IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 August 2023, accepted 31 August 2023, date of publication 13 September 2023, date of current version 20 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315121

RESEARCH ARTICLE

Exploiting TTP Co-Occurrence via GloVe-Based Embedding With MITRE ATT&CK Framework

CHANHO SHIN, INSUP LEE, (Student Member, IEEE), AND CHANGHEE CHOI
Cyber Technology Center, Agency for Defense Development, Seoul 05661, South Korea

Corresponding author: Changhee Choi (changhee84@add.re.kr)

This work was supported by the Agency for Defense Development, Republic of Korea.

ABSTRACT The digital transformation of various systems has brought great convenience to our daily lives, but it has also increased the level of cyberattacks. As the number of cyberattacks has increased, so has the number of reports analyzing them, MITRE publishes the ATT&CK Matrix which analyzes the tactics and techniques of attacks based on real-world examples. As the flow of attacks has become more understandable through TTP information, researchers have been using it with deep learning models to detect or predict attacks, which makes embedding essential to train the model. In previous studies on embedding TTPs, embedding is limited to simple statistical methods such as one-hot encoding and TF-IDF. Such methods do not consider the order of TTPs and the conceptual similarity between TTPs, therefore do not capture the rich information that TTPs contain. In this paper, we propose embedding TTP with GloVe, a method using a co-occurrence matrix. To properly evaluate the semantic embedding performance of TTP, we also propose a measurement called Tactic Match Rate (TMR). In the experimental results, 8 out of 14 tactics showed a TMR of more than 0.5. Especially the "TA0007 (Discovery)" tactic showed the highest TMR of 0.87. Through correlation analysis, the experimental result shows that the reason for the different embedding performances of the tactic is affected by the frequency of the technique in the same tactic, with at most a 0.96 score. We also experimentally demonstrated that the neutrality of TTP affects learning performance.

INDEX TERMS ATT&CK, cyber threat intelligence, embedding evaluation, GloVe.

I. INTRODUCTION

With the computerization of various systems and the advent of the digital transformation era, cybersecurity is becoming increasingly important to societies. In particular, as Advanced Persistent Threats (APT) and state-sponsored attacks increase, the damage has brought not only economic but also political and diplomatic effects [1]. The increasing number of cyberattacks has led to an increase in the number of reports analyzing them. MITRE has published ATT&CK® [2] to share the tactics and techniques used in cyberattacks. Since then, many researchers have used it to create analytical reports. This has made it possible to abstract massive reports into high-level Tactics, Techniques, and Procedures (TTP) information, making it easier to understand the flow of attacks and share them quickly. Especially APT attacks are often carried out to achieve a specific goal, which makes it necessary to deal with the case at a high level.

From the semantic point of view, each TTP string contains a lot of information. The single-word "T1055 (Process Injection)" technique means that an attacker injected malicious codes into a specific process using various methods for privilege escalation or defense evasion. Therefore, many researchers have been conducting diverse studies [3], [4], [5], [6], [7], [8] such as predicting the next attack [9] using TTP, which contains rich information. Since TTP information is a simple string, the embedding process is required to use it for training. Previous studies have used very simple embeddings such as one-hot vectorization [3], [10]. Although they tried to reflect tactical information in the vectorization process, this approach has the limitation that it does not reflect any statistical information of TTP data. The following study attempted to reflect the statistical information of TTP by using Term Frequency-Inverse Document

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

Frequency (TF-IDF) [4]. TF-IDF has the limitation that it cannot be used in documents where the order is important because it only uses simple statistical information. Also, it does not take into account the conceptual similarity between words. Previous research [11] has shown statistically that there is a strong relationship between TTP. By reflecting this strong relationship between TTPs through co-occurrence distribution, conceptual similarity can be considered in embedding. Therefore, we propose TTP embedding using Global Vectors for Word Representation (GloVe) [12] which is designed based on co-occurrence and conceptual similarity.

In this paper, we have further improved and analyzed the preliminary work [13] that proposes embedding TTP with GloVe considering co-occurrence issues. Usually embedding performance evaluation is done through downstream tasks such as Part of Speech tagging (POS) and Named Entity Recognition (NER). Since the paper aims to see how well the semantics of a TTP is embedded, it is more important to see how close similar techniques within the same tactic are embedded. Thus we propose Tactic Match Rate (TMR), a measurement to evaluate the embedding result. A higher TMR indicates that the embedding vectors are close to each other in the same tactic. In our experiments, the "TA0007 (Discovery)" tactic performed well with a TMR of 0.87, but the "TA0010 (Exfiltration)" tactic showed weak embedding tendencies, with a TMR of 0.19. Through correlation analysis, we experimentally demonstrated that the reason why embedding performance varies depending on the tactic is due to the different types and number of techniques that appear together in each tactic. We have also shown experimentally that there are limitations to TTP embedding through static word embedding due to the presence of neutrality in TTP words. The contributions of this paper are summarized as follows.

- We embed TTPs with GloVe and found that the embedding tendency varied by the tactic. Experimental results showed a correlation value of up to 0.96 between the co-occurrence matrix and embedding performance for each tactic.
- We focus on the semantic meaning of TTPs and propose a measurement, TMR. TMR can evaluate the semantic similarity of TTP information.
- We experimentally show that there is a limitation of static word embedding due to the existence of neutrality in TTP words.

The paper is organized as follows. Section II introduces research related to Cyber Threat Intelligence (CTI) with TTPs and embeddings in cybersecurity. Section III describes the prior knowledge and dataset preparation for the experiments. Section IV describes the TTP embedding with GloVe and the proposed measurement called TMR. Section V describes the overall distribution results, tactic-specific results analysis, and TTP neutrality analysis. Section VI presents conclusions, limitations, and future work.

II. RELATED WORK

Previous CTI research has focused on malware classification and attack group classification. To classify malware and attack groups, they use signature features such as API information [14], [15], [16], [17], [18], [19] or network traffic [20], [21], [22], [23]. However, the problem with signature-based classification is that attackers can mask their intention [1], [24].

