MUCAMP: Generating Cyber Campaign Variants via TTP Synonym Replacement for Group Attribution

Insup Lee, *Student Member, IEEE*, and Changhee Choi

***Abstract*—As cyberattack operators have progressed to encompass group and nation-state levels, the nature of attacks has evolved into more sophisticated forms such as cyber campaigns. In response to these large-scale campaigns, tactical cyber threat intelligence (CTI) which focuses on tactics, techniques, and procedures (TTPs) has gained significant attention. However, the data-driven aspects of tactical CTI confront two primary challenges: (i) the extreme scarcity of campaign data and (ii) the difficulty of effectively integrating security domain knowledge. To this end, this paper presents MUCAMP, a novel campaign generation method that operates in the context of limited campaign data while also considering the unique characteristics of large-scale attacks. The proposed method assumes that campaigns are TTP sequences, and based on this assumption, it generates valid campaign variants by replacing target TTP words with *TTP synonyms*, and preserves the strategic goals of the seed campaigns. MUCAMP offers a scalable and interpretable augmentation strategy, enhancing CTI effectiveness under data scarcity and facilitating rapid adaptation to evolving threat landscapes. We also prepared a dataset consisting of 858 real-world campaigns labeled by security experts, including 14 tactics and 206 techniques, enabling reliable performance evaluation. Experimental results demonstrate that each component of MUCAMP contributes to embedding-based group attribution by improving the separability of the correct group from alternative candidates, while effectively reflecting domain knowledge.**

***Index Terms*—Cyber threat intelligence, data augmentation, MITRE ATT&CK, natural language processing.**

I. INTRODUCTION

ALTHOUGH advances in communication and technology offer many positive contributions to daily life, they also increase the number of potential attack vectors and exacerbate the damage caused by cyber threats [2]. These advances have led to a paradigm shift from traditional cyberattacks to sophisticated forms of advanced persistent threats (APTs) and *cyber campaigns*. These campaigns cause significant negative impacts, such as financial losses and social disruption, to achieve the threat actors' aggressive objectives in the cyber domain. For example, a notorious cybercrime group called Lazarus is believed to be responsible for several campaigns, such as the Sony Pictures Hack and the WannaCry ransomware [3]. The Sony Pictures Hack leaked sensitive data (e.g., employees' personal information and emails) and unreleased copies of Sony movie, causing financial damage. Meanwhile, the WannaCry ransomware affected hundreds of thousands of computers in 150 countries, encrypting data and demanding ransom in Bitcoin.

To proactively defend against and mitigate cyber campaigns, cyber threat intelligence (CTI) has emerged as one of the most effective weapons for cyber defenders [4]. CTI refers to the continuous gathering of knowledge from various intelligence sources to gain a deeper understanding of an attacker's intent and context. Several examples of intelligence sources include low-level raw data such as kernel logs [5]–[10] and network traffic [11]–[20]. However, to further consider *large-scale* attacks such as state-sponsored campaigns, we need to approach them at a higher, more abstract level and avoid becoming entrenched in low-level data. From a cyber kill chain perspective, threat actors in large-scale attacks tend to have clear objectives, although the specific steps vary. To address these issues, tactical threat intelligence focuses on high-level indicators of compromise (IOCs), including tactics, techniques, and procedures (TTPs). Specifically, MITRE has presented the ATT&CK® (adversarial tactics, techniques, and common knowledge) framework [21], a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK affects recent tactical CTI research [22]–[30] and influences diverse issues, including real-time detection [22], phishing detection [26], attack graph construction [29], and group attribution [30]. This paper focuses on group attribution, which categorizes the attack groups most likely to have operated a particular cyber campaign.

The most significant factor in ensuring CTI performance is preparing a sufficient amount of reliable campaign data, as without data sufficiency, a CTI model can experience overfitting and create bias in group attribution results. Note that large-scale campaigns have fewer frequencies since attackers need more time to prepare sophisticated operations. Due to the difficulty of collecting campaign samples, data augmentation can be an effective solution. While there have been several studies on data augmentation for traditional security problems [31]–[36], only one study [37] has investigated data augmentation approaches for cyber campaigns consisting of TTPs. Although this study [37] has shown promise in terms of campaign augmentation, several issues still need to be addressed.

We review the campaign augmentation problem from three perspectives: (i) campaign data sparsity, (ii) domain knowl-

Manuscript received 29 September 2024; revised 6 May 2025. The preliminary version of this paper has been presented in the Proceedings of the Korea Institute of Military Science and Technology (KIMST 2023) [1]. This work was supported by Agency for Defense Development, Republic of Korea *(Corresponding author: Changhee Choi.)*

Insup Lee is with the Ministry of National Defense, Seoul 04383, Republic of Korea (e-mail: insuplee94@gmail.com).

Changhee Choi is with the Department of Cyber Defense, Sejong University, Seoul 05006, Republic of Korea (e-mail: choich@sejong.ac.kr).

Insup Lee and Changhee Choi were formerly with the Agency for Defense Development, Republic of Korea.

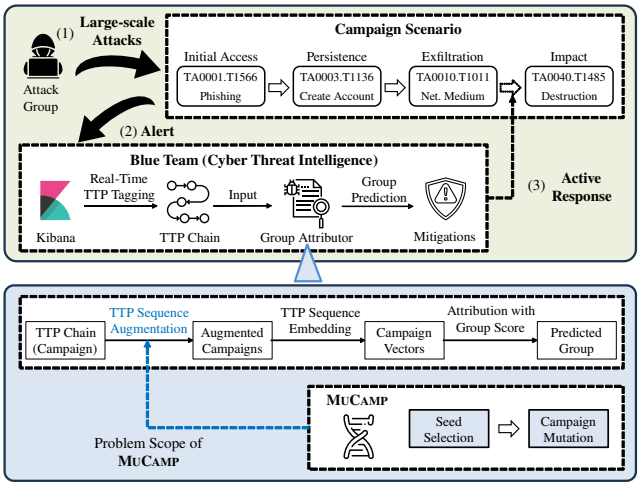


Fig. 1. System overview. An attack group launches a large-scale attack in a scenario consisting of multiple TTPs, alerting the blue team. The blue team then aims to classify the attack group by using TTP chain embeddings to identify effective mitigations that counter the campaign for active response. TTP sequence data augmentation is an effective approach to improve group attribution, motivating MuCamp to generate campaign variants via mutation.

edge, and (iii) generation quality. First, cyber campaigns require a longer preparation time and occur less frequency than traditional attacks do, making it challenging to prepare relevant datasets. The nature of data insufficiency in cyber campaigns hinders the training of deep generative models in terms of augmenting campaign data. Second, generating campaigns without considering their characteristics limits the impact of research. Large-scale attacks such as state-sponsored attack scenarios have a drastic impact, and obtaining the related domain knowledge is critical. Lastly, validating the generated campaigns requires quantitative analysis and domain knowledge-related aspects. To this end, we have derived the following three challenges from previous work.

Challenge 1) How can we design a few-shot generative model with limited campaign data? To address the extreme sparsity of campaign training data, we need to design a model that enables generation without overfitting in a few-shot scenario. Given limited source campaign data, augmentation via *mutation* (i.e., changing several parts of the seed data) can be a practical choice.

Challenge 2) How can we consider domain knowledge in campaign generation? Traditional studies such as [37] consider only the machine learning aspect, with the exception of the security domain. We could reflect on the nature of the cyber kill chain and large-scale attacks (e.g., the objectives and length of campaigns) to design the generative model.

Challenge 3) How can we guarantee the quality of the generated campaign? To ensure validity, we could investigate the group attribution improvement after campaign augmentation with varying parameters under security considerations.

