IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024CIP0005

Publicized:2024/10/23

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER Special Section on Cryptography and Information Security

Identifying Relationships between Attack Patterns using Large Language Models*

Takuma TSUCHIDA[†a)], **Rikuho MIYATA**[†], ***Nonmembers***, **Hironori WASHIZAKI**[†], ***Member***, **Kensuke SUMOTO**[†], ***Nonmember***, **Nobukazu YOSHIOKA**[†], ***and*** **Yoshiaki FUKAZAWA**[†], ***Members***

SUMMARY The Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) and Common Attack Pattern Enumeration and Classification (CAPEC) frameworks are essential knowledge bases that catalog traditional attack patterns and their interrelationships (e.g., abstract–concrete relationships). In addition, a knowledge base named Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS) focuses on artificial intelligence (AI)/machine learning (ML)-related attack patterns. Newly discovered attack patterns are incorporated into these knowledge bases manually, potentially leading to missed relationships or delayed information updates. This study introduces a methodology that uses large language models (LLMs) to identify abstract–concrete relationships between attack patterns, aiding in rapid classification and in the rapid development of a defensive strategy. We trained BERT, GPT, and SVM models on ATT&CK, CAPEC, and their combined datasets for relation classification among attack patterns. The evaluation results show that the fine-tuned GPT-3.5 model outperformed the other investigated models, showing potential applicability even to AI/ML-related attack patterns and emphasizing the importance of using training data in the same format as test data. This study also finds that GPT-3.5 effectively focuses on critical descriptive terms, bolstering its performance. The proposed methodology is effective in discerning attack-pattern relationships, demonstrating its potential applicability in the AI security domain.

key words: *LLM, BERT, GPT, attack pattern, relation classification*

1. Introduction

To counteract evolving cyber threats, robust security measures are essential. Threat intelligence, which is critical for understanding attackers' strategies and actions, is supported by frameworks such as Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) [2] and Common Attack Pattern Enumeration and Classification (CAPEC) [3], which compile traditional attack patterns, and Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS) [4], which focuses on artificial intelligence (AI)/machine learning (ML)-related attack patterns. These knowledge bases define vulnerabilities, attack patterns, and their interrelationships. However, because defining relationships between attack patterns is primarily a manual process, the risk of missing relationships and delaying reflecting information exists.

Therefore, this paper proposes a method that uses large language models (LLMs) (i.e., BERT and GPT) and support vector machines (SVMs) to identify abstract–concrete relationships between attack patterns. This paper contributes to enhancing cybersecurity by rapidly identifying new attack patterns and incorporating them into knowledge bases. Rapid response to new attacks, especially with the advancement of AI technology, presents a critical challenge for security professionals. An automated process for identifying the abstract–concrete relationships between new and known attacks clarifies these connections, enabling efficient classification of new threats and formulating appropriate defense strategies. In addition, the ability to identify omitted relationships enhances the completeness and accuracy of the knowledge bases. Consequently, this study establishes a foundation for security experts and researchers to comprehensively understand attack patterns and respond swiftly, contributing to reinforcing security in response to the latest threats. It promotes the timely update of security knowledge bases, supporting the development of an overarching defense framework.

We train ML models on ATT&CK, CAPEC, and the combined datasets, using classification tasks to determine abstract–concrete relationships between attack patterns. Experiments are conducted with various combinations of models and training data for each test dataset, and comparisons and evaluations are made. In addition, through SHapley Additive exPlanations (SHAP) analysis, we elucidate the features that each model prioritizes in the classification task, interpreting the predictive behavior of ML models.

There are four research questions in the present study.

RQ1. Can LLMs be used to identify relationships between attack patterns? The ability of LLMs to identify relationships is critical for determining LLMs' practicality in cybersecurity, directly affecting the efficiency of relationship identification and defense strategy formulation.

RQ2. Which combination of models (SVM, BERT, GPT-3.5) and training data (ATT&CK, CAPEC, combined) shows the highest evaluation metrics? Identifying the most effective model–data combination is essential for optimizing model performance, which can enhance prediction accuracy and practicality in real-world cybersecurity applications.

RQ3. Can our model also be applied to AI/ML-related

[†]The authors are with Waseda University, Tokyo, 169–8555 Japan

*This study expands upon our initial work presented at the Tenth International Conference on Dependable Systems and Their Applications (DSA2023) [1], transitioning from binary to ternary classification. We also introduce additional methodologies for our analysis.

a) E-mail: takuma.t@fuji.waseda.jp

attack patterns? With the advancement of the AI field, verifying the applicability of our model, trained on conventional attack patterns, to address the growing concerns about AI vulnerabilities is essential.

RQ4. What words in the attack descriptions do each model consider important in the relation classification task? Clarifying this aspect can demystify the models' black-box nature, facilitating future model development and application improvements.

The contributions of this paper are summarized as follows:

- A method that uses LLMs to identify relationships between attack patterns is proposed.
- The optimal combination of models and datasets for specifying relationships between attack patterns is identified.
- The applicability of the models in this study to AI/ML-related attack patterns is verified.
- Features prioritized by each model during the classification task are clarified.

This paper is organized as follows. Section 2 discusses threat intelligence frameworks, current challenges, and the technologies used in the present study. Section 3 introduces related work pertinent to the theme of this study, whereas Section 4 details the methodology, including the fine-tuning of each model. Section 5 presents the experimental results and the discussions based on these findings. Section 6 outlines the usage scenarios and benefits of our method for different stakeholders in cybersecurity. Finally, Section 7 concludes the paper and outlines future work.

2. Background

2.1 Threat Intelligence Frameworks and Challenges of the Current Knowledge Bases

Threat intelligence plays a critical role in deciphering attack patterns—a series of actions an attacker performs. ATT&CK and CAPEC serve as exhaustive resources that catalog attack methodologies with IDs, descriptions, and interrelations, aiding security professionals in developing robust defense strategies. ATT&CK defines techniques and their more specific implementations, known as sub-techniques, with an abstract–concrete relationship. Additionally, some ATT&CK techniques and sub-techniques are directly linked to corresponding attack patterns in CAPEC, creating a structured linkage between these datasets at the technique and sub-technique levels. CAPEC describes five types of relationships between attack patterns, including ParentOf and ChildOf, which represent abstract–concrete relationships. Figure 1 shows the "Active Scanning" attack pattern from ATT&CK, an abstract network exploration method. Its specific instance, "Scanning IP Blocks," focuses on IP range data collection, making it a concrete example under Active Scanning. ATLAS is based on the ATT&CK framework, with its techniques and sub-techniques designed to complement those in ATT&CK. Although structured similarly to ATT&CK, ATLAS focuses on a different domain—specifically, targeting AI/ML-related attack patterns. In addition, CAPEC does not have a direct relationship with ATLAS because both its structure and domain focus differ substantially from those of ATLAS. However, this study explores the potential for models trained on ATT&CK and CAPEC to be applied to AI/ML-related attacks using ATLAS data.