To address the problem of signature-based classification, researchers have been trying to determine the intent and goal of an attack rather than simply using signatures. Since MITRE published ATT&CK, researchers have been using TTP information in reports to classify attack groups [9] and goals [3]. Researchers automatically extracted TTPs from collected cyberattack reports and used them as a dataset. The extracted TTPs were used to vectorize the reports, and deep learning networks or rules were used to predict attack groups and goals, respectively. The proposed models were successful in making predictions for each purpose but were limited by the small number of training data, which could lead to bias. Other researchers have attempted to classify malware families based on TTP information [6], detect attacks by correlating API calls with TTP information [7], or identify features of malware by mapping Control-Flow Graph (CFG) information to TTP [8].

Since TTP information is simply a string, embedding is essential for training a model. To use TTP information, Shin et al. [3] encoded each technique with a one-hot vector and viewed each tactic as a word, and then converted the report into a sentence of 14 words (tactics) to use in the experiment. The limitation is that it's based on one-hot encoding, so it doesn't reflect the associations between techniques. Another study used the TF-IDF [4] to weigh the TTP information and perform embedding for classification. TF-IDF is an embedding method that statistically represents the importance of a word based on how many times it appears in a particular document. TF-IDF can exclude one-size-fits-all techniques by using inverse document frequency, which makes it possible to capture the characteristics of words. Lee et al. [4] used TF-IDF to embed weighted TTPs for each group. The problem with TF-IDF is that it does not represent the conceptual relationship between words, which means replacing some TTPs with similar TTPs can significantly change the embedding results. APT or state-sponsored attacks have a strict temporal or logical sequence, which makes the first two methods unsuitable. For example, "T1485 (Data Destruction)" and "T1561 (Disk Wipe)" belong to the same tactic and are closely related, but both methods consider those two techniques as completely different words and fail to capture the relationship.

Also, there have been studies that have used natural language processing to use the raw data itself, not just the TTP information [5], [25], [26]. Andrew et al. [5] used natural language processing to match Linux commands to TTP. Other studies have used natural language processing

TABLE 1. Samples of MITRE ATT&CK TTP descriptions.

Tactic	(sub)Technique	Description
TA0001 Initial Access	T1566.002 Spearphishing	Adversaries may send phishing messages to gain access to victim systems.
TA0002 Execution	T1106 Native API	Adversaries may interact with native OS application programming Interface(API) ...
TA0007 Discovery	T1057 Process Discovery	Adversaries may attempt to get information about running processes on a system.
TA0005 Defense Evasion	T1484 Domain Policy Modification	Adversaries may modify the configuration settings of a domain to evade defenses and/or ...

to match Common Vulnerabilities and Exposures (CVE) information [27], Common Attack Pattern Enumeration and Classification (CAPEC) information [27], and Common Vulnerability Scoring System (CVSS) information [28] to TTPs. However, training with only low-level data can exclude domain knowledge and result in learning unnecessary information.

On the other hand, some researchers have paid attention to the characteristics of TTPs and utilized TTP information for CTI research. Rhaman et al. [11] used attack group and software information from the MITRE ATT&CK website to examine the most common TTP and the distribution of co-occurring TTP. The study statistically showed a strong association between TTP, but the study was limited in that it did not attribute any further contributions.

Although there have been many studies using TTPs, it was difficult to find a study that considers the relationship between techniques within a tactic. Therefore, in this paper, we propose a TTP embedding method with GloVe which can consider the co-occurrences between TTPs.

III. DATASET PREPARATION

This section describes the backgrounds to prepare the dataset, including MITRE ATT&CK used for labeling and the data collection process.

A. MITRE ATT&CK

American defense contractor Lockheed Martin Corporation has brought the traditional military term Kill Chain into cyberspace, coining the term Cyber Kill Chain [29]. It is a defense strategy that effectively detects, blocks, and responds to attacks by dividing an attacker's attack into seven steps to counter sophisticated attacks. However, the cyber kill chain framework has the limitation that they merely list the attacker's actions over time and do not provide a connection to what techniques are used, or what strategies are employed by attack groups.

MITRE has released the ATT&CK [2] to overcome these limitations. ATT&CK summarizes tactics, techniques, and sub-techniques based on real-world examples. Table 1 provides a sample of a TTP as defined in ATT&CK. "Tactic" refers to the tactical goal the attack is trying to achieve. For example, an attacker may want to achieve "TA0001 (Initial Access)" to a victim's resources and network. "Technique" refers to the action the attacker performs to accomplish the tactic. The attacker may perform "T1566 (Phishing)" to achieve "TA0001 (Initial Access)." "Sub-technique" is a more detailed technique. To perform a phishing attack, an attacker may use an attachment file, such as "T1566.001 (Spearphishing Attachment)", or a link, such as "T1566.002 (Spearphishing Link)". MITRE manages ATT&CK matrix and mitigation for enterprise, mobile, and ICS environments. Mitigation provides defenders with technology to stop or prevent ATT&CK techniques and sub-techniques. MITRE also manages the TTP used by popular groups, software, and campaigns.

TABLE 2. Sample documents and corresponding TTP labels for each document used in the experiments.

Report	TTP labels
ESET-LoJax.pdf	TA0002.T1106, TA0003.T1543, TA0005.T1014...
Winnti...pdf	TA0005.T1564, TA0005.T1140, TA0007.T1057...
waterbug...pdf	TA0042.T1584, TA0001.T1566, TA0001.T1566...
ENISA...pdf	TA0001.T1566, TA0001.T1190, TA0001.T1189...

The latest version of MITRE ATT&CK is currently 13.1. Since MITRE releases version updates at least once or twice a year, we used version 10.0, which was used until April 2022, as the base for labeling and classification to maintain labeling consistency. In version 10.0 of the ATT&CK, there are 14 tactics, 188 techniques, and 379 sub-techniques.

B. DATA COLLECTION

For our experiments, we used the "Cyber-criminal Campaign Collections (CCC)" [30] dataset, which is a dataset of reports organized by year. Since there are no TTP tags except for recently published reports, we hired an information security expert to manually label the TTP information for training and evaluation.