In this context, we present MuCamp, a mutation method for cyber campaigns represented as TTP sequences that facilitates few-shot generation while considering the characteristics of large-scale campaigns. Fig. 1 illustrates a system overview of MuCamp in a large-scale attack scenario consisting of four tactics (Initial Access, Persistence, Exfiltration, and Impact). Assuming that a blue team identifies the campaign, we aim to improve the attack group attribution to derive effective mitigations that correspond to the attack group for active response. Note that, for large-scale attacks, we focus on TTPs to approach this problem from a high level, motivating MuCamp to address the insufficiency of TTP sequences. The main contributions of the paper are summarized as follows.

- We propose a few-shot campaign generator, MuCamp, which is inspired by a lightweight text augmentation method in the natural language processing domain. TTP sequence augmentation via MuCamp successfully improves group attribution.
- We considered multiple aspects of security when designing MuCamp, including (i) the consistency of the attack goals and (ii) the impact of the TTP sequence length, thus offering advantages relevant to real-world CTI and cybersecurity scenarios.
- We constructed a reliable campaign dataset in collaboration with security experts to address a realistic campaign scenario. The experts conducted TTP labeling for the given security reports, and the TTPs consist of 14 tactics and 206 techniques.
- We conducted extensive experiments to assess the contribution of each MuCamp component in terms of enhancing group attribution, specifically focusing on scenarios related to the Lazarus group.

The remainder of this paper is organized as follows. In Section II, we discuss the relevant studies. Section III describes the data preparation process, and Section IV provides details about the system and security models. We present the preliminaries and details of MuCamp in Section V and Section VI, respectively. In Section VII, we discuss the experimental results. Section VIII concludes the paper.

II. Related Work

In this section, we review the relevant studies on cyber threat intelligence (CTI) and the previous efforts to overcome data insufficiency.

A. Cyber Threat Intelligence

CTI has gained significant attention as a crucial defense mechanism against sophisticated attacks such as advanced persistent threats (APTs) and cyber campaigns. CTI employs a broad range of indicators of compromise (IOCs) [5]–[16] and increasingly explores the potential of machine learning and deep learning for more advanced solutions [17]–[20], [38]–[41]. In particular, we investigate tactical CTI based on TTPs [22]–[30], explicitly focusing on large-scale attack scenarios.

CTI data sources. Depending on its purpose, CTI collects information from various intelligence sources, such as at the kernel [5]–[10] and network [11]–[20] levels. Zeng *et al.* [6] employed a log-based knowledge graph coupled with semantic embedding to bridge the semantic gap between low-level and high-level events, namely audit events and systemic behaviors. This method facilitates the automated clustering

of semantically similar behaviors, thereby obviating the necessity for domain-specific expertise. Furthermore, several studies [8]–[10] have focused on provenance tracking systems, considering real-time detection [8], soundness in Linux namespaces [9], and provenance graph modeling with graph neural networks [10]. Zhao *et al*. [11] extracted 14 malicious DNS features to detect APT malware infection. Meanwhile, Bi *et al*. [14] focused their attention on APT activities within the realm of the Industrial Internet of Things (IIoT), and the authors employed a Stackelberg game model to analyze the strategic interactions between the adversary and the defender in the context of APTs in IIoT environments.

Machine learning-based CTI. Recent studies have incorporated machine learning techniques to harness information from diverse intelligence sources and leverage the data-driven aspects of CTI. Notable approaches include random forest [17], [18], [20], contrastive learning [38], bidirectional encoder representations from transformers (BERT) [19], [39], domain adaptation [40], and reinforcement learning [41]. In the domain of intrusion detection system (IDS), Do *et al*. [17], [18] conducted research involving the extraction of intelligence from BRO IDS log files, employing a random forest algorithm to detect APT attacks and command and control (C&C) servers. Kuehn *et al*. [39] proposed ThreatCrawl, which not only contributes to automatic website categorization but also presents a crawling path for extracting IOCs from documents via language models, enhancing the effectiveness of CTI retrieval.

Tactical CTI. In contrast to conventional cyber threats, modeling large-scale attacks necessitates an elevated analytical approach that transcends kernel and network level considerations. This demand motivates tactical CTI studies with MITRE ATT&CK TTPs, e.g., attack automation for red teams [23], phishing detection [26], ransomware detection [27], and group attribution [30]. Song *et al*. [27] suggested analyzing ransomware similarity on the basis of the ATT&CK matrix; this approach involves the application of the term frequency-inverse document frequency (TF-IDF) when constructing ransomware representation vectors, calculating the cosine similarity. Kim *et al*. [25] explored the potential for predicting cyber attacks by examining sequences of TTPs in conjunction with Bayesian networks. Zhang *et al*. [29] introduced a large language model-based framework that constructs TTP-tagged attack graphs from unstructured CTI reports, offering a multi-layered schema for behavior modeling. On the other hand, Lee *et al*. [30] addressed the complexity of group attribution and demonstrated the possibility of representing attack group patterns with simple embeddings and group scores. While group attribution is challenging in large-scale campaigns, it is a critical issue given the adverse effects associated with these attacks, and it motivates us to address the attribution problem.

B. Addressing Data Insufficiency

Preparing sufficient high-quality data is essential for achieving satisfactory performance in data-driven security. There have been several studies on data augmentation for traditional security problems [31]–[36], such as API system calls [31], packets [32], password guessing [34], and CVEs [36]. Shin *et al*. [31] utilized sequence generative models such as sequence to sequence (Seq2Seq) [42] and sequence generative adversarial networks (SeqGAN) [43] to augment anomalies in Linux system call datasets. The generated sequence of abnormal system calls improves the performance of host intrusion detection systems. Wang *et al*. [32] proposed PacketCGAN, a novel packet generative model based on conditional generative adversarial networks (CGAN). The authors preprocessed the input data in packet byte matrix (PBM) format, followed by the application of PacketCGAN, enhancing traffic classification with sufficient PBM. However, previous studies on cyber data augmentation are unsuitable for large-scale scenarios, and more consideration of higher-level aspects such as TTP sequences is needed. One study [37] investigated data augmentation for cyber campaigns consisting of TTPs but considered only machine learning aspects when designing methods and evaluations, i.e., domain knowledge about large-scale attack scenarios is needed.

We reviewed previous studies on data augmentation in terms of security, and this process revealed that the data insufficiency of tactical CTI needs to be addressed. Since no work has deep dived into the campaign augmentation problem, we have designed a campaign generation method called MuCamp, whose architecture and mechanisms will be detailed in Section V.

III. Dataset Preparation

This section describes the MITRE ATT&CK and data labeling process as preliminaries for preparing a reliable campaign dataset.

A. MITRE ATT&CK

The MITRE ATT&CK framework [21] provides a comprehensive matrix for categorizing the various attack methodologies that are associated with the cyber kill chain perspective. It aims to systematically model malicious activities in terms of tactics, techniques, and procedures (TTPs), enabling an in-depth investigation of attack patterns. Each version of MITRE ATT&CK is characterized by its unique set of identifiers and TTP configurations. Specifically, version 10.1 encompasses 14 tactics (e.g., Initial Access, Lateral Movement, and Impact), 188 techniques (e.g., Process Injection, PowerShell, and Masquerading), and 379 sub-techniques (e.g., Network Device Authentication and Vulnerability Scanning). This paper focuses on version 10.1, acknowledging the variability such as the number of techniques across the different versions. Note that emerging cyber threats continuously introduce new TTPs, as the threat landscape evolves. For instance, the recent rise of cryptocurrency-related activities and ransomware has led to the incorporation of 'T1657' (Financial Theft) in the MITRE ATT&CK framework. When the MITRE ATT&CK framework is updated, the new version typically requires the re-execution of all TTP-related components, including (i) campaign generation through mutation and (ii) campaign embeddings for group attribution. MuCamp has a distinct

advantage in these scenarios due to its lightweight architecture, as will be discussed in Section VI.