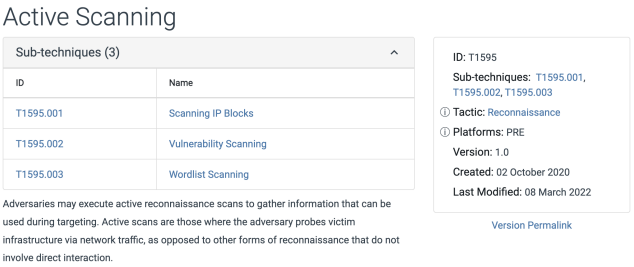


Fig. 1: Example of ATT&CK technique (active scanning)

These knowledge bases draw from incident reports, scholarly work, public threat data, and security community contributions. MITRE vets new patterns from these sources for database inclusion. However, experts manually define relationships between attack patterns. This manual process can lead to omissions in defining relationships or to delays in information reflection. In research on threat evaluation for ML systems using AI incident databases and the literature, Tidjon et al. [5] have identified new threats not yet included in ATLAS. Such newly identified threats should be promptly recognized and integrated into knowledge bases to enhance the comprehensiveness and timeliness of threat intelligence. Our methodology enables the automatic identification of relationships between attack patterns, facilitating swift and precise clarification of connections between new and existing patterns. This automation greatly aids security teams in risk assessment and defense strategy implementation.

2.2 Machine Learning and Natural Language Processing Techniques

Term Frequency–Inverse Document Frequency (TF–IDF) statistics are used to evaluate word importance in a document, assigning higher values to more distinctive words. In the TF–IDF method, documents are transformed into numerical vectors, aiding in document search, classification, and clustering to understand their topics and contents.

SVMs [6] are supervised learning algorithms mainly used for classification tasks. They identify a decision boundary that best separates different classes, maximizing the margin between data points to enhance classification accuracy, even with limited training data, and help prevent overfitting.

The Bidirectional Encoder Representations from Transformers (BERT) language model [7] uses a transformer-based architecture [8] to analyze text contextually by considering words with their entire surrounding context in both

directions, unlike traditional models, which only account for one-directional context. Pre-trained on extensive text, BERT interprets word meanings on the basis of contextual understanding, enhancing performance across various natural language processing tasks.

Generative Pre-trained Transformer (GPT) models leverage a transformer-based architecture and attention mechanism for contextual word relationship analysis; they are pre-trained on vast text data for enhanced language comprehension and generation. Successive versions, including GPT-1 [9], GPT-2 [10], GPT-3 [11], GPT-3.5 [12] and GPT-4, exhibit progressive improvements. In the present study, we use GPT-3.5-turbo-1106, which offers broad context handling and complexity management. It supports various natural language processing applications, excelling in benchmarks with fine-tuning; it also supports few-shot, one-shot, and zero-shot learning approaches, enabling adaptability without task-specific dataset updates.

SHAP [13] is an interpretative framework that uses Shapley values from game theory to elucidate contributions of features in ML model predictions. It evaluates feature impact by considering all possible feature combinations and assessing how the inclusion or exclusion of a feature alters predictions. SHAP enhances model transparency, particularly illuminating feature significance in complex and deep learning models, aiding in understanding prediction processes.

3. Related Works

Our initial study [1] focused on using a binary classification task with BERT, specifically applied to ATT&CK and CAPEC data, to determine whether an abstract–concrete relationship exists between attack patterns. However, in this paper, we extend the scope by transitioning from binary to ternary classification, enabling us to not only detect the existence of a relationship but also identify the directionality, discerning which attack pattern is abstract and which is concrete. Because of this change in task scope, the previous BERT-based results are not included in this paper. In addition, we introduce SVM and GPT models alongside BERT and conduct a comparative analysis to evaluate their effectiveness in this expanded task. Furthermore, we have incorporated additional ATLAS data to investigate the applicability of these models specifically to AI/ML-related attack patterns.

In previous research, Miyata et al. [14] used transformer models and graph structures to identify attack-pattern relationships, focusing on CAPEC's five relationships with the Longformer and BERT models. They identify undefined relationships and assess their validity graphically. The present study shares similarities with Miyata's approach in that BERT is used to identify relationships between attack patterns within CAPEC. Still, it differs by experimenting with various model combinations, including SVM, BERT, and GPT, using training datasets from ATT&CK, CAPEC, or both. In addition, we explore model applicability to AI system attack patterns using specialized datasets such as ATLAS.

Numerous studies leverage security-related databases, including efforts to associate different databases. For instance, the literature includes studies such as automatic mapping from Common Vulnerabilities and Exposures (CVEs) descriptions to ATT&CK [15], research on classifying CVEs into Common Weakness Enumerations (CWEs) [16], and investigations into automatically tracing related CAPEC-IDs from CVE vulnerability information [17]. Other research efforts have been aimed at linking attackers' actions with specific databases, such as by mapping Linux commands to ATT&CK [18].

In the present study, we identify the relationships between attack patterns through a text classification task. This task utilizes various ML and deep learning approaches, including SVM, Random Forest, convolutional neural network (CNN), and transformer-based models [19], [20]. Research has shown that transformers excel in understanding abstract text meanings, whereas TF–IDF is proficient in detail-oriented tasks [21]. Combining BERT and TF–IDF enhances the classification accuracy [22]. GPT-based text classification has also been investigated. Chiu et al. achieved an accuracy as high as 85% with GPT-3 in few-shot learning for sexist and racist text [23]. In addition, supplementing training data with GPT-3–generated examples was found to boost accuracy in data-scarce scenarios [24]. In addition to these approaches, various techniques for leveraging LLMs have been explored in recent cybersecurity studies. For example, chain-of-thought (CoT) prompting and zero-shot/few-shot prompting have been applied to complex reasoning tasks within LLMs, demonstrating strong potential for improving understanding and interpretation of attack patterns [25]–[30]. Retrieval-augmented generation (RAG) has also been used to enhance the contextual relevance of responses generated by LLMs, particularly in scenarios requiring access to external knowledge sources [28]. Additionally, SecureBERT and SecBERT have been fine-tuned to improve their effectiveness in security-related tasks, such as vulnerability detection and classification [27], [29]. The present study compares these approaches with the proposed method, with a detailed analysis provided in the Discussion section.

The present study also addresses attack patterns against vulnerabilities in AI systems. Research on threat assessment for ML systems using AI incident databases and the literature [5] has identified new threats to ML systems that are not yet reflected in ATLAS. Continuously incorporating and analyzing the latest information is critical to address evolving attack methods and new vulnerabilities. McGregor [31] proposed a method to collect and catalog real-world AI failure instances in a database to prevent and mitigate AI incidents. The authors of another study investigated the use of big-data tools to develop models for collecting and analyzing AI system vulnerabilities; their results highlighted the importance of precise data handling, integrating diverse sources, and extracting critical insights for structured vulnerability information [32]. In the present study, we aim to leverage LLMs

to swiftly delineate relationships between new and known attacks, contributing to general and AI-specific security domains.