The TTP descriptions are detailed as they descend to sub-techniques, resulting in low frequencies of individual sub-techniques. We built the dataset by converting sub-techniques to techniques to avoid low frequencies relative to the number of TTPs. The dataset was divided into cases with and without tactical information based on the experiment. For example, "ESET.pdf" in Table 2 contains "TA0002.T1106,

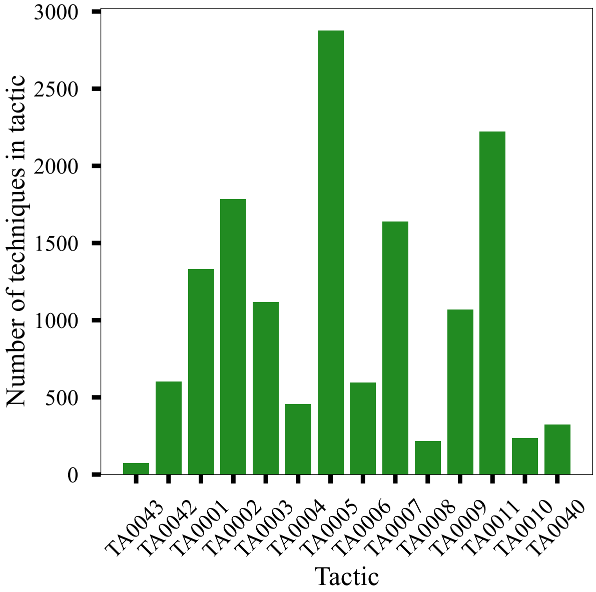


FIGURE 1. The distribution of the dataset over tactics. The number of techniques appearing in the dataset differs from tactic to tactic. The x-axis is the order provided by the MITRE ATT&CK matrix.

TA0003.T1543" and so on. If techniques are used for training without tactic information, they will be replaced by the TTP sequence of "T1106, T1543". The dataset contains 179 techniques with tactic information ($D_{tac.tech}$) and 170 techniques without tactic information (D_{tech}), totaling 1431 data samples. Considering that there are a total of 188 techniques as of version 10.0, the technique coverage is 90%.

Fig. 1 shows the distribution of the dataset over tactics. The total number of occurrences for all techniques is 14535. Note that the number of techniques over tactics is very different. The tactic "TA0005 (Defense Evasion)", "TA0011 (Command and Control)", "TA0002 (Execution)" appear 2877, 2220, and 1785 times, respectively, covering 47.3% of the dataset. On the other hand, the tactic "TA0043 (Reconnaissance)", "TA0008 (Lateral Movement)", "TA0010 (Exfiltration)", and "TA0040 (Impact)" appear 74, 216, 234, and 322 times, respectively, for a total of only 5.8%.

As well as successful attacks, there are many reports of attacks that failed due to early detection by security devices. These attacks did not achieve their final goal, so the techniques used to achieve the final goal are often not described. Therefore, tactics such as "TA0010 (Exfiltration)" and "TA0040 (Impact)", which are in the final stage of the APT, appear less frequently. "TA0043 (Reconnaissance)" tactic includes social engineering techniques that are difficult to detect. And it also includes scanning from outside the network, which is hard to use due to the victim's high-security level. As a result, they appear less frequently in the report. In contrast, techniques that belong to the "TA0011 (Command and Control)" tactic are more common because they are less likely to be detected once penetration into the victim's network is accomplished. We also see many "TA0005 (Defense Evasion)" to avoid detection, and "TA0002 (Execution)" to execute the commands.

TABLE 3. Samples of co-occurrence matrix.

Count	T1203	T1057	T1005	T1204
T1203	0	8.78	7.08	75.00
T1057	8.78	0	28.50	17.81
T1005	7.08	28.50	0	13.08
T1204	75.00	17.81	13.08	0

TABLE 4. Samples of co-occurrence probability.

Probability ratio	k=T1005	k=T1204	k=T1555
$P(k\|T1203)$	0.007	0.075	0.007
$P(k\|T1057)$	0.013	0.008	0.014
$\frac{P(k\|T1203)}{P(k\|T1057)}$	0.538	9.375	0.500

IV. EMBEDDING TTPs WITH GloVe

This section describes the TTP embedding method using GloVe and the proposed measurement, TMR.

A. GloVe-BASED TTP EMBEDDING

Recent analysis shows that not only the number of occurrences is important, but also the type and frequency of co-occurring TTPs. Therefore, TTP information is more suitable for the co-occurrence matrix than the simple frequency. GloVe embedding is an unsupervised learning algorithm used for embedding in the field of natural language processing. It first extracts statistical information from the number of occurrences of words to generate a co-occurrence matrix and co-occurrence probability. Then, a loss function based on the co-occurrence probability is used to learn the ability to make inferences between words.

Table 3 shows a sample of a co-occurrence matrix. All rows and columns consist of the TTPs that appear. The training dataset has a total of 170 TTPs, not including tactics, so the size of the entire matrix is 170 × 170. Position (i, j) in the matrix means the number of times TTP_i and TTP_j co-occurred within a window size N. $N = 10$ means that we want to compute the co-occurrence of the target word with the words within ten to its right. In our experiment, we added up the number of co-occurrence by weighting the distance d, how close two TTPs are to each other within a window size N. Note that the word immediately adjacent is $d = 1$, and if there are three words in between, $d = 4$. The number of co-occurrence, *Count* was calculated by using (1).

$$Count_{n+1} = Count_n + \frac{1}{d} \tag{1}$$

Table 4 shows a sample of a co-occurrence probability based on the co-occurrence matrix. The co-occurrence

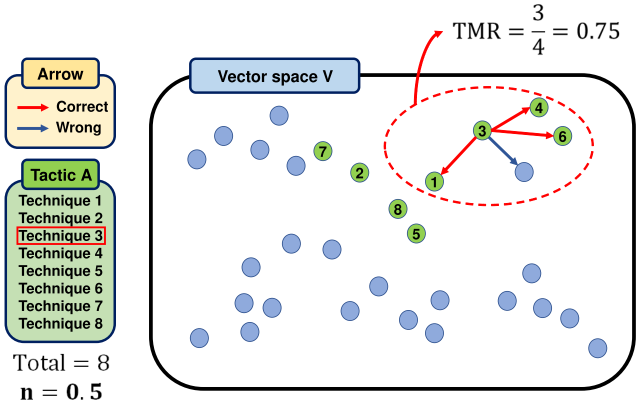


FIGURE 2. A brief description of the TMR. It depicts how to calculate the TMR for technique 3, which belongs to tactic *A*.