Since tactical CTI based on TTPs employs a high-level approach, its accuracy might be lower than that of CTI with low-level logs. However, addressing large-scale attacks requires an abstract approach, as large-scale attackers often employ varied patterns despite having similar tactical sequences. Additionally, we rely on the MITRE ATT&CK framework that is widely used by scholars and organizations to define the scope of attacks. In experiments, the dataset was labeled with TTP identifiers such as 'TA0003.T1556.004', where 'TA0003' denotes the tactic (Persistence), 'T1556' the technique (Modify Authentication Process), and '004' the sub-technique (Network Device Authentication).

B. Data Collection

We used 858 security reports from the APT & Cybercriminals Campaign Collection [44] as source data because they cover real-world cases. Each report in our dataset is mapped to a unique cyber campaign, providing insights into the attack group that has conducted the campaign. Although several methods for automatic TTP labeling of security reports exist, such as rcATT [45] and TRAM [46], the lack of sufficient training data can result in skewed distributions during the labeling process. To mitigate the biases inherent in model-based labeling due to limited training data, we utilized cybersecurity experts to perform TTP labeling, integrating their domain expertise with practical knowledge of cybersecurity incidents. To minimize human errors during expert labeling, we engaged multiple experts (six) to crosscheck the labeling results.

Furthermore, these experts considered the context of each APT attack phase when assigning TTP labels. For instance, suppose a report mentions '*Keylogging to capture passwords otherwise obscured from viewing*.' The technique 'Keylogging' is identified as 'T1056.001', which is linked to multiple tactics such as Credential Access (TA0006) and Collection (TA0009). Considering the context of data collection, the experts labeled it 'TA0009.T1056.001.' Similarly, each security report comprises a sequence of TTPs. We assume that a sequence of tagged TTPs represents a cyber campaign; thus, we focus on these sequence data in our experiments.

IV. System Model and Security Model

To clarify the scope of the problem, we describe the system and security models considered in this work. The system model outlines the preliminaries to understand group attribution, while the security model describes the considered attack scenarios, providing the details for each attack step. According to the system and security models, we make two key assumptions: (i) cyber campaigns are represented as sequences of TTPs, and (ii) our analysis specifically targets group attribution scenarios associated with the Lazarus group.

A. System Model

Among the various topics in tactical CTI, we focus on group attribution as defined in Camp2Vec [30], which enables

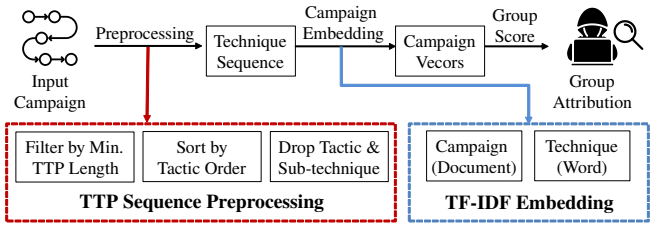


Fig. 2. Attack group attribution via Camp2Vec [30]. Preprocessing consists of three steps: (i) filtering the input TTP sequences by a minimum TTP length, (ii) sorting the TTPs according to the tactic order defined in MITRE ATT&CK, and (iii) removing tactics and sub-techniques from the TTPs. After preprocessing, the TF-IDF is used to embed the technique sequence to form campaign vectors. To measure the similarity between the vectors and the target attack groups, we used a group score based on cosine similarity.

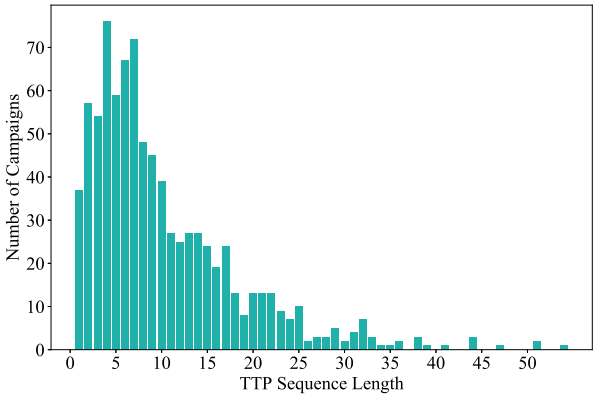


Fig. 3. Distribution of TTP lengths in the campaign. During preprocessing, we set the minimum TTP length to 5.

lightweight statistics-based embedding and group detection. Compared with traditional deep learning-based approaches, this group attribution may exhibit a relatively low performance due to its simplicity. However, given that the extreme sparsity of campaign data prevents the utilization of deep learning, attribution via Camp2Vec can be a reasonable alternative. As shown in Fig. 2, group attribution consists of three steps: (i) TTP sequence preprocessing, (ii) campaign embedding with the TF-IDF, and (iii) attribution via a group score that is based on cosine similarity.

1) Preprocessing: Considering that practical solutions, such as Kibana from the ELK Stack, provide MITRE ATT&CK-relevant information, we assume that the input to group attribution has the form of a TTP sequence. The preprocessing for a given TTP sequence includes three phases. First, we filtered the input based on a minimum length criterion, which is defined as the number of TTPs in the input. Fig. 3 shows the length distribution for 858 campaigns, indicating that the number of campaigns per TTP length tends to increase up to a length of 4 and then decreases. We therefore set a minimum length of 5 after considering data sufficiency and campaign pattern retention. Second, we reordered the input TTPs according to the tactic orders defined in ATT&CK, aligning with the cyber kill chain model. This reordering is necessary as the original TTP sequence itself does not reflect the actual attack process since its order originates from the

TABLE I
TECHNIQUE SEQUENCE AFTER PREPROCESSING.

Type	TTP Sequence
Raw	TA0001.T1566.002 → TA0005.T1070.003 → TA0002.T1204.002 → TA0003.T1547.001 → TA0008.T1021.002 → TA0011.T1071.001
Preprocessed	T1566 → T1204 → T1547 → T1070 → T1021 → T1071

contents of security reports. Third, we excluded tactics and sub-techniques to retain only the *techniques* from the TTPs, reducing the number of TTP categories and focusing on the most crucial features. An example of a preprocessed TTP sequence with a length of 6 is presented in Table I.

2) Cyber campaign embedding: While campaign embedding is not limited to a specific algorithm, we considered the term frequency-inverse document frequency (TF-IDF) as used in Camp2Vec [30]. Despite the simplicity of TF-IDF, the study demonstrated the possibility of campaign embedding by considering the relationship between a document and words to be similar to that between a campaign and techniques. The tf and idf in Camp2Vec are defined as follows:

$$tf(t,c) = log\left[1 + freq(t,c)\right], \tag{1}$$

$$idf(t) = log\left[(1+n)/(1+df(t))\right], \tag{2}$$

where t represents the technique, c denotes the campaign, $freq(t,c)$ is the frequency of the technique in the campaign, n is the total number of campaigns, and $df(t)$ represents the number of campaigns in the campaign set that contain the technique. A higher TF-IDF value indicates a technique that is frequent in certain campaigns but less common in others, thus emphasizing its importance.

3) Group attribution: According to the group attribution presented in [30], we inferred the group similarities from the campaign similarities. The campaign similarity is calculated as follows:

$$similarity(x,y) = cos(\theta) = \frac{x \cdot y}{\|x\|\|y\|}, \tag{3}$$

where x and y represent the embedded campaign vectors. The set of attack groups is expressed as:

$$G = \{G_1, G_2, \cdots, G_m\}, \tag{4}$$

where m is the number of groups, and each G_i has varying numbers of campaigns. We define *group attribution* as the process of determining the most likely attack group that has operated the input campaign. This attribution involves calculating a *group score* by averaging the similarities between the input campaign and past campaigns of each group. The group score for group G_i is defined as:

$$score_i = \frac{1}{N_i} \sum_{n=1}^{N_i} similarity(vec_{input}, vec_n), \tag{5}$$

where N_i is the number of campaigns in G_i. The attack group with the highest score for the input campaign is attributed to be the threat actor. Note that successful group attribution provides valuable insights into the attack group, including key targets, employed tools, and tactical patterns. For instance, if the attribution points to Lazarus, we should consider that Lazarus frequently targets high-value industries, such as financial institutions, cryptocurrency exchanges, and the energy sector. The group typically employs tools such as Mimikatz and Cobalt Strike, and utilizes tactics related to credential theft and data exfiltration; this enhancement in group attribution plays a crucial role when preparing appropriate mitigations.