4. Methodology

The present study aims to identify the relationships between pairs of attack patterns using machine learning and natural language processing. The three target relations are abstract–concrete relationship, concrete–abstract relationship, and no relation. Figure 2 presents the overall flow of the method used in the present study. Initially, information on attack patterns is extracted from ATT&CK and CAPEC to create attack-pattern pairs, which are then split into training and test data. The arrows or lines connecting the attacks represent the relationships between them. The training data are subsequently used to train the SVM model and fine-tune the BERT and GPT-3.5 models. Along with zero-shot learning using GPT-3.5, these four methods are applied to perform ternary classification of the test data. This process identifies whether the relationship between two attack patterns is abstract–concrete, concrete–abstract, or no relation.

4.1 Create dataset

This study focuses on relationships between techniques and sub-techniques in ATT&CK and ATLAS and between the ParentOf and ChildOf relationships in CAPEC. This approach is based on the fact that some ATT&CK techniques reference related attack patterns in CAPEC, creating a structured linkage between these datasets. In ATT&CK and ATLAS, techniques are treated as abstract attack patterns, whereas sub-techniques are considered more concrete implementations. By pairing these techniques and sub-techniques, we establish abstract–concrete relationships. Similarly, in CAPEC, the ParentOf and ChildOf relationships are used to pair attack patterns, where the Parent patterns are treated as abstract and the Child patterns are treated as their more concrete counterparts. Using these pairs from ATT&CK, ATLAS, and CAPEC, we achieve a consistent evaluation of abstract–concrete relationships across different datasets. Attack-pattern descriptions are labeled according to their relationships. For example, in CAPEC, "Command Injection" (ID: 248) serves as a Parent pattern representing a broad category of executing unauthorized system commands, whereas "SQL Injection" (ID: 66) acts as the Child pattern, narrowing this category specifically to database manipulations. This illustrates an abstract–concrete (A–C) relationship. All abstract–concrete pairs defined in each database are used. For fine-tuning, experiments with no-relation (NR) pair quantities (1, 1.5, 2, and 3 times the abstract–concrete pairs) identify the optimal number for best results.

Training data include datasets from ATT&CK and CAPEC and the combined dataset obtained by merging the two. Four dataset types, including ATLAS, serve as test data. Although ATLAS data are not used for training because of the limited number of entries representing abstract–concrete relationships, they are crucial in testing the models' applicability to AI/ML-related attack patterns. This setup enables us to assess whether models trained on traditional attack pattern datasets (ATT&CK and CAPEC) can generalize effectively to the AI/ML domain as represented by ATLAS. Models are trained and tested across these datasets to evaluate which training data combination yields the highest accuracy for the test datasets.

Table 1: Hyperparameters for the SVM Model

Hyperparameter	Value
C	1.0
Kernel	RBF
γ	scale
Cross-validation	5-fold Stratified

4.2 Vectorization with TF–IDF and Training SVM Classifier

TF–IDF is a method used to evaluate the importance of a word within a document, taking into account both the term frequency (tf) and the measure of how rare the word is across all documents (idf) to calculate the significance of a specific word. When a document contains N texts, the TF–IDF value t_{sw} of word w in text s can be calculated as shown in Eq.1, where N_w is the number of texts in which the word w appears:

$$t_{sw} = tf_w \times idf_w = tf_w \times log\left(\frac{N}{N_w}\right) \tag{1}$$

In this method, TF–IDF vectorization is applied to the attack descriptions to represent the critical words numerically. First, TF–IDF vectorization is performed for the descriptions of each of the two attack patterns. The difference between these two vectors is then calculated and defined as a new feature vector that numerically represents the differences in content between the two texts.

We train an SVM algorithm with this feature vector to identify attack-pattern relationships. SVM, a supervised learning algorithm, determines the optimal decision boundary for classification. The hyperparameters selected for the SVM model are detailed in Table 1. We used the default settings provided by the Support Vector Classifier (SVC) implementation in the scikit-learn library. The regularization parameter C was set to 1.0, which provides a balance between maximizing the margin and minimizing classification errors. The kernel type was set to the Radial Basis Function (RBF), a common choice for handling non-linear data. The γ parameter, which controls the influence of individual training examples, was set to "scale," which automatically scales γ on the basis of the number of features. We use stratified five-fold cross-validation during training to maintain label ratios across subsets for an unbiased distribution. Evaluation was based on the average accuracy and F1 score across subsets.

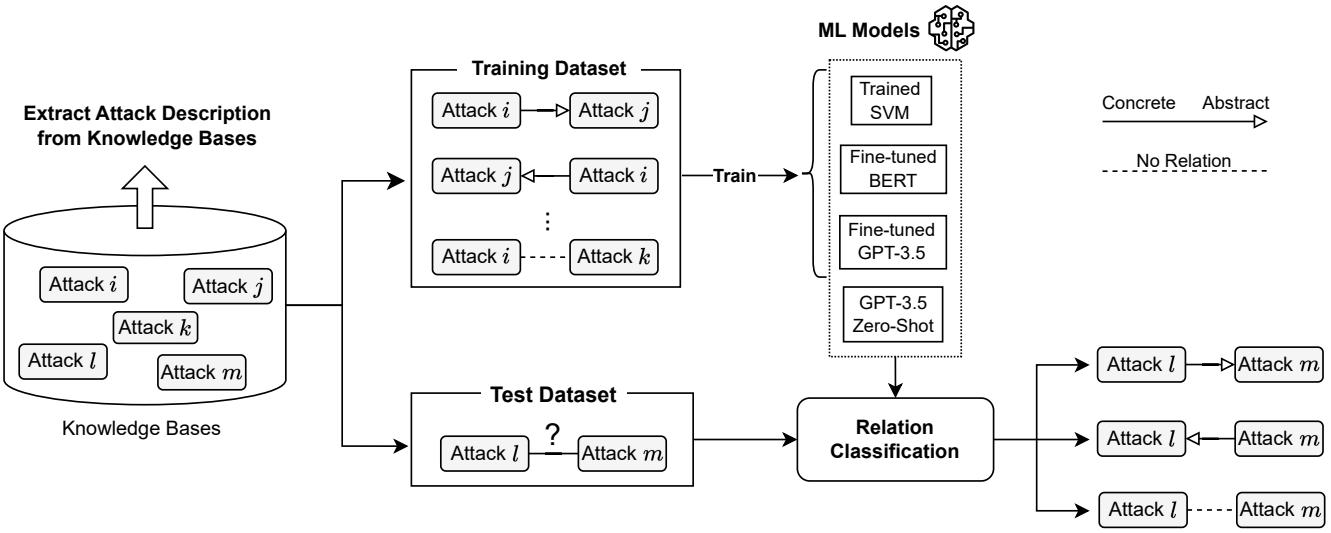


Fig. 2: Overall flow of the method

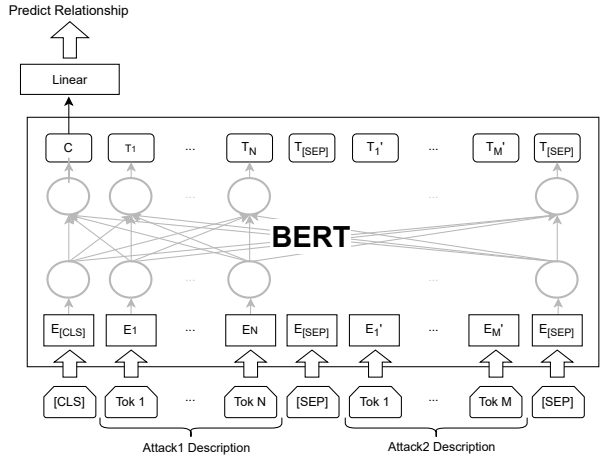


Fig. 3: Conceptual diagram of a model for predicting relationships between attack patterns using BERT

4.3 Fine-tuning BERT

The BERT model uses unique tokens: [CLS] and [SEP]. A [CLS] token is inserted at the beginning when a document or a pair of sentences is input into the model. After the model processes it, the output vector corresponding to the [CLS] token is designed to hold information about the classification of the entire document or pair of sentences. The [SEP] token is inserted to separate two different sentences or documents input into the model, facilitating understanding of the relationship between documents. Figure 3 shows a conceptual diagram of a model for predicting the relationship between pairs of attack patterns using BERT.

