. He has published papers in various journals and conferences. His work aims to enhance cybersecurity measures by leveraging advanced machine learning techniques to predict and mitigate potential threats.

IECK-CHAE EUOM (Member, IEEE) received the B.S. degree in computer science engineering from Chonnam National University, South Korea, in 2003, the M.S. degree in software engineering from Korea Advanced Institute of Science and Technology, South Korea, in 2015, and the Ph.D. degree in information security from Chonnam National University, in 2019. He is currently a Professor with the Department of Data Science, Chonnam National University. Previously, he was a Researcher with the Institute of Cyber Security, KEPCO KDN. His research interests include industrial control systems, cyber-physical systems, vulnerability assessment, the AI applied Internet of Things, digital forensic, and other issues of system security.