B. Security Model

This study focuses on large-scale attacks with malicious intent (e.g., exfiltration to the target system), the details of which are presented in Section VI. Since we assume a system model based on group attribution, we need to prepare the test campaign scenario to evaluate group score-based attribution. Considering that preparing a high-quality scenario is costly, we chose the most significant group as the scenario operator by using two criteria: (i) the number of conducted campaigns and (ii) the negative social impact caused by that group. As a result, we chose Lazarus as the target group because it has conducted many campaigns, including the Sony Pictures Hack and WannaCry ransomware, which have caused extensive international damage.

We hired offensive security researchers (red team) to design a realistic Lazarus-style scenario drawing inspiration from a security report [47]. In this scenario, the threat actor utilizes document malware called ThreatNeedle to compromise industries. The attack begins with spear-phishing emails containing COVID-19 information for initial access. Upon gaining entry, the attacker deploys a remote access trojan (RAT) to establish persistence and facilitate lateral movement across Windows servers. The RAT activates the SSH protocol on the victim's system and gathers data, leading to data exfiltration. This scenario is characterized by two principal aspects: (i) the simplicity of testing and detection, and (ii) compatibility with X86 operating systems. During the experiment, our red team performed a penetration test following this scenario, and we utilized Elastic Kibana to supply a sequence of TTPs for the campaign scenario, enabling us to assess the group attribution performance after campaign generation.

V. TTP DATA AUGMENTATION WITH EDA

Before explaining the architecture of MUCAMP in detail, we describe the TTP data augmentation method proposed in [37]. To the best of our knowledge, the research [37] is the only study with the aims of generating campaign data. As shown in Fig. 4, the authors preprocessed the input features of the CTI reports via tokenization, cleaning, and stemming

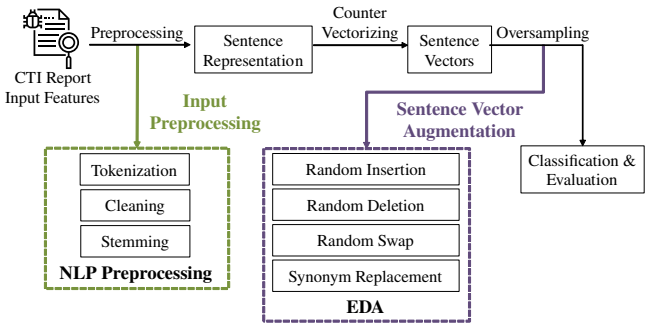


Fig. 4. Oversampling via easy data augmentation (EDA) for vectorized campaigns [37]. Given the input features of CTI reports, NLP methods (tokenization, cleaning, and stemming) are used to conduct preprocessing for sentence representation. To improve the classification performance, oversampling (e.g., EDA consisting of four augmentation methods) after sentence vectorization can be an effective solution.

techniques. After preprocessing, they produced sentence vectors via embedding and then employed easy data augmentation (EDA) [48] to augment the vectors, improving the sentence classification performance. EDA is a representative text augmentation method that uses four sub-methods to transform the original sample: random insertion (RI), random deletion (RD), random swap (RS), and synonym replacement (SR). RI selects words (excluding stop words) within a sentence and inserts them at randomly selected positions, whereas RD removes random words with a specific probability. RS randomly chooses two words in a sentence and swaps their positions, and SR replaces certain words with synonyms, where the main challenge is to define *synonym*. Despite its simplicity, EDA improves text classification performance especially on small datasets [48], which enables a *few-shot* generation approach.

Although previous work [37] has demonstrated the possibility of TTP augmentation with EDA, this study has several limitations, particularly in terms of its inadequate consideration of the characteristics of large-scale campaigns. First, cyber campaigns are not isolated security events but are methodically orchestrated as long-term operations often aimed at achieving specific objectives. This characteristic necessitates understanding the underlying objectives of these attacks, ensuring that their fundamental tactical goal is preserved during the data augmentation process. Second, applying the four existing EDA methods requires more consideration of each method's distinct properties in terms of their relevant domain knowledge. For example, RD can inadvertently eliminate crucial TTP information, thereby obscuring the patterns of attack groups. Furthermore, the length of TTPs in a campaign can signify the unique characteristics of a group, making it a crucial factor. These challenges motivate our research in terms of developing a more sophisticated method for campaign generation, which is firmly rooted in a comprehensive understanding of campaign-specific knowledge.

VI. MuCamp Design

This paper proposes MuCamp, a lightweight but effective campaign variant generation method that considers security domain knowledge. In this section, we describe (i) the design goals, (ii) the MuCamp process, and (iii) the improved group attribution after data enhancement with MuCamp.

A. Design Goals

MuCamp aims to achieve three design goals: (i) few-shot generation, (ii) domain knowledge reflection, and (iii) the generation of high-quality campaigns. We briefly summarize our approach to achieving each design goal with the MuCamp components.

- **Few-shot campaign generation.** The simple but effective few-shot generation approach is to leverage text augmentation within the domain of natural language processing (NLP), recognizing that a campaign (i.e., the TTP sequence) can be conceptualized as a form of *sequence* data. Specifically, we introduce a campaign mutation technique that involves substituting TTP words with their synonyms. We define a *TTP synonym* for a specific technique as an alternative technique that falls within the same tactical category. Since MuCamp uses lightweight augmentation strategies without neural networks, it is capable of operating with standard CPU resources, thus avoiding the need for high-performance GPUs. In Section VII-E, we will evaluate the feasibility of MuCamp to demonstrate its suitability for low-complexity environments, even in scaled-up scenarios.
- **Design with domain knowledge.** Unlike traditional security threats, large-scale attacks have a clear objective (i.e., goal tactic) that develops as part of a long-term process. The *goal tactics* therefore need to remain unchanged during mutation to guarantee consistent objectives. In addition, the length of the TTP sequence may be a crucial component, considering that the long sequence satisfies the entire process of the cyber kill chain. In Section VII-C and Section VII-D, we also analyze the importance of the TTP sequence length.
- **Validity of the generated campaign.** Ensuring the quality of generated campaigns is necessary but challenging due to their complexity. We therefore prepare a reliable campaign dataset for experiments by employing expert-based labeling. Then, based on the dataset, we quantify the improvement in group attribution [30]. Furthermore, we investigate the impacts of the parameters, such as the mutation level (Section VII-A) and generation amount per seed (Section VII-B), on group attribution.

While we anticipate that the group attribution will improve after data augmentation, this is not our sole objective. Instead, our primary focus involves generating valid mutated campaigns while considering diverse security aspects. The detailed campaign mutation process is described in the following sections.