probability $P(TTP_i|TTP_j)$ is a conditional probability, meaning the probability of TTP_i appearing when TTP_j appears. From Table 4, we can see that the probability of "T1204 (User Execution)" appearing when "T1203 (Exploitation for Client Execution)" appears is about 9.375 times higher than when "T1057 (Process Discovery)" appears. This is because an "TA0002 (Execution)" tactic is more likely to be used with the same "TA0002 (Execution)" tactic than a "TA0007 (Discovery)" tactic. Putting it all together, the loss function L for the GloVe was calculated by using (2).

$$L = \sum_{m,n=1}^{W} f(X_{mn})(w_m^T \tilde{w}_n + b_m + \tilde{b}_n - \log X_{mn}), \qquad (2)$$

where

$$f(x) = min(1, (\frac{x}{x_{max}})\frac{3}{4}),$$

B. EMBEDDING EVALUATION WITH TACTIC MATCH RATE

In natural language processing, embedding performance evaluation is usually done through downstream tasks such as NER and POS tagging. GloVe has also measured its performance on such downstream tasks. However, to evaluate the unique performance of the embedding method, GloVe also analyzed the performance through word analogy and word similarity. Since our goal is to see how well the semantics of the TTP are reflected in the embedding, it is important to see how close the techniques within the same tactic are embedded. Natural language has grammatical and semantic properties that allow it to answer questions like:

- Is Seoul to Korea, as Tokyo is to *[mask]*?
- eat, ate, *[mask]*?

Unlike natural language, TTP information has no grammatical properties, only semantic properties. Semantic similarity in TTP information refers to the similarity of techniques used to achieve the same goal. Therefore, if TTP information is well learned, the embeddings of techniques that belong to the same tactic should be similar to each other. In this paper, we propose an embedding measurement called Tactic Match Rate (TMR) to measure the semantic similarity of tactics.

The TMR of a $Technique_x$ and the average TMR of a $Tactic_A$ were calculated by using (3) and (4), respectively.

$$TMR(Tech_x) = \frac{1}{K} \sum_{Tech_y}^{TopK(Tech_x)} f(Tech_x, Tech_y),$$

where

$$K = \lfloor len(Tac(Tech_x)) \times n \rfloor,$$
$$f(x, y) = \begin{cases} 1, & \text{if } Tac(Tech_x) \cap Tac(Tech_y) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \qquad (3)$$

$$AvgTMR(Tactic_A) = \frac{1}{|Tactic_A|}(\sum_{Tech_x}^{Tactic_A} TMR(Tech_x)) \qquad (4)$$

First, select the *Top-K* TTPs that are similar to the embedding value of the TTP. If $Tech_x$ and $Tech_y$ belong to the same tactic, count them as 1, and if they belong to different tactics, count them as 0. The summation score averaged by K is defined as the TMR of the corresponding TTP. The same computation is performed for all techniques in $Tactic_A$, and the score averaged over the number of techniques is defined as the "Average TMR". Note that the number of techniques included in each tactic is different, so K should be defined differently for each tactic to be a fair measurement. In this paper, we set $n = 0.5$ and conducted experiments.

Fig. 2 shows a simple visualization of how to calculate the TMR. All techniques are embedded and displayed in the vector space V. The techniques that belong to tactic A are marked as green circles, and the rest are marked with blue circles. Since the total number of tactic A is 8, we search for 4 techniques with similar embedding values to technique 3 at $n = 0.5$. Since three of the four techniques belong to the same tactic as technique 3, the TMR of technique 3 is 0.75.

V. PERFORMANCE EVALUATION

This section describes the experimental results for all techniques, the reason why performance varies by tactic, and the experimental result of the existence of neutrality in TTP words.

A. THE FULL DISTRIBUTION RESULTS

Fig. 3 shows the embedding performance for all techniques on the dataset. We used 170 techniques for training without tactic information. The x-axis shows TMR from 0 to 1 in increments of 0.1. The y-axis shows the number of techniques that correspond to that TMR, i.e., ($x = 0.5, y = 22$) means that there are 22 techniques with TMR between 0.5 and 0.6. We can see that techniques with a TMR of 1 have the highest number of techniques. We can also see that 106 of the 170 techniques have a TMR of 0.5 or higher, meaning that 62.4% of techniques have a high TMR. However, there are also 13 techniques with a TMR of 0, which means that some techniques have a very low embedding performance. This is because tactics have different embedding performances,

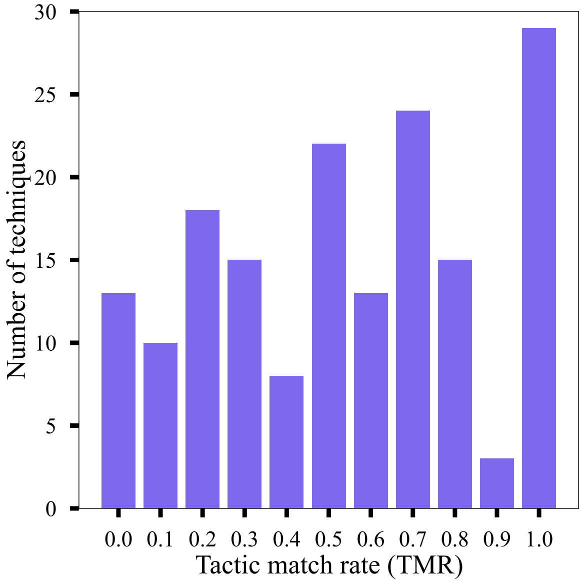


FIGURE 3. The embedding result of the entire technique. The *x*-axis shows the TMR from 0 to 1 in increments of 0.1, and the *y*-axis is the number of techniques that belong to each TMR.