To distinguish between two attack descriptions, we insert a [CLS] token at the start, a [SEP] token after the first description, and another [SEP] at the end of the second. This format helps the model identify similarities and differences between attack description pairs, facilitating relationship learning. BERT's encoder processes the tokenized input to generate contextual embeddings. The encoder achieves bidirectional context understanding, converting each token into a contextual vector representation. The final hidden state of the [CLS] token serves as the basis for relationship prediction, with a linear transformation and the softmax function determining class probabilities. The model thus classifies the relationship between attack patterns. We fine-tuned the BERT model using the hyperparameters listed in Table 2. A learning rate of 2e-5 was selected after multiple options were tested. To accommodate the constraints of GPU memory, a batch size of 8 was chosen for training. The model was trained for 12 epochs on the ATT&CK dataset, 10 epochs on the CAPEC dataset, and 8 epochs on the combined dataset, on the basis of the highest observed accuracy for each. The learning rate was dynamically adjusted using a linear scheduler, with the first 10% of the total steps designated as warmup steps. The model optimization was performed using the AdamW optimizer, and Cross-Entropy Loss was used as the criterion for guiding the training process. We used stratified five-fold cross-validation during fine-tuning, evaluating on the basis of average accuracy and F1 scores.

Table 2: Hyperparameters for the BERT Model

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	[1e-5, **2e-5**, 5e-5, 8e-6]
Warmup Steps	10% of total steps
Loss Function	Cross-Entropy Loss
Batch Size	8
Epoch	[8, 10, **12**] (ATT&CK) [8, **10**, 12] (CAPEC) [**8**, 10, 12](Combined)
Cross-validation	5-fold Stratified

Prompt 1: Prompt format for zero-shot learning

```
messages=[
    {"role": "system", "content": "Classify the relationship
          between Description1 and Description2 as abstract to
          concrete, concrete to abstract, or irrelevant. Output 0
          for abstract to concrete, 1 for concrete to abstract,
          and 2 for irrelevant."},
    {"role": "user", "content": "Description1:[Attack 1],
          Description2:[Attack 2]"}
]
```

Prompt 2: Prompt format for fine-tuning

```
messages=[
    {"role": "system", "content": "Classify the relationship
          between Description1 and Description2."},
    {"role": "user", "content": "Description1:[Attack 1],
          Description2:[Attack 2]"},
    {"role": "assistant", "content": "[Correct label]"}
]
```

4.4 Fine-tuning GPT-3.5

In the present study, we use two approaches with GPT-3.5: zero-shot learning and fine-tuning. GPT-3.5's API is publicly available, allowing for various tasks such as text generation, embedding, and fine-tuning. "Chat Completions API" provided for text generation outputs messages generated by the model in response to a list of input messages. In this API, the user sequentially describes a system message to clarify the assistant's operation instructions, a user message for specific requests or texts, and an assistant message for the expected response from the assistant. Ultimately, the assistant generates a reply to the last user message.

This study employs GPT-3.5-turbo-1106 for predicting relationships between attack patterns. We use the Chat Completions API, where system messages guide the classification task, user messages describe attack pairs, and assistant messages indicate expected labels for fine-tuning. For zero-shot learning or when a fine-tuned model is used, the assistant message is skipped, prompting the model to predict labels directly. Prompts 1 and 2 detail the prompt formats for zero-shot learning and fine-tuning, respectively. We fine-tuned the GPT-3.5 model using the hyperparameters listed in Table 3. During the auto-configuration process, the model was trained for 3 epochs. The batch size was automatically set to 1 for the ATT&CK dataset, 2 for the CAPEC dataset, and 4 for the combined dataset, optimizing for the size and characteristics of each dataset. Additionally, a learning rate multiplier of 2 was applied, enabling the learning rate to be adjusted dynamically during training to ensure more effective convergence.

4.5 SHAP Analysis

This study applies SHAP to BERT and GPT-3.5, quantifying each feature's effect on predictions for interpretability. For BERT, classification probabilities from the output layer serve as SHAP values. GPT-3.5's prediction confidence, which is calculated by assigning 1 to the predicted label's index and 0 to others, is treated as SHAP values. Because GPT-3.5's tokenizer is unavailable, we use GPT-2's publicly available tokenizer for SHAP analysis. SHAP's visualization tools are also used for the visual impact analysis of features.

Table 3: Hyperparameters for the GPT-3.5 Model

Hyperparameter	Value
Epoch	3
Batch Size	1 (ATT&CK) 2 (CAPEC) 4 (Combined)
LR multiplier	2

Table 4: Results with different numbers of NR (ATT&CK)

NR	Accuracy	F1 score
411	0.861	0.861
617	0.807	0.862
822	**0.908**	**0.897**
1,233	0.907	0.880

Table 5: Results with different numbers of NR (CAPEC)

NR	Accuracy	F1 score
529	0.778	0.774
794	0.765	0.756
1,058	0.804	**0.786**
1,587	**0.816**	0.769

Table 6: Results with different numbers of NR (combined)

NR ATT&CK:CAPEC	Accuracy	F1 score
411:529	0.838	0.838
822:529	0.845	0.842
822:1,058	**0.874**	**0.862**

5. Evaluation

First, we discuss the selection of the number of data points in the dataset. Second, we describe the results of each model's identifying relationships between attack patterns and the results of the SHAP analysis. Third, we discuss answers to research questions.

5.1 Selection of the number of data

We conducted experiments with multiple numbers of NR attack pairs in the datasets to determine the most effective number for fine-tuning. Tables 4, 5, and 6 respectively show the results of experiments in which the numbers of NR pairs in the ATT&CK, CAPEC, and the combined datasets are compared. For ATT&CK, the highest evaluation metric was achieved when the number of NR pairs was 822. Thus, the ATT&CK dataset was set to have 411 A–C and C–A pairs and 822 NR pairs. For CAPEC, 1,587 NR pairs yielded the highest accuracy and 1,058 NR pairs had the highest F1 score. Given that larger class sizes can bias classification, thereby enhancing accuracy, we prioritized the F1 score, selecting 1,058 NR pairs. Thus, the CAPEC dataset included 529 A–C and C–A pairs and 1,058 NR pairs. For the combined dataset, testing three NR ratios between ATT&CK and CAPEC showed the best results at [822:1,058]. Thus, we set the combined dataset to have 940 A–C and C–A pairs and

1,880 NR pairs. The datasets were split into training and testing sets in an 80:20 ratio. Table 7 provides a detailed overview of the datasets used in our evaluation, including the type of data, number of entries for training and testing, and respective structures, as well as the version used and the date retrieved.