B. MuCamp Architecture

As shown in Fig. 5, MuCamp architecture consists of two phases: (i) seed selection and (ii) campaign mutation. Upon receiving an input TTP sequence in the format defined in Section III-A, MuCamp generates campaign variants, each

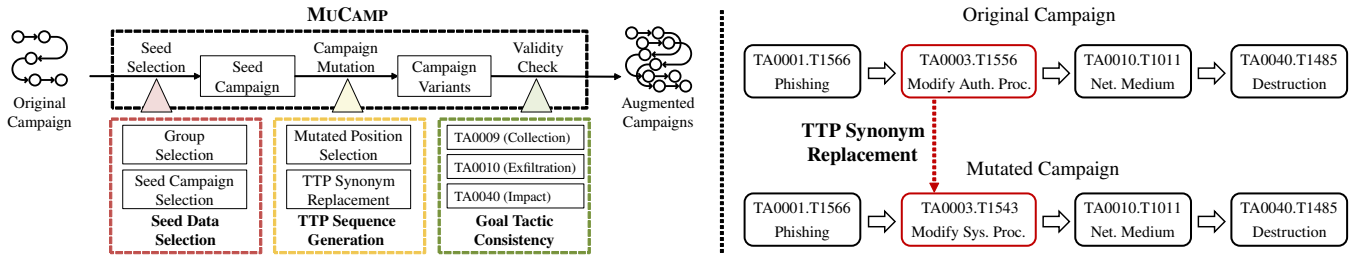


Fig. 5. MuCamp architecture. The primary elements of MuCamp comprise *seed selection* and *campaign mutation*. During seed selection, upon identifying a target group for augmented campaigns, MuCamp selects seed campaigns based on two criteria: (i) the presence of the goal tactic and (ii) the length of the TTP sequence. Subsequently, during campaign mutation, MuCamp identifies specific positions within the seed campaigns for mutation and substitutes them with corresponding TTP synonyms. This step ensures the validity of the mutated campaigns, maintaining the goal tactic's consistency throughout the mutation process. The figure to the right illustrates an example of a TTP synonym replacement, where TA0003.T1556 (Modify Authentication Process) is replaced with TA0003.T1543 (Create or Modify System Process).

bearing the same label as the seed campaign. This section describes the core components of MuCamp, emphasizing the security considerations.

1) Group selection: Because MuCamp focuses on attack group attribution as its system model, mutated campaigns are tagged with identical group information to seed campaigns, e.g., Lazarus, MenuPass, and Ajax Security. In other words, MuCamp aims to augment campaign data by *mutating* the target group's seed campaigns. Thus, the first step in MuCamp is to decide which groups to handle. While MuCamp can generate campaign variants for any attack group, we focus on the Lazarus variants, as described in Section IV-B.

2) Seed campaign selection: After selecting the target group, MuCamp selects *seed* campaigns for mutation. The two criteria for the seed campaign are (i) the inclusion of goal tactics and (ii) the length of the TTP sequence. First, the most crucial information related to the campaign is attackers' intent since we are considering large-scale attacks. Upon examining the 14 tactics outlined by MITRE ATT&CK [49], we identify three tactics as *goal tactics*: Collection (TA0009), Exfiltration (TA0010), and Impact (TA0040). The rationale behind the selection of goal tactics lies in their pivotal roles in facilitating attackers' objectives via information gathering (Collection), network-based data stealing (Exfiltration), and system disruption (Impact). We assume that these three goal-oriented tactics should remain consistent even during mutation when considering campaigns in which at least one of the target tactics is a seed candidate. Second, after selecting the campaigns with these goal tactics, we select those with lengthy TTP sequences as the seed campaigns. Since this work assumes that cyber campaigns have the form of a *sequence*, the length of the campaign indicates the combination of each attack phase. In other words, a lengthy sequence tends to follow a continuous and sophisticated attack process that is conceptualized by the cyber kill chain. Conversely, abbreviated TTP sequences might obscure identifiable patterns of attack groups, which motivated us to consider the TTP sequence length as a significant factor.

3) Mutated position selection: After seed selection, we decide which TTP position in the seed campaign to mutate. First, we exclude TTPs for goal tactics from mutated positions to guarantee the *goal consistency* of the MuCamp. Selecting a TTP location based on a specific metric causes deterministic campaign generation given a seed campaign. To ensure generation diversity, we randomize the modified TTP locations, except for the goal tactics' locations. We then need to consider how many locations to mutate (i.e., the mutation level) because drastic changes will break the attack group characteristics held by the seed campaign. In contrast, minor changes limit generation diversity, leading to campaign reproduction rather than mutation. We investigate the impact of the mutation level in Section VII-A.

4) TTP synonym replacement: The next step is to modify the selected TTP of the seed campaigns. Motivated by a previous study [48], we use EDA, which specifically focuses on synonym replacement. We excluded random swaps, insertions, and deletions for the following reasons.

- **Random swap (RS).** As cyber campaigns have become increasingly sophisticated, even identical attack groups adopt various tactics from a cyber kill chain perspective. We excluded RS to add TTP modifications for our campaign generation and to reflect the practical attack group rather than simply sorting or swapping existing campaigns.
- **Random insertion (RI).** We can augment TTP sequences by adding random TTPs to the original campaign data. However, RI is not suitable when considering goal tactic consistency, as we want the tactics intended by the original seed campaign (a key concept in cyber kill chains) to remain unchanged.
- **Random deletion (RD).** Removing randomly selected TTPs has consequences that are similar to those incurred by the use of RI. Deleting a few words in the traditional NLP domain does not significantly damage the meaning since sentences are generally sufficiently long; however, RD for campaign data can significantly contaminate the intent or other characteristics, considering that 32.98% of our data have a TTP length of less than 5.

Therefore, we focus on synonym replacement to generate TTP sequences by mutating selected TTP positions. Note that while diversity in generated data is one of the primary considerations, excessive variation can compromise the validity of the generated sequences. Considering the appropriate level of diversity in the definition of a *TTP synonym*, we applied two

criteria: (i) maintaining validity in alignment with cyber kill chain principles and (ii) achieving sufficient diversity through random selection while ensuring a reasonable level of validity. The MITRE ATT&CK framework categorizes techniques that correspond to similar phases of an attack into the same tactic within the cyber kill chain, thereby establishing connections between the techniques that fall under the same tactic. In MUCAMP, the *TTP synonym* is defined as a technique that is randomly selected from the corresponding tactic. Note that the degree of similarity between techniques can vary, even within the same tactic. For instance, in the case of the Persistence (TA0003) tactic, the Account Manipulation (TA0003.T1098) and Create Account (TA0003.T1136) techniques may share more similarities with each other than with the Traffic Signaling (TA0003.T1205) technique. However, incorporating the TTP similarity into synonym replacement could excessively constrain generation diversity, as it would result in a fixed synonym for each technique, thereby reducing the variability necessary for robust data generation. In addition to diversity, we investigate the impact of the generation quantity, denoted by N_m, on effective group attribution in Section VII-C.

5) Validity check: The final step involves checking the validity of the generated TTP sequences, ensuring that the fundamental nature of the original campaigns remains consistent throughout the mutation process. Since we align cyber campaigns with cyber kill chains and MITRE ATT&CK, we assume that the nature of these campaigns depends on the objectives represented by three goal tactics. By excluding goal tactics from the candidates of the mutated positions, the ultimate objectives of the campaigns are preserved even after TTP synonym replacement. The reason for each goal tactic in the context of real-world campaign objectives is as follows.

- **Collection (TA0009).** One of the key objectives of large-scale campaigns is the acquisition of sensitive information, such as intellectual property, personal data, or confidential documents. In real-world campaigns, attackers often employ automated tools to continuously gather data over extended periods.
- **Exfiltration (TA0010).** This tactic is especially critical in the final stages of data theft scenarios. To achieve Exfiltration as a goal tactic, large-scale campaigns frequently utilize sophisticated methods designed to evade detection, such as encrypted channels or disguising malicious traffic as legitimate communication.
- **Impact (TA0040).** We commonly observe this tactic in destructive cyber campaigns, which aim to execute malware attacks or deploy ransomware. This tactic seeks not only to steal data but also to cause damage, undermine trust, or apply pressure on targeted organizations or governments.

If MUCAMP operates as anticipated, mutated campaigns will always meet the criteria of the validity check, given its pre-consideration during the mutated position selection phase. After campaign generation, the attack group attribution is enhanced by data augmentation as described in the next subsection.