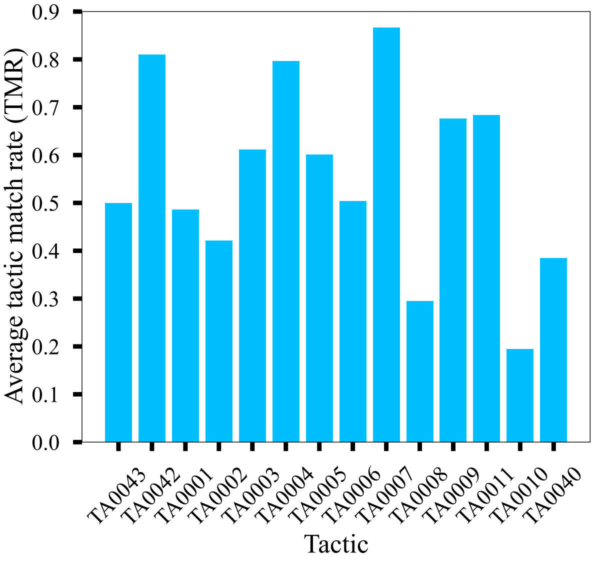


FIGURE 4. Average TMR of each tactic. "TA0007 (Discovery)" has the highest average TMR at 0.87 and "TA0010 (Exfiltration)" has the lowest average TMR at 0.19.

as shown in Fig. 4. Tactics like "TA0042 (Resource Development)", "TA0003 (Persistence)", "TA0004 (Privilege Escalation)", "TA0005 (Defense Evasion)", "TA0006 (Credential Access)", "TA0007 (Discovery)", "TA0009 (Collection)", "TA0011 (Command and Control)" have an average TMR above 0.5, especially the "TA0007 (Discovery)" tactic with a value of 0.87. On the other hand, tactics like "TA0043 (Reconnaissance)", "TA0001 (Initial Access)", "TA0002 (Execution)", "TA0008 (Lateral Movement)", "TA0010 (Exfiltration)", "TA0040 (Impact)" have average TMR values below 0.5, especially the "TA0010 (Exfiltration)" tactic, which has a low value of 0.19. In Section V-B, we'll analyze why each tactic has a different average TMR.

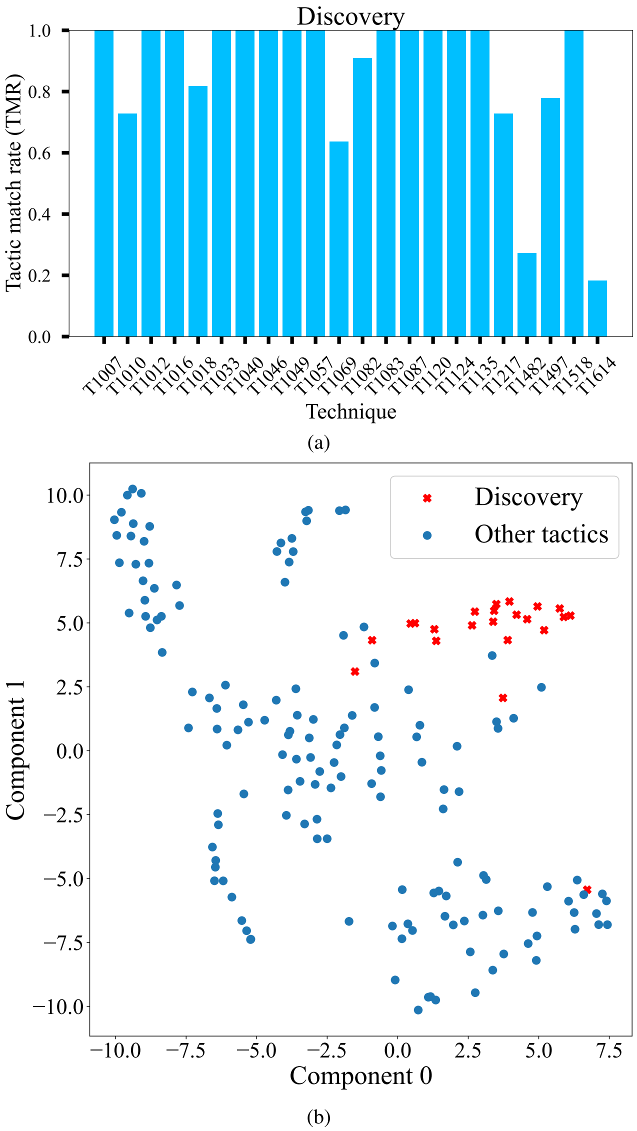
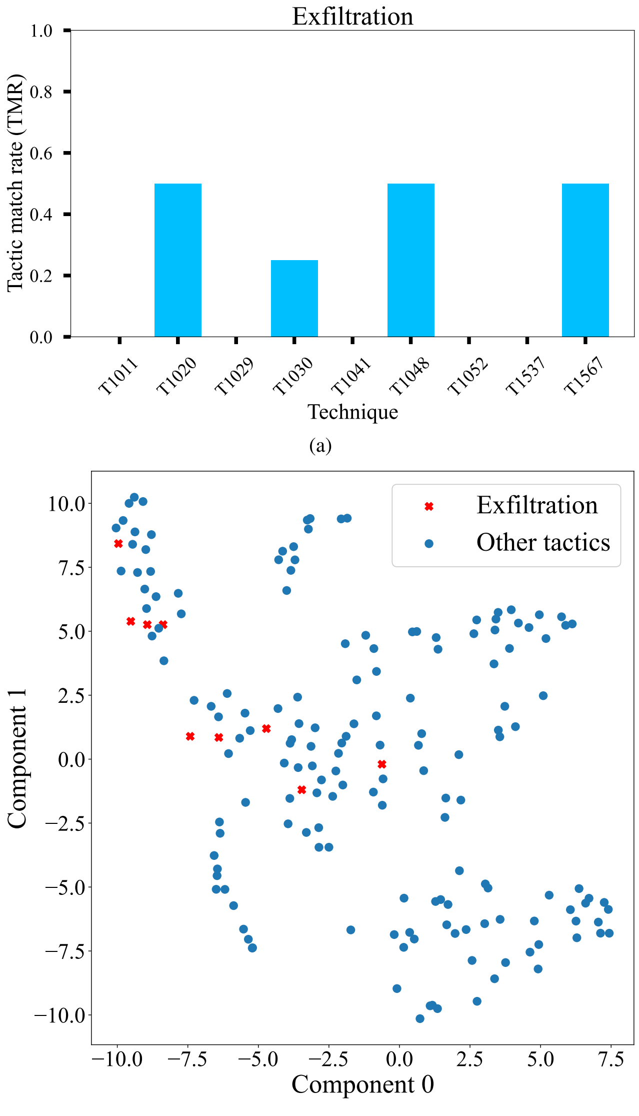


FIGURE 5. Embedding result of "TA0007 (Discovery)" with (a) TMR for each technique and (b) t-SNE visualization.