5.2 Experimental results

5.2.1 Results of the classification task

Tables 8, 9, and 10 present the results of classification tasks performed on four types of test data using SVM, BERT, and GPT trained on three types of training data. The results for GPT include zero-shot learning outcomes.

A comparison of the three models reveals that GPT-3.5 consistently achieves the highest evaluation metrics. For three test data other than ATLAS, models trained on data from the same knowledge base generally showed the highest accuracy (except for the combined test data of GPT). For the ATLAS test data, models trained on the ATT&CK dataset exhibited greater accuracy than those trained on CAPEC. However, the highest evaluation metrics for the ATLAS test data were observed in models trained on the combined dataset.

A comparison between GPT's zero-shot learning and the fine-tuned model (Table 10) reveals that fine-tuning substantially improved the evaluation metrics across all of the test data. Specifically for ATLAS data, models fine-tuned on the combined data showed an accuracy increase greater than 0.5 points compared with zero-shot learning.

5.2.2 Results of SHAP analysis

SHAP analysis of BERT and GPT-3.5 highlights the key features influencing predictions, aiding interpretation through visual contribution charts. Red and blue highlights in attack descriptions indicate positive and negative effects on predictions, respectively, with the color intensity indicating the strength of the effect. This visualization clarifies which words are deemed important by the model.

Figure 4 shows the SHAP analysis results for BERT fine-tuned on the CAPEC dataset for classifying the abstract–concrete attack pair of "Active scanning" and "Scanning IP Blocks" from ATT&CK. In this attack description, words such as "scan" and "gather information" represent the main points of the attack technique, whereas "IP" denotes the word indicating the specific means of attack. These words are assumed to be critical in the relationship classification, indicating that the results of the SHAP analysis were interpreted accordingly. In Fig. 4, the effect of each feature is demonstrated when the input attack pair is predicted to have an abstract–concrete relationship, with words indicating the attack's main points and red highlights indicating the specific methods. Thus, the model trained on CAPEC can be interpreted as significantly valuing these words.

Table 11 summarizes the aspects each model emphasizes during the relation classification task revealed by SHAP

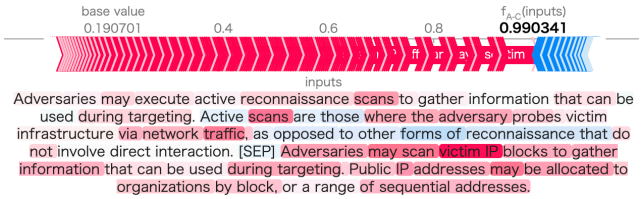


Fig. 4: SHAP analysis of BERT fine-tuned on CAPEC data (target: ATT&CK attack pairs)

analysis. ATT&CK and CAPEC are analyzed as traditional attack patterns, whereas ATLAS is considered for AI/ML attack patterns. "MP" in the table represents the main point of attack methods or objectives, and "CW" represents words symbolic of specific attacks. In addition, "H," "M,", and "L" indicate high, medium, and low importance, respectively.

GPT-3.5's zero-shot learning often prioritized words deemed irrelevant to classification tasks while neglecting critical and specific terms of attacks. By contrast, fine-tuned models showed a tendency. Although most fine-tuned BERT models assigned moderate importance to an attack's main points and concrete words, GPT-3.5 consistently emphasized them highly across traditional and AI/ML-related attack patterns. This distinction between BERT and GPT models was particularly noticeable for AI/ML-related attack patterns, indicating GPT-3.5's effectiveness in identifying keywords for relationship determination even when trained on data from traditional attacks.

5.3 Discussion

We here answer the research questions and discuss the obtained experimental results.

5.3.1 RQ1. Can LLMs be used to identify relationships between attack patterns?

Experiments demonstrated that the fine-tuned GPT-3.5 model exhibited exceptional results, achieving evaluation metrics greater than 90%. This outcome indicates that using LLMs to identify relationships between attack patterns is effective. However, in the case of zero-shot learning using GPT-3.5, the accuracy was less than 60%, confirming the critical importance of fine-tuning with specific training data when leveraging LLMs.

Thus, we answer the question as follows:

> RQ1. Can LLMs be used to identify relationships between attack patterns? **Fine-tuning is necessary, and the fine-tuned GPT-3.5 model can identify relationships between attack patterns with an accuracy greater than 90%.**

5.3.2 RQ2. Which combination of models (SVM, BERT, GPT-3.5) and training data (ATT&CK, CAPEC, combined) shows the highest evaluation metrics?

Among the three models, GPT-3.5 showed the highest evaluation metrics. Moreover, training with the combined dataset,

Table 7: Data Overview of Datasets Used in the Evaluation

Dataset	Type of data	Label	Number of Entries (Training / Test)	Structure	Version	Date Retrieved
ATT&CK	Techniques and Sub-techniques	A-C / C-A	329 / 82	Attack-pattern description pairs with labels	v13.1	2023-10-09
		NR	657 / 165			
CAPEC	ParentOf / ChildOf	A-C / C-A	423 / 106		v3.8	2023-01-16
		NR	846 / 212			
combined	Integrated dataset combining ATT&CK and CAPEC	A-C / C-A	752 / 188		N/A	N/A
		NR	1503 / 377			
ATLAS	Techniques and Sub-techniques	A-C / C-A	- / 36		v4.5.0	2023-11-27
		NR	- / 36			

Table 8: Results for the SVM

Training data		Test data			
		ATT&CK	CAPEC	combined	ATLAS
ATT&CK	Accuracy	**0.927**	0.514	0.697	0.630
	F1 score	**0.927**	0.484	0.692	0.629
CAPEC	Accuracy	0.646	**0.801**	0.731	0.463
	F1 score	0.645	**0.801**	0.729	0.465
combined	Accuracy	0.919	0.786	**0.851**	**0.694**
	F1 score	0.919	0.786	**0.852**	**0.694**

Table 9: Results for BERT

Training data		Test data			
		ATT&CK	CAPEC	combined	ATLAS
ATT&CK	Accuracy	**0.908**	0.584	0.714	0.496
	F1 score	**0.897**	0.489	0.668	0.474
CAPEC	Accuracy	0.643	**0.804**	0.682	0.474
	F1 score	0.577	**0.786**	0.682	0.450
combined	Accuracy	0.902	0.794	**0.874**	**0.526**
	F1 score	0.888	0.785	**0.862**	**0.512**

Table 10: Results for GPT-3.5

Training data		Test data			
		ATT&CK	CAPEC	combined	ATLAS
No training	Accuracy	0.537	0.597	0.543	0.393
(Zero-Shot)	F1 score	0.517	0.553	0.502	0.345
ATT&CK	Accuracy	**0.960**	0.849	0.905	0.824
	F1 score	**0.960**	0.849	0.904	0.826
CAPEC	Accuracy	0.952	**0.969**	**0.954**	0.796
	F1 score	0.951	**0.969**	**0.954**	0.793
combined	Accuracy	0.935	0.943	0.940	**0.898**
	F1 score	0.935	0.943	0.940	**0.899**

which has more data, was not always the most suitable training approach. Models trained on data from the same knowledge base as the test data showed higher evaluation metrics, suggesting that training models with training data in the same format as the test data is more important than merely increasing the volume of training data.