Algorithm 1: Data-Augmented Group Attribution

Input: Source campaign X_{src}; Scenario data X_{SCN}; Minimum TTP length l_{min}; Mutation level α; Seed selection ratio β; Number of mutations per seed N_m.
Output: Expected group y^*.

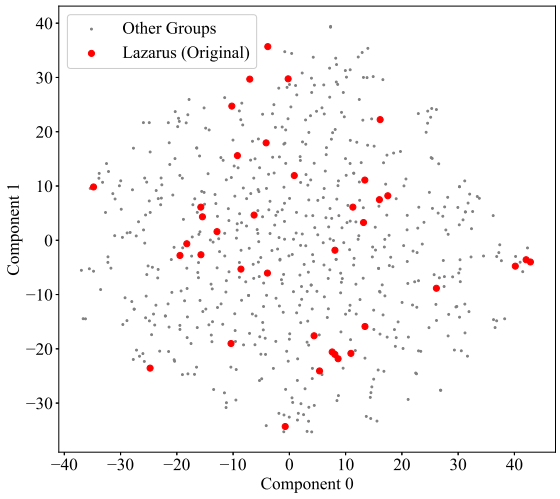
/* Step 1: Dataset Preparation */
Conduct TTP Tagging and prepare X_{src}.
Preprocess X_{src} with l_{min}.
Mutated campaigns $X_{\text{mut}} \leftarrow []$
/* Step 2: Campaign Generation by MUCAMP */
$G \leftarrow$ SelectGroup(X_{src})
$C_{\text{seed}} \leftarrow$ SelectSeedCampaign(X_{src}, G, β)
for $i = 1$ to N_m **do**
  $pass_{\text{validity}} \leftarrow$ False
  while $pass_{\text{validity}}$ == False **do**
    $p \leftarrow$ SelectMutatedPosition(C_{seed}, α)
    $C_{\text{mut}} \leftarrow$ ReplaceSynonym(p)
    if GoalTactic(C_{mut}) == GoalTactic(C_{seed}) **then**
      $pass_{\text{validity}} \leftarrow$ True
      Append C_{mut} to X_{mut}.
    end if
  end while
end for
/* Step 3: Attack Group Attribution */
Vectorizer $\leftarrow$ Train(Camp2Vec, X_{src})
$X_{\text{aug}} \leftarrow X_{\text{src}} + X_{\text{mut}}$
$c_X, c_{\text{SCN}} \leftarrow$ Embedding(Vectorizer, X_{aug}, X_{SCN})
$y^* \leftarrow$ GroupAttribution(c_X, c_{SCN}).

C. Data-Augmented Group Attribution

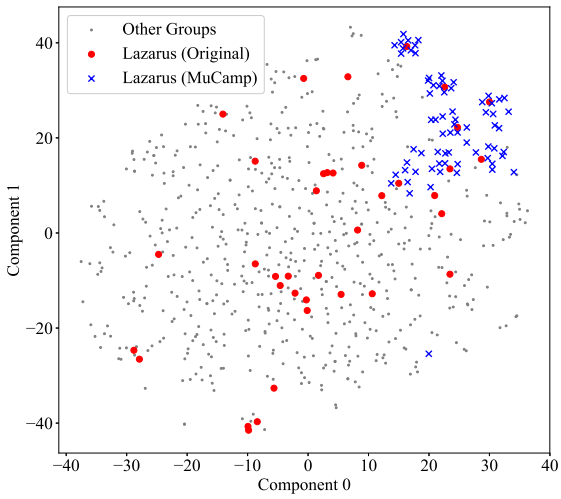
MUCAMP defines the output format of generated campaigns as a TTP sequence rather than as the campaign vectors suggested in [37]. TTP sequence generation allows us to pursue data-augmented group attribution independently of the embedding method, although we use Camp2Vec [30] as a reference. We assume that data-augmented group attribution consists of three steps: (i) dataset preparation, (ii) campaign generation via MUCAMP, and (iii) group attribution via campaign vectors. Algorithm 1 illustrates an example pseudocode for data-augmented group attribution. Initially, we collect and preprocess the original campaign data as described in Section III and Section IV, including source campaign X_{src} and scenario campaign X_{SCN}. Subsequently, we generate campaign variants via MUCAMP in a process that involves four steps: group selection, seed campaign selection, mutated position selection, and TTP synonym replacement. Then, we conduct a validity check to guarantee that the defined goal tactics of seed campaigns remain consistent. The final stage encompasses embedding-based attack group attribution. After training Camp2Vec [30] with the original campaigns X_{src} (excluding the generated data), we apply the trained vectorizer to both the augmented campaigns and the scenario campaign. Given the campaign vectors, we next execute group attribution by identifying the group with the highest group score [30].

To provide an intuitive understanding of how campaign mutation aids in attack group attribution, we visualize the

embedded campaign vectors on a two-dimensional plane. As shown in Fig. 6(a) and Fig. 6(b), we use t-distributed stochastic neighbor embedding (t-SNE) with 1000 iterations for dimensionality reduction. The displayed 634 dots indicate the original campaign data filtered with a minimum TTP length of 5, and the crosses represent mutated campaigns. We generate campaign variants via MUCAMP by setting the parameters α (mutation level) to 0.5, β (seed selection ratio) to 0.2, and N_m (number of mutations per seed) to 10. As shown in Fig. 6(a), the distribution of the embedded vectors does not exhibit a distinct pattern. This lack of clarity is due to the increasing sophistication of attack groups and the advancement of their campaigns. However, as shown in Fig. 6(b), mutated campaigns tend to cluster and form specific patterns. These observations suggest that MUCAMP has the potential to enhance embedding-based attack group attribution. In the following section, we describe a quantitative analysis that was conducted by using various parameters to further explore this potential.



(a)



(b)

Fig. 6. Visualization with t-SNE on the embedded campaign vectors of the (a) original (without augmentation) and (b) MUCAMP-augmented campaigns. After data augmentation, mutated campaigns (blue crosses) are generated near the seed campaigns.

VII. PERFORMANCE EVALUATION

We prepared a dataset consisting of 858 campaigns, labeled by security experts as described in Section III. After preprocessing, we used 634 campaigns from 341 attack groups as our base data. Note that few attack groups performed multiple campaigns; 265 attack groups performed only one campaign in the dataset. To focus on the groups with multiple campaigns, we list the top 10 attack groups based on the number of campaigns as shown in Fig. 7. The group includes well-known adversaries such as Lazarus, APT28, and Deep Panda, which operated 38, 23, and 8 campaigns, respectively. In addition to the 634 campaigns, we also prepare test data with one real-world scenario from Lazarus as described in Section IV-B. Since our scenario data are constructed to reflect a diverse range of attack patterns and TTPs, the resulting TTP sequence has a length of 24.

In terms of attack group attribution, we analyzed three phases: (i) fitting Camp2Vec with the 634 campaigns, (ii) embedding the campaigns of the top-10 attack groups and the target scenario (Lazarus), and (iii) measuring the group scores among the embedded campaigns. To implement MUCAMP, we set the mutation level α to 0.2, the generation amount N_m to 10, and the seed campaign selection ratio β to 0.2 (i.e., the number of seed campaigns N_s is 7 out of 38 Lazarus campaigns). Consequently, the total number of augmented campaign variants is $N_s \times N_m = 7 \times 10 = 70$.

To compare performance, we analyzed the group scores calculated for the target scenario (Section. IV-B). Instead of comparing with other generation methods in CTI, we used MUCAMP with varying parameters as our baseline for several key reasons. While a state-of-the-art study on TTP augmentation [37] exists, its implementation details lack sufficient reproducibility and focus on vectorized TTPs rather than raw TTP sequences. We also excluded traditional augmentation methods such as SMOTE [50] from our baselines, as SMOTE is designed to balance categories, which diverges significantly from real-world cybersecurity scenarios. Since our objective is to integrate domain-specific knowledge into TTP augmentation, we emphasize a comprehensive investigation of MUCAMP with diverse parameters.