B. THE RESULTS FOR EACH TACTIC

Fig. 5 shows (a) a bar graph of the TMR of the techniques in the "TA0007 (Discovery)" tactic and (b) a plot using t-distributed Stochastic Neighbor Embedding (t-SNE). As shown in Fig. 5(a), all techniques in the "TA0007 (Discovery)" tactic have a high TMR of more than 0.6 except for "T1614 (System Location Discovery)" and "T1482 (Domain Trust Discovery)". Also, the techniques that belong to the "TA0007 (Discovery)" tactic cluster well with the same tactic except for some points, shown in Fig. 5(b).

Fig. 6 shows (a) the TMR of the techniques in the "TA0010 (Exfiltration)" tactic as a bar graph, and (b) a plot using t-SNE. In Fig. 6(a), five out of nine techniques have a TMR of 0.0. Also, the techniques that belong to the "TA0010 (Exfiltration)" tactic are very scattered, as shown in Fig. 6(b). In other words, the "TA0007 (Discovery)" tactic is well embedded and clustered to belong to the same tactic, whereas the "TA0010 (Exfiltration)" is not.



(a)

(b)

FIGURE 6. Embedding result of "TA0010 (Exfiltration)" with (a) TMR for each technique and (b) t-SNE visualization.

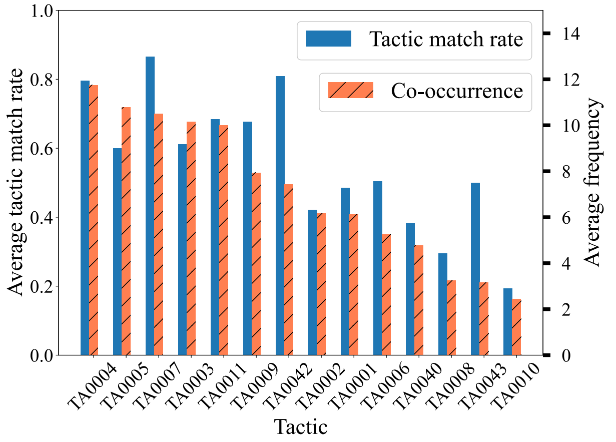


FIGURE 7. Average TMR and average co-occurrence matrix score for each tactic. The correlation between the two graphs is 0.82.

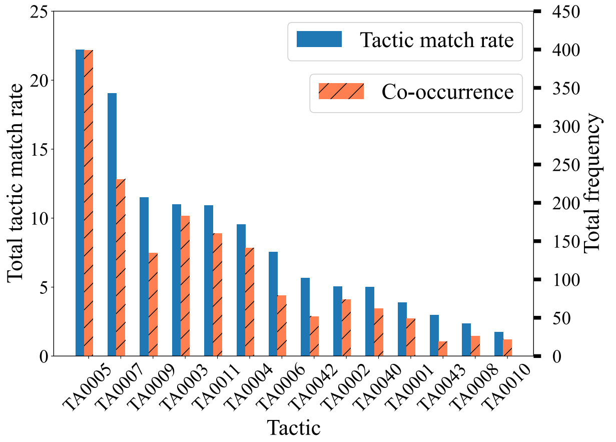


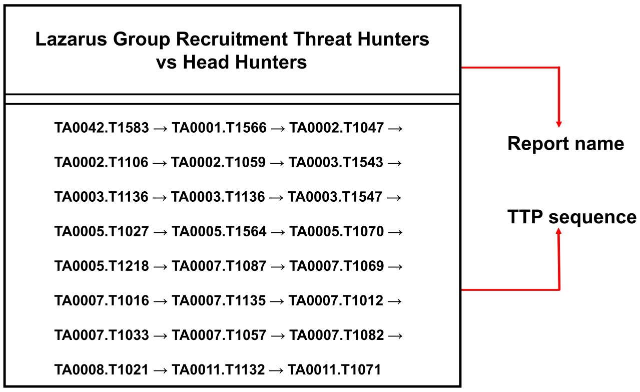
FIGURE 8. Total TMR and total co-occurrence matrix score for each tactic. The correlation between the two graphs is 0.96.

Fig. 7 shows the average TMR (blue) and average co-occurrence value (brown) for each tactic. The correlation between the two bar graphs is 0.82. Fig. 8 shows the total TMR (blue) and total co-occurrence value (brown) by tactic, and the correlation between the two bar graphs is very high at 0.96. It means that the embedding tendency by tactic is highly correlated with the co-occurrence matrix.

GloVe is an embedding method that learns based on a co-occurrence matrix. The larger the value of the matrix, the more likely two words are to co-occur, and the smaller the value, the less likely they are to co-occur. From Fig. 7 and Fig. 8, we can see that different tactics have very different co-occurrence for techniques that belong to the same tactic. It is the reason why the performance of TTP embedding with GloVe varies by tactic.

Different tactics likely have different frequencies of co-occurring techniques. In an APT attack, the attacker will perform "TA0007 (Discovery)" tactics to understand the system and network structure after penetrating the system. Since they have already successfully penetrated the system, they can aggressively conduct discovery without the threat of detection, resulting in multiple co-occurrences of techniques that belong to the "TA0007 (Discovery)" tactic. The same goes for "TA0003 (Persistence)" tactics to maintain continuous access to the victim, and "TA0004 (Privilege Escalation)" tactics to ensure severe attacks.