When ATLAS, which lacks training data, was used as the test data, models trained on the combined dataset showed the highest evaluation metrics across all of the investigated models. This result indicates that training datasets composed of multiple data types are adequate when data are scarce or when facing data from a new knowledge base. Training with ATT&CK and CAPEC data might have enabled the models to better understand the relationships between attack patterns in the new type of knowledge base, ATLAS. For the ATLAS test data, models trained on ATT&CK data alone outperformed those trained on CAPEC. This difference in performance is likely due to the closer alignment between ATLAS and ATT&CK, which is a result of ATLAS being modeled after the ATT&CK framework. The similarity in data structure contributed to the improved accuracy of models trained on ATT&CK data for ATLAS.

Table 11: Elements Emphasized by Each Model in Attack Descriptions Based on SHAP Analysis

		Attack pattern			
		Traditional		AI/ML	
Base model	Training data	MP	CW	MP	CW
BERT	ATT&CK	M	M	M	M
	CAPEC	H	H	M	M
	Combined	M	M	M	M
GPT-3.5	Zero-Shot	L	L	L	L
	ATT&CK	H	M	H	M
	CAPEC	H	H	H	H
	Combined	H	H	H	H

Note: MP, main point; CW, concrete words
H, High importance; M, Medium importance;
L, Low importance

To further assess the effectiveness of our fine-tuned GPT-3.5 model, we compared its performance with that of other prominent approaches, including CoT (zero-shot and few-shot), RAG (zero-shot and few-shot), the fine-tuned SecureBERT, and the fine-tuned SecBERT, all of which were evaluated using ATT&CK as the test data. SecureBERT and SecBERT were trained on ATT&CK data, whereas CoT and RAG were performed using GPT-3.5 and GPT-4o models with zero-shot and few-shot prompting techniques.

For the CoT approach, the system message of the zero-shot prompt in Prompt 3 was used. In the RAG approach, relevant procedure examples from the ATT&CK knowledge base were retrieved and incorporated into the user message of the zero-shot CoT prompt, providing additional context. The few-shot prompts for both CoT and RAG were created by adding examples to the respective zero-shot prompts. The hyperparameters for fine-tuning SecureBERT and SecBERT were the same as those detailed in Table 2.

The results, summarized in Table 12, demonstrate that GPT-4o showed a clear improvement in performance compared with GPT-3.5 because of its advanced architecture. However, our fine-tuned GPT-3.5 model consistently outperformed the other investigated methods, including GPT-4o with CoT and RAG, in both accuracy and F1 score. These

Prompt 3: The system message for CoT(zero-shot learning)

```
Classify the relationship between Description1 and Description2
     as abstract to concrete, concrete to abstract, or
     irrelevant.
Follow these steps:
   1. Read and understand the content of Description1 and
        Description2.
   2. Determine if Description1 is abstract and Description2 is
        concrete. If so, this is an "abstract to concrete"
        relationship.
   3. Determine if Description2 is abstract and Description1 is
        concrete. If so, this is a "concrete to abstract"
        relationship.
   4. If Description1 and Description2 do not have a direct
        relationship, this is an "irrelevant" relationship.
   5. Finally, output the classification result. Use the
        following guidelines to determine the output:
      - If Description1 and Description2 have an "abstract to
           concrete" relationship, output "Conclusion": 0".
      - If Description1 and Description2 have a "concrete to
           abstract" relationship, output "Conclusion": 1".
      - If Description1 and Description2 have an "irrelevant"
           relationship, output "Conclusion": 2".

Answer in the following format:
   {{
      "Step 1": "Explanation of the understanding of
           Description1 and Description2.",
      "Step 2": "Determination if Description1 is abstract and
           Description2 is concrete.",
      "Step 3": "Determination if Description2 is abstract and
           Description1 is concrete.",
      "Step 4": "Determination if Description1 and Description2
           do not have a direct relationship.",
      "Conclusion": 0 or 1 or 2
   }}
```

Table 12: Comparison of Accuracy and F1 Score among Different Approaches

Approach	Accuracy	F1 Score
Proposed Method: Fine-tuned GPT-3.5	**0.960**	**0.960**
GPT-3.5		
CoT (Zero-Shot)	0.579	0.570
CoT (Few-Shot)	0.613	0.604
RAG (Zero-Shot)	0.543	0.506
RAG (Few-Shot)	0.577	0.518
GPT-4o		
CoT (Zero-Shot)	0.748	0.733
CoT (Few-Shot)	0.789	0.782
RAG (Zero-Shot)	0.827	0.830
RAG (Few-Shot)	0.819	0.811
Fine-tuned SecureBERT	0.936	0.926
Fine-tuned SecBERT	0.834	0.823

results indicate that the fine-tuning process enabled GPT-3.5 to adapt specifically to the nuances of the ATT&CK data, leading to superior performance even against more advanced models with complex prompting techniques.

In the comparison between CoT and RAG, we observed that RAG did not improve accuracy for GPT-3.5 but did enhance performance for GPT-4o. The additional context provided by RAG often included irrelevant information, leading to confusion and lower performance in GPT-3.5. However, GPT-4o was better equipped to handle this additional information without an adverse effect on its performance.

Finally, SecureBERT outperformed the standard BERT model (Table 9), indicating its better alignment with cybersecurity tasks. However, its performance did not surpass that of our fine-tuned GPT-3.5 model, demonstrating the latter's superior adaptability and performance in identifying attack-pattern relationships.

Thus, we answer the question as follows:

RQ2. Which combination of models (SVM, BERT, GPT-3.5) and training data (ATT&CK, CAPEC, combined) shows the highest evaluation metrics? **GPT-3.5 achieved the highest evaluation metrics, particularly when trained on datasets from the same knowledge base as the test data. Training with the combined dataset, which integrates multiple data types, proved most effective for new or data-scarce knowledge bases.**

5.3.3 RQ3. Can our model also be applied to AI/ML-related attack patterns?

Most of the investigated models showed a tendency for diminished accuracy when ATLAS was used as test data compared to when traditional attack patterns such as ATT&CK and CAPEC were used. Specifically, for BERT and SVM, the accuracy fell below 70%, indicating insufficient effectiveness against attack patterns related to AI systems. However, GPT-3.5 fine-tuned on Combined data achieved nearly 90% evaluation metrics for ATLAS. According to SHAP analysis, the fine-tuned GPT-3.5 model was more effective in capturing words representing the main point of the attack and specific methods of attack for both traditional and AI/ML-related attack patterns compared with the fine-tuned BERT model. This capability strongly contributed to the high accuracy of the GPT-3.5 model. Therefore, the results of this study suggest that the fine-tuned GPT-3.5 model is particularly effective for attack patterns specialized in AI system vulnerabilities.