A. Effect of Mutation Level

We investigated how the mutation level α affects the group attribution, since the mutation level may affect the quality and validity of the generated data. Compared with other domains such as NLP, TTP sequences typically have short lengths, with those of 64.57% ($\frac{554}{858}$) of the data being less than 10. When we set a minimum TTP length of 5, using an α value under 0.2 might result in no mutation. In the experiments, we chose α values ranging from 0.2 to 0.7 with an interval of 0.1.

As shown in Table II, we first present the group scores without campaign augmentation. The results indicate that the attack group attribution is successful for the Lazarus scenario, achieving the highest score of 0.1662 for the correct attribution. However, the score gap between Lazarus and the other

TABLE II
GROUP SCORE BETWEEN THE LAZARUS SCENARIO AND THE SELECTED ATTACK GROUPS.

Attack Group									
Lazarus	APT28	Turla	APT32	Sandworm	MuddyWater	APT29	G-3390	menuPass	Deep Panda
0.1662	0.1302	0.1155	0.1115	0.1053	0.1058	0.0716	0.1015	0.0952	0.0698

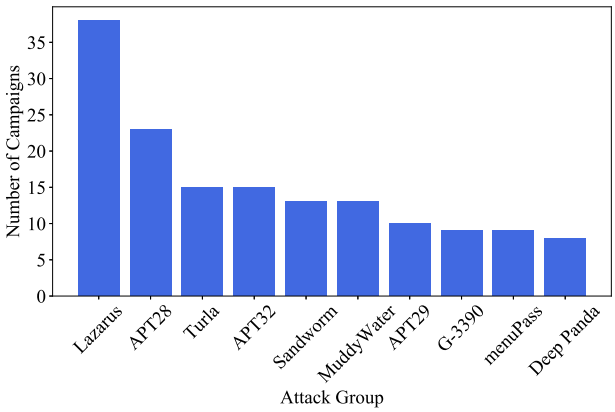


Fig. 7. Number of campaigns for the top-10 attack groups in our dataset. We focus on Lazarus as the targeted attack scenario group considering the impact and number of campaigns.

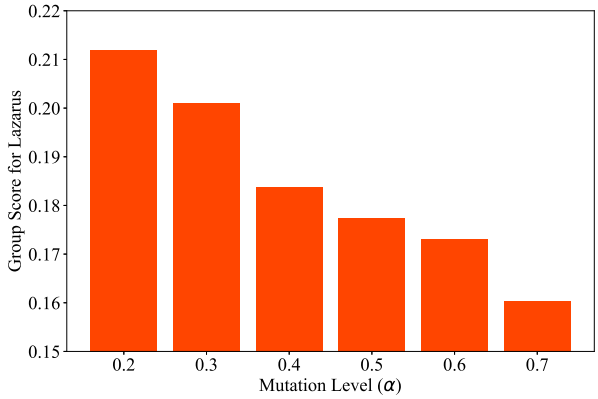


Fig. 8. Group score for Lazarus according to the mutation level α.

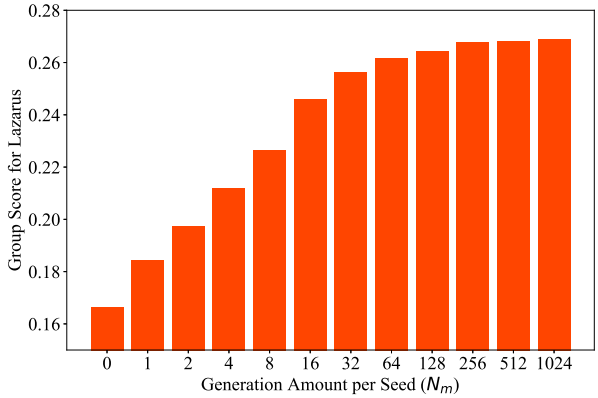


Fig. 9. Group score for Lazarus according to the campaign generation amount per seed N_m.

groups suggests room for improvement, considering that the second-highest group score is 0.1302 for APT28. Introducing valid data augmentation would increase the score gap between the correct group (Lazarus) and the other groups. Fig. 8 shows the tendency of the group scores for Lazarus, which were augmented with MUCAMP for different α values. Since we only generated the Lazarus campaigns and the augmented data were used for testing (not training), note that the scores for the other groups are consistent after augmentation and are the same as those in Table II. As shown in Fig. 8, data augmentation with an α of 0.2 increases the group score from 0.1662 to 0.2120. Notably, the score gap between Lazarus and APT28 increases by 127.2%, i.e., from 0.036 (0.1662 - 0.1302) to 0.0818 (0.2120 - 0.1302). However, the group score decreases as α increases, e.g., when α changes from 0.2 to 0.7, the score diminishes from 0.2120 to 0.1603. This decrease at higher α values is attributed to the distortion of the seed campaign's pattern, i.e., the unique characteristics of the attack group.

Note that a low α value results in only minor mutations from the original seed campaigns, leading to low generation *diversity*. This low diversity may achieve higher generation *quality*, assuming that seed campaigns are appropriately selected. Conversely, a high α value results in greater generation diversity, increasing the likelihood of mistakenly associating generated campaigns with the original seed campaigns. For instance, using an α of 0.7 lowers the group score from 0.1662 to 0.1603, demonstrating that data augmentation alone does not guarantee performance improvement. To ensure the validity and quality of the generated data, balancing this trade-off by selecting an appropriate α value is crucial.

B. Effect of Generation Amount per Seed

We suppose that if we can generate high-quality campaign data, the improvements in group attribution will likely be proportional to the generation amount. Specifically, the total generation amount is influenced by two primary factors: (i) the generation amount per seed N_m and (ii) the number of selected seed campaigns N_s. To examine the impact of N_m on the overall performance, we fixed N_s at 7 and varied N_m from 1 to 1024, doubling the value at each step. For instance, an N_m value of 4 implies a total generation amount of 28 (7×4).

Fig. 9 illustrates that the group score progressively increases with increasing N_m. However, this performance enhancement does not continue indefinitely and tends to plateau beyond a certain threshold. Nonetheless, the impact of N_m is particularly significant when limited data are available. For example, with an N_m of 64, the augmented dataset size becomes 448, i.e., the total number of campaigns associated with

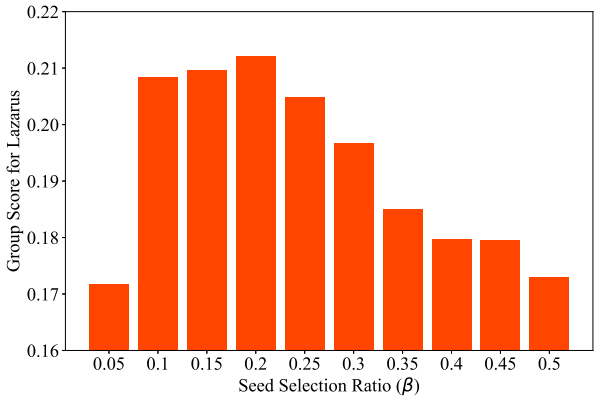


Fig. 10. Group score for Lazarus according to the seed selection ratio β. The number of seed campaigns N_s is calculated by multiplying β and the number of Lazarus campaigns (38).