A sample of a real-world report labeled TTP is shown in Fig. 9. The report is about an APT attack that occurred in 2020 against employees of a pharmaceutical company. Attackers used a ".doc" document masquerading as a job offer. A malicious macro within the document infected the victim's computer. After infection, the attacker performed the following tactics: "TA0007 (Discovery)" to explore resources, "TA0003 (Persistence)" to ensure further access, "TA0002 (Execution)" to carry out attacks, and "TA0009 (Collection)" to gather information. With a total of eight "TA0007 (Discovery)", four "TA0005 (Defense Evasion)", and three "TA0003 (Persistence)" appearances, it is clear that the number of co-occurring techniques varies by tactic. Exceptionally, the "TA0002 (Execution)" tactic also



Lazarus Group Recruitment Threat Hunters vs Head Hunters

TA0042.T1583 → TA0001.T1566 → TA0002.T1047 →
TA0002.T1106 → TA0002.T1059 → TA0003.T1543 →
TA0003.T1136 → TA0003.T1136 → TA0003.T1547 →
TA0005.T1027 → TA0005.T1564 → TA0005.T1070 →
TA0005.T1218 → TA0007.T1087 → TA0007.T1069 →
TA0007.T1016 → TA0007.T1135 → TA0007.T1012 →
TA0007.T1033 → TA0007.T1057 → TA0007.T1082 →
TA0008.T1021 → TA0011.T1132 → TA0011.T1071

FIGURE 9. A sample cybersecurity report of an attack carried out by the Lazarus group. Depending on the tactic, the frequency of co-occurring techniques may differ. In the report, 8 techniques out of 24 TTPs belong to "TA0007 (Discovery)".

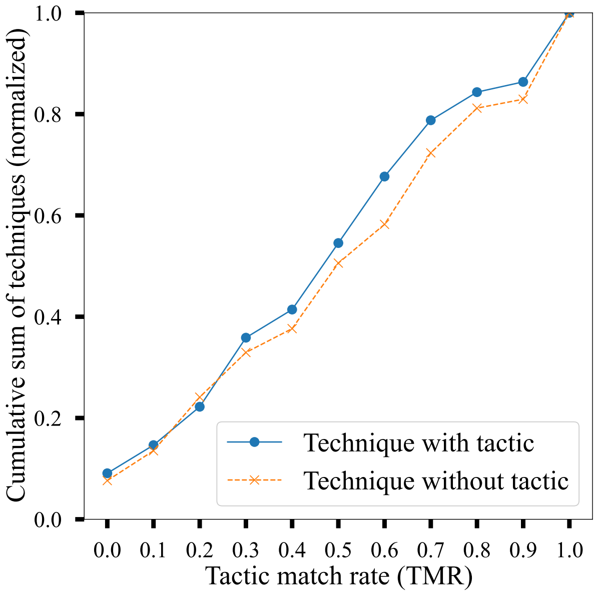


FIGURE 10. Cumulative summation of training with and without tactic information.

appeared three times, but the techniques used, "T1047 (Windows Management Instrumentation)", "T1106 (Native API)", and "T1059 (Command and Scripting Interpreter)" are all techniques with a TMR of 0.5 or higher, and are well embedded within the "TA0002 (Execution)" tactic.

C. THE NEUTRALITY OF TECHNIQUE

Neutrality is the property that a word or sentence can be interpreted in more than one way. Some of the techniques in MITRE ATT&CK can belong to multiple tactics. For example, the "T1053 (Scheduled Task/Job)" technique belongs to "TA0002 (Execution)", "TA0003 (Persistence)" and "TA0004 (Privilege Escalation)". It is a technique that uses a scheduled task for repeated malicious code. "T1072 (Software Deployment Tools)" technique belongs to the "TA0002 (Execution)" and "TA0008 (Lateral Movement)" tactics. It is a technique that uses software to move laterally through the network. In general, a technique that belongs to multiple tactics should have a good TMR because the number of techniques in the same tactic increases, but some techniques do not. This is because a single technique can belong to multiple tactics for a single use, but it can also belong to multiple tactics depending on what it's being used for and what it's trying to accomplish. In other words, some techniques have neutrality.

Since GloVe embedding is a static word embedding, it does not reflect contextual information. Therefore, techniques such as "T1072 (Software Deployment Tools)" whose tactics vary depending on the usage have a problem with relatively low embedding performance. The paper shows this experimentally.

The experiment compares the performance of dataset $D_{tac.tech}$ with tactic information and dataset D_{tech} without tactic information. Fig. 10 shows a cumulative graph of the number of techniques. Both lines are normalized cumulative sum, which means the entire number of techniques is 1.0 at $TMR = 1.0$. Since the number of techniques with high TMR values indicates better embedding, the higher the upper right corner of the cumulative graph means a model with better overall embedding performance. Note that the TMR is slightly higher when learning techniques with tactic. It indicates the technique itself has neutrality, so learning the words with the tactic information has a better performance by stating them together.

VI. CONCLUSION

TTP information is suitable for GloVe embedding based on a co-occurrence matrix because the type and number of co-occurring TTPs are critical, not just the number of appearances. In this paper, we proposed an embedding method using GloVe for TTPs of MITRE ATT&CK which is presented as a new standard in the CTI field. In addition, we have proposed the TMR, an effective embedding evaluation considering the semantic information of TTP. With the proposed measurement, we were able to evaluate which tactics were clustered with the same technique and which were not. Especially, the "TA0007 (Discovery)" tactic showed meaningful results, with a TMR of 0.87. However, the "TA0010 (Exfiltration)" tactic had a lower TMR of 0.19. It is due to the large differences in the frequency of co-occurring techniques between the tactics. Experimental results showed a correlation value of up to 0.96 between the co-occurrence matrix and embedding performance for each tactic. It means that the co-occurrence matrix and the embedding performance of each tactic are highly correlated.

Also, since GloVe is static word embedding, it is difficult to correctly embed techniques where the tactic depends on the context. We showed this experimentally through the difference in TMR performance with and without tactic information. Therefore, large-scale and contextual word embedding methods such as BERT would have an advantage in overcoming these limitations, as does the NLP field.

However, the number of data is critical for large language models. TTP information has difficulties not only in terms of data acquisition but also in terms of transfer learning methods, so we need a way to solve them. In our future work, we plan to investigate transfer learning with large language models and to improve the embedding performance.

REFERENCES

[1] Kaspersky Lab Global Research & Analysis Team. (2018). *Olympic Destroyer is Here to Trick the Indsutry*. [Online]. Available: https://securelist.com/olympicdestroyer-is-here-to-trick-the-industry/84295/

[2] *MITRE ATT&CK*. Accessed: Jun. 5, 2023. [Online]. Available: https://attack.mitre.org

[3] C. Shin and C. Choi, "Cyberattack goal classification based on MITRE ATT&CK: CIA labeling," *J. Korean Soc. Internet Inf.*, vol. 23, no. 6, pp. 15–26, 2022.