Thus, we answer the question as follows:

RQ3. Can our model also be applied to attack patterns specific to vulnerabilities in AI systems? **The fine-tuned GPT-3.5 model achieved ~90% evaluation metrics, showing its applicability to AI-specific attack patterns, unlike BERT and SVM, whose accuracy was less than 70%.**

5.3.4 RQ4. What words in the attack descriptions do each model consider important in the relation classification task?

In this study, we conducted a SHAP analysis based on the premise that the main points describing the attack's purpose, means, and impact, as well as the terms relating to the specific means, are important in identifying the abstract–concrete relationship of the attack pair. The fine-tuned BERT moderately prioritized words representing attack main points and specific methods, whereas the fine-tuned GPT-3.5 placed a higher importance on them. This variance likely stems from the architectural, training, and pre-training data differences between BERT and GPT-3.5. The differing levels of importance that each model places on specific words are believed to contribute to the variance in performance in the relationship classification task.

Thus, we answer the question as follows:

RQ4. What words in the attack descriptions do each model consider important in the relation classification task? **Compared with the fine-tuned BERT, the fine-tuned GPT-3.5 places greater importance on words that represent the main points and specific attack methods, leading to higher accuracy in classification tasks.**

6. Usage

6.1 Application scenarios

One application scenario of the method proposed in this paper is rapidly identifying the relationships between newly discovered and known attack patterns, enabling efficient comprehension of the attack methodologies. This method is beneficial for various stakeholders and offers the following advantages:

First, for those registering or documenting attack patterns, this method automatically categorizes new attack methodologies into the appropriate abstract-level categories, facilitating swift reflection in the knowledge base. Consequently, this categorization enhances the organization and documentation of attack information, enabling rapid information sharing.

Second, for security professionals and incident response teams, elucidating the relationships between new and existing attack patterns enables quick access to pertinent information. This information serves as a foundation for practical risk assessment and the formulation of countermeasures.

Third, the proposed approach can identify missing relationships in the knowledge base, benefiting knowledge engineers and security analysts by enhancing the database's completeness and accuracy. Consequently, it contributes to knowledge sharing and the evolution within the cybersecurity community, improving overall security levels.

For example, consider a scenario within the ATLAS framework where a known attack pattern, Exfiltration via ML Inference API (AML.T0024), is already documented. This attack involves adversaries exfiltrating private information through an ML Model Inference API. Suppose a new, more specific attack, Exfiltration via ML Inference API: Extract ML Model (AML.T0024.002), is discovered. Using the proposed method, AML.T0024.002 can be quickly identified as a concrete instance of AML.T0024. Recognizing this relationship enables security teams to apply existing mitigation strategies defined for AML.T0024, such as Restricting the Number of ML Model Queries, to the newly identified attack, AML.T0024.002. This proactive approach helps mitigate risks more effectively.

6.2 Challenges in implementation and operation

Implementing and carrying out the proposed method presents several challenges. Knowledge bases that catalog attack patterns, such as ATT&CK, CAPEC, and ATLAS, are regularly updated, necessitating periodic retraining of models to reflect the most current information. Failure to regularly update the models may diminish their effectiveness over time. Additionally, in real-world environments, the stability and performance of the model are crucial to ensure consistent and accurate identification of attack patterns, especially as the threat landscape evolves.

6.3 Limitations and risks

The proposed method has several limitations and risks. For instance, misclassifications by the model could result in substantial security vulnerabilities. Additionally, the method is highly dependent on the quality and diversity of the training datasets. If the training data is skewed toward specific types of attacks, there is a risk that the model may become overfitted to those patterns, thereby reducing its effectiveness in detecting and responding to other types of attacks.

6.4 Necessity of evaluation using real data

New attack techniques are discovered daily, making evaluations of how well the model adapts to these evolving threats using real-world data critical. Additionally, it is crucial to assess the model's scalability and ability to maintain performance when applied to large datasets and diverse attack patterns. Furthermore, using real data to evaluate the rates of false positives and false negatives helps determine the method's accuracy and identifies areas for improvement to enhance its practical effectiveness.

7. Conclusions and future work

In the threat intelligence frameworks of ATT&CK, CAPEC, and ATLAS, numerous attack patterns and their relationships are defined. However, the manual process of associating newly discovered attack patterns with existing ones can lead to missed connections and delays in updating information. Therefore, this study proposed a method using LLMs (BERT, GPT) and an SVM to identify abstract–concrete relationships between attack patterns. This approach facilitates rapid categorization of newly discovered attack patterns into major abstracted ones, aiding in implementing swift and appropriate defensive measures.

Experimental results demonstrated that the GPT-3.5 model fine-tuned on conventional attack data achieved high accuracy across all of the test datasets, including those related to AI/ML-related attacks. SHAP analysis revealed GPT-3.5's exceptional ability to capture critical words denoting the main points and specific methods of attacks within descriptions. The fine-tuned GPT-3.5 model consistently outperformed the other investigated methods, including GPT-4o with CoT and RAG, in both accuracy and F1 score. We also confirmed that using training data from the same knowledge base as the test data is critical for model construction. However, training models with a combination of datasets

from multiple knowledge bases proved effective when training data were severely lacking or when attack data from new knowledge bases were addressed.

In the present study, we discussed the nature of each model based on visualizations provided by SHAP analysis. Nonetheless, the proposed approach relies on qualitative human interpretation without providing numerical evidence. In future work, we aim to conduct a quantitative analysis by comparing the SHAP values of each feature. Further exploration will also involve applying new GPT models and other LLMs as innovative models for this methodology.

References

[1] T. Tsuchida, R. Miyata, H. Washizaki, N. Yoshioka, and Y. Fukazawa, "Automatic detection of abstract–concrete relationships between attack patterns of att&ck and capec with fine-tuned bert," Tenth International Conference on Dependable Systems and Their Applications (DSA2023), Aug. 2023.

[2] MITRE Corporation, "Adversarial tactics, techniques, and common knowledge."

[3] MITRE Corporation, "Common attack pattern enumeration and classification."

[4] MITRE Corporation, "Adversarial threat landscape for artificial-intelligence system."

[5] L.N. Tidjon and F. Khomh, "Threat assessment in machine learning based systems." arXiv preprint arXiv:2207.00091, 2022.

[6] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol.20, pp.273–297, 1995.

[7] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proceedings of NAACL-HLT 2019, Minneapolis, Minnesota, pp.4171–4186, Association for Computational Linguistics, June 2-7 2019.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol.30, 2017.

[9] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training." Preprint, work in progress, 2018.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," OpenAI blog, vol.1, no.8, p.9, 2019.

[11] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," Minds and Machines, vol.30, pp.681–694, 2020.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," Advances in neural information processing systems, vol.33, pp.1877–1901, 2020.

[13] S.M. Lundberg and S.I. Lee, "A unified approach to interpreting model predictions," Advances in neural information processing systems, vol.30, 2017.

[14] R. Miyata, H. Washizaki, K. Sumoto, N. Yoshioka, Y. Fukazawa, and T. Okubo, "Identifying missing relationships of capec attack patterns by transformer models and graph structure," 2023 IEEE/ACM 1st International Workshop on Software Vulnerability (SVM), pp.14–17, IEEE, 2023.

[15] 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, 2022.

[16] E. Aghaei, W. Shadid, and E. Al-Shaer, "Threatzoom: Hierarchical neural network for cves to cwes classification," Security and Privacy in Communication Networks, ed. N. Park, K. Sun, S. Foresti, K. Butler, and N. Saxena, Cham, pp.23–41, Springer International Publishing, 2020.

[17] K. Kanakogi, H. Washizaki, Y. Fukazawa, S. Ogata, T. Okubo, T. Kato, H. Kanuka, A. Hazeyama, and N. Yoshioka, "Tracing cve vulnerability information to capec attack patterns using natural language processing techniques," Information, vol.12, no.8, 2021.

[18] Y. Andrew, C. Lim, and E. Budiarto, "Mapping linux shell commands to mitre att&ck using nlp-based approach," 2022 International Conference on Electrical Engineering and Informatics (ICELTICs), pp.37–42, IEEE, 2022.

[19] M.N. Asim, M.U. Ghani, M.A. Ibrahim, *et al.*, "Benchmarking performance of machine and deep learning-based methodologies for urdu text document classification," Neural Computing & Applications, vol.33, pp.5437–5469, 2021.

[20] S. Rehman, A. Irtaza, M. Nawaz, and H. Kibriya, "Text document classification using deep learning techniques," 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE), pp.1–6, 2022.

[21] B. Joshi, N. Shah, F. Barbieri, and L. Neves, "The devil is in the details: Evaluating limitations of transformer-based methods for granular tasks," Proceedings of the 28th International Conference on Computational Linguistics, ed. D. Scott, N. Bel, and C. Zong, Barcelona, Spain (Online), pp.3652–3659, International Committee on Computational Linguistics, Dec. 2020.

[22] J.W. Sun, J.Q. Bao, and L.P. Bu, "Text classification algorithm based on tf-idf and bert," 2022 11th International Conference of Information and Communication Technology (ICTech)), pp.1–4, IEEE, 2022.

[23] K.L. Chiu, A. Collins, and R. Alexander, "Detecting hate speech with gpt-3." arXiv preprint arXiv:2103.12407, 2021.

[24] S.V. Balkus and D. Yan, "Improving short text classification with augmented data using gpt-3," Natural Language Engineering, pp.1–30, Aug 2023.

[25] X. Liu, Y. Tan, Z. Xiao, J. Zhuge, and R. Zhou, "Not the end of story: An evaluation of ChatGPT-driven vulnerability description mappings," Findings of the Association for Computational Linguistics: ACL 2023, ed. A. Rogers, J. Boyd-Graber, and N. Okazaki, Toronto, Canada, pp.3724–3731, Association for Computational Linguistics, July 2023.

[26] C. Zhang, L. Wang, D. Fan, J. Zhu, T. Zhou, L. Zeng, and Z. Li, "Vtt-llm: Advancing vulnerability-to-tactic-and-technique mapping through fine-tuning of large language model," Mathematics, vol.12, no.9, 2024.

[27] R. Fayyazi and S.J. Yang, "On the uses of large language models to interpret ambiguous cyberattack descriptions." arXiv preprint arXiv:2306.14062, 2023.

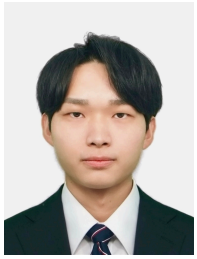
[28] R. Fayyazi, R. Taghdimi, and S.J. Yang, "Advancing ttp analysis: Harnessing the power of large language models with retrieval augmented generation." arXiv preprint arXiv:2401.00280, 2024.

[29] U. Kumarasinghe, A. Lekssays, H.T. Sencar, S. Boughorbel, C. Elvitigala, and P. Nakov, "Semantic ranking for automated adversarial technique annotation in security text," Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, ASIA CCS '24, New York, NY, USA, p.49–62, Association for Computing Machinery, 2024.

[30] J. Zhang, H. Bu, H. Wen, Y. Chen, L. Li, and H. Zhu, "When llms meet cybersecurity: A systematic literature review." arXiv preprint arXiv:2405.03644, 2024.

[31] S. McGregor, "Preventing repeated real world ai failures by cataloging incidents: The ai incident database," Proceedings of the AAAI Conference on Artificial Intelligence, vol.35, no.17, pp.15458–15463, May 2021.

[32] O. Neretin and V. Kharchenko, "Model for describing processes of ai systems vulnerabilities collection and analysis using big data tools," 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT), pp.1–5, 2022.



Takuma Tsuchida is a student at the School of Fundamental Science and Engineering, Department of Communications and Computer Engineering from Waseda University, Tokyo, Japan. He is engaged in research on analyzing attack patterns using Large Language Models.

Rikuho Miyata received Bachelor of Engineering degree in Computer Science and Communications Engineering from Waseda University, Tokyo, Japan in 2023. He is now a master course student of Department of Computer Science and Communications Engineering, Waseda University.

Hironori Washizaki received his Doctoral degree in information and computer science from Waseda University in 2003. Currently, he is a Professor and the Associate Dean of the Research Promotion Division at Waseda University in Tokyo, a Visiting Professor at the National Institute of Informatics, and an Advisor at the University of Human Environments. He also works in industry as an Outside Director of eXmotion. His research interests include reliable and intelligent software engineering, machine learning engineering, and ICT education. He is leading a professional IoT/AI/DX education project called SmartSE. He has served as Chair of IPSJ SIGSE and Convenor of ISO/IEC/JTC1 SC7/WG20. He has been elected IEEE Computer Society 2025 President.

Kensuke Sumoto received the B.E. degree in Computer Science and Engineering from Waseda University, Tokyo, Japan, in 2022. He is now a master course student of Department of Computer Science and Communications Engineering, Waseda University. His research interests include analysis of vulnerability reports with natural language processing.

Nobukazu Yoshioka is a senior researcher / Professor at the Research Institute for Science and Engineering at Waseda University, Japan. Dr. Nobukazu Yoshioka received his B.E degree in Electronic and Information Engineering from Toyama University in 1993. He received his M.E. and Ph.D. degrees in School of Information Science from Japan Advanced Institute of Science and Technology in 1995 and 1998, respectively. From 1998 to 2002, he was with Toshiba Corporation, Japan. From 2002 to 2004 he was a researcher, and from 2004 to 2021, he had been an associate professor of National Institute of Informatics, Japan. Since 2021, he has been a Senior Researcher of Waseda Research Institute for Science and Engineering, Waseda University, Japan. His research interests include Security and Privacy Software Engineering and Software Engineering for Machine Learning-based Systems. He is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE) and Japan Society for Software Science and Technology (JSSST), the Japanese Society for Artificial Intelligence (JSAI) and IEEE CS.

Yoshiaki Fukazawa received the B.E, M.E. and D.E. degrees in electrical engineering from Waseda University, Tokyo, Japan in 1976, 1978 and 1986, respectively. He is now a professor at the Department of Information and Computer Science, Waseda University. His research interests include software engineering, especially the reuse of object-oriented software and agent-based software. He is a member of IPSJ, IEICE, JSSST, ACM and IEEE.