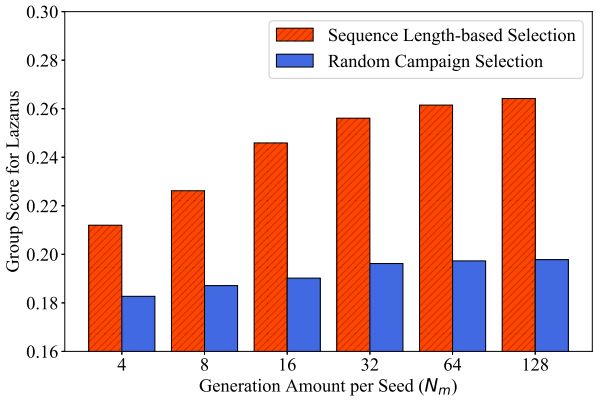


Fig. 11. Group score for Lazarus with different seed selection algorithms according to the campaign generation amount per seed N_m.

Lazarus expands from 38 to 486 (38 + 448). This campaign data augmentation substantially enhances the group score by 57.34%, i.e., improving from 0.1662 to 0.2615.

C. Effect of Seed Selection Ratio

In addition to N_m, an alternative approach to increasing the generation amount involves increasing the number of selected seed campaigns N_s. The value of N_s is derived by multiplying the seed selection ratio β by 38, which represents the number of original campaigns from the Lazarus group. Remember that we selected seed campaigns according to two criteria: (i) the inclusion of goal tactics and (ii) the length of the TTP sequence. A higher N_s involves incorporating shorter TTP sequences that might contain less comprehensive information about the target group, potentially affecting the attribution performance. In our experiments, we fixed N_m at 4 and varied β from 0.05 to 0.5 with an interval of 0.05.

As shown in Fig. 10, the group score improves to a β of 0.2, after which it begins to decrease. Specifically, the group score reaches 0.2120 when β is set to 0.2 and decreases to 0.1729 with a β of 0.5. This decline in score with higher β suggests that the seed campaign, in such cases, captures limited characteristics of the target group. In other words, due to overly broad seed selection, a higher β value causes mutations in the seed campaigns that differ significantly from the original Lazarus campaigns. Nevertheless, we observe a gradual improvement in scores at lower β values (ranging from 0.05 to 0.2), where the advantages of data augmentation outweigh the score reduction caused by increasing β. In our experiments, a β of 0.2 yields the best performance, effectively balancing the trade-off.

D. Effect of Seed Selection Algorithm

We validated a seed selection algorithm for MUCAMP that prioritizes the length of the TTP sequence. We compared this algorithm against a randomized selection method, presenting group scores for different N_m (ranging from 4 to 128, increasing in multiples of 2). We deliberately excluded the baseline of the opposite algorithm (seed selection by the *short* sequence) due to its tendency to generate excessive redundancy in campaign data.

As illustrated in Table II and Fig. 11, both the MUCAMP and random selection algorithms demonstrate an improvement in group scores across all N_m. Specifically, the scores increase from 0.1662 (without augmentation) to 0.2120 (with MUCAMP) and 0.1827 (with random selection). Note that the performance gap between the selection algorithms becomes apparent as N_m increases. This gap increases from 0.0293 (0.2120 - 0.1827) to 0.0664 (0.2642 - 0.1978) when N_m increases from 4 to 128. As discussed in Section VII-A, we have observed that the validity of the generated campaigns correlates with an improvement in scores proportionate to the increase in N_m. This observation suggests that the MUCAMP seed selection algorithm contributes to generating reasonable campaign data.

E. Feasibility Analysis

To further investigate the feasibility and complexity of MUCAMP, we measured the generation time required to produce the TTP sequences for the Lazarus group. We varied three key parameters that affect the generation time: α (mutation level), β (seed selection ratio), and N_m (generation amount per seed). Given that there are 38 distinct campaigns associated with Lazarus, a β value of 0.2 implies the selection of approximately 7 seeds (0.2×38). The total generation amount is calculated by multiplying N_m and the number of seeds. For example, with $\beta = 0.2$ (i.e., the number of seeds is 7) and $N_m = 128$, the resulting total generation amount is 896.

As illustrated in Table III, the generation time remains below one second across all of the experimental cases. This result underscores that MUCAMP is a computationally lightweight generation method, excluding the use of neural networks in its architecture. Cases (1)∼(4) present scenarios wherein the generation amount per seed, N_m, varies from 128 to 1024. The generation time linearly increases with respect to N_m, ranging from 0.0231 seconds to 0.1804 seconds. This linearity is intuitive considering that the total generation amount scales proportionally with N_m. In cases (4) and (5), although we doubled the seed selection ratio β (from 0.2 to 0.4), the generation time exhibits a modest increase of only 37.98%

TABLE III
FEASIBILITY ANALYSIS

Case	α	β	N_m	Number of Seeds	Generation Amount	Generation Time (Secs)
(1)	0.2	0.2	128	7	896	0.0231
(2)	0.2	0.2	256	7	1,792	0.0446
(3)	0.2	0.2	512	7	3,584	0.0875
(4)	0.2	0.2	1,024	7	7,168	0.1804
(5)	0.2	0.4	1,024	15	15,360	0.2489
(6)	0.4	0.4	1,024	15	15,360	0.4844

(from 0.1804 seconds to 0.2489 seconds). The reason for this relatively small increase is that a higher seed selection ratio incorporates shorter TTP sequences and seed campaigns, which subsequently demand fewer mutations and a reduced generation time. Furthermore, cases (5) and (6) show that the mutation level α linearly affects the generation time, increasing from 0.2489 seconds to 0.4844 seconds as α increases.

We have observed that MUCAMP has low computational complexity, requiring less than one second of processing time on a single CPU (Intel Xeon Gold 5215, 2.5 GHz, 10 Cores) to generate over 15,000 campaigns. Note that the number of original campaigns for Lazarus is limited to only 38, which is significantly smaller than the number of generated campaigns. This observation demonstrates the scalability and efficiency of MUCAMP, indicating that the computational complexity can be effectively managed by adjusting the generation parameters.

VIII. CONCLUSION

In this paper, we have presented MUCAMP, which improves attack group attribution in tactical CTI via campaign data augmentation. We introduced TTP synonym replacement to implement valid campaign variant generation when there is limited campaign data, effectively capturing the distinct characteristics of campaign operators, especially the consistency of their goal tactics. Experimental results on expert-labeled datasets revealed that each component of MUCAMP contributes to improving the embedding-based group attribution. MUCAMP exhibited low computational complexity, facilitating seamless adaptation to newly updated MITRE ATT&CK versions and the integration of emerging TTPs. In future work, we plan to explore its applicability to other attack groups beyond Lazarus and evaluate adaptive mutation strategies based on campaign complexity.

ACKNOWLEDGMENT

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

REFERENCES

[1] I. Lee, C. Shin, and C. Choi, "Mutating cyber camapaign with ttp word replacement," in *Proc. of the Korea Institute of Military Science and Technology*, 2023, pp. 1603–1604.

[2] M. Wazid, A. K. Das, V. Chamola, and Y. Park, "Uniting cyber security and machine learning: Advantages, challenges and future research," *ICT Express*, vol. 8, no. 3, pp. 313–321, 2022.

[3] P. Kalnai and M. Poslusny, "Lazarus group: a mahjong game played with different sets of tiles," in *Proc. Virus Bulletin International Conference*, 2018.

[4] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," *Computers & Security*, vol. 87, p. 101589, 2019.

[5] A. Mohaisen, A. G. West, A. Mankin, and O. Alrawi, "Chatter: Classifying malware families using system event ordering," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 283–291.

[6] J. Zeng, Z. L. Chua, Y. Chen, K. Ji, Z. Liang, and J. Mao, "Watson: Abstracting behaviors from audit logs via aggregation of contextual semantics." in *NDSS*, 2021.

[7] R. Yang, X. Chen, H. Xu, Y. Cheng, C. Xiong, L. Ruan, M. Kavousi, Z. Li, L. Xu, and Y. Chen, "Ratscope: Recording and reconstructing missing rat semantic behaviors for forensic analysis on windows," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1621–1638, 2020.

[8] H. Irshad, G. Ciocarlie, A. Gehani