[4] I. Lee and C. Choi, "Camp2Vec: Embedding cyber campaign with ATT&CK framework for attack group analysis," *ICT Exp.*, Jun. 2023, doi: 10.1016/j.icte.2023.05.008.

[5] Y. Andrew, C. Lim, and E. Budiarto, "Mapping Linux shell commands to MITRE ATT&CK using NLP-based approach," in *Proc. Int. Conf. Electr. Eng. Informat. (ICELTICs)*, Sep. 2022, pp. 37–42.

[6] V. Chierzi and F. Merces, "Evolution of IoT Linux malware: A MITRE ATT&CK TTP based approach," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, Dec. 2021, pp. 1–11.

[7] Y. S. Takey, S. G. Tatikayala, S. S. Samavedam, P. R. L. Eswari, and M. U. Patil, "Real time early multi stage attack detection," in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, vol. 1, Mar. 2021, pp. 283–290.

[8] J. Fairbanks, A. Orbe, C. Patterson, J. Layne, E. Serra, and M. Scheepers, "Identifying ATT&CK tactics in Android malware control flow graph through graph representation learning and interpretability," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 5602–5608.

[9] C. Choi, C. Shin, and S. Shin, "Cyber attack group classification based on MITRE ATT&CK model," *J. Korean Soc. Internet Inf.*, vol. 23, no. 6, pp. 1–13, 2022.

[10] Y. Shin, K. Kim, J. J. Lee, and K. Lee, "Focusing on the weakest link: A similarity analysis on phishing campaigns based on the ATT&CK matrix," *Secur. Commun. Netw.*, vol. 2022, pp. 1–12, Apr. 2022.

[11] M. Rayhanur Rahman and L. Williams, "Investigating co-occurrences of MITRE ATT&CK techniques," 2022, *arXiv:2211.06495*.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[13] C. Shin, I. Lee, and C. Choi, "Towards GloVe-based TTP embedding with ATT&CK framework," in *Proc. Korea Inst. Military Sci. Technol.*, Daejeon, South Korea, 2023, pp. 1606–1607. [Online]. Available: https://www.kimst.or.kr/

[14] G. G. Sundarkumar, V. Ravi, I. Nwogu, and V. Govindaraju, "Malware detection via API calls, topic models and machine learning," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2015, pp. 1212–1217.

[15] M. Belaoued and S. Mazouzi, "Statistical study of imported APIs by PE type malware," in *Proc. Int. Conf. Adv. Netw. Distrib. Syst. Appl.*, Jun. 2014, pp. 82–86.

[16] M. Wang, C. Zhang, and J. Yu, "Native API based windows anomaly intrusion detection method using SVM," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous, Trustworthy Comput. (SUTC)*, vol. 1, Jun. 2006, p. 6.

[17] J. Yu-Chin Cheng, T.-S. Tsai, and C.-S. Yang, "An information retrieval approach for malware classification based on windows API calls," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 4, Jul. 2013, pp. 1678–1683.

[18] A. Pektaş and T. Acarman, "Malware classification based on API calls and behaviour analysis," *IET Inf. Secur.*, vol. 12, no. 2, pp. 107–117, Mar. 2018.

[19] F. Ahmed, H. Hameed, M. Z. Shafiq, and M. Farooq, "Using spatio-temporal information in API calls with machine learning algorithms for malware detection," in *Proc. 2nd ACM workshop Secur. Artif. Intell.*, Nov. 2009, pp. 55–62.

[20] N. Villeneuve and J. Bennett, "Detecting APT activity with network traffic analysis," Trend Micro, Tokyo, Japan, Tech. Rep., 2012.

[21] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[22] S. Nari and A. A. Ghorbani, "Automated malware classification based on network behavior," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2013, pp. 642–647.

[23] M. Abuthawabeh and K. Mahmoud, "Enhanced Android malware detection and family classification, using conversation-level network traffic features," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4A, pp. 607–614, Jul. 2020.

[24] Kaspersky Lab Global Research & Analysis Team. (2018). *Lazarus Under the Hood*. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07180244/Lazarus_Under_The_Hood_PDF_final.pdf

[25] M. Boffa, G. Milan, L. Vassio, I. Drago, M. Mellia, and Z. Ben Houidi, "Towards NLP-based processing of honeypot logs," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, Jun. 2022, pp. 314–321.

[26] Z. Hussain, J. K. Nurminen, T. Mikkonen, and M. Kowiel, "Command similarity measurement using NLP," in *Proc. 10th Symp. Languages, Appl. Technol.*, 2021, pp. 13:1–13:14.

[27] B. Ampel, S. Samtani, S. Ullman, and H. Chen, "Linking common vulnerabilities and exposures to the MITRE ATT&CK framework: A self-distillation approach," 2021, *arXiv:2108.01696*.

[28] M. R. Shahid and H. Debar, "CVSS-BERT: Explainable natural language processing to determine the severity of a computer security vulnerability from its description," in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2021, pp. 1600–1607.

[29] Lockheed Martin. *Cyber Kill Chain*. Accessed: Jun. 12, 2023. [Online]. Available: https://lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[30] CyberMonitor. *CyberCriminal Campaign Collections*. Accessed: Mar. 20, 2023. [Online]. Available: https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections

CHANHO SHIN received the B.Eng. degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing the M.S. degree in cybersecurity. He is currently a Cybersecurity Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include AI-based cybersecurity, natural language processing, and learning with noisy labels.

INSUP LEE (Student Member, IEEE) received the B.Eng. degree in cyber defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing the Ph.D. degree in cybersecurity. He is currently a Cybersecurity Researcher with the Cyber Technology Center, Agency for Defense Development, Seoul. His research interests include AI-based cybersecurity, intelligent networks, and generative models.

CHANGHEE CHOI received the B.S. degree in computer science from Yonsei University, Seoul, South Korea, in 2008, and the M.S. and Ph.D. degrees in computer science from KAIST, Daejeon, South Korea, in 2010 and 2013, respectively. In 2013, he joined the Agency for Defense Development (ADD), South Korea. His current research interests include AI-based cybersecurity, generative model, digital image forensics, machine learning, and image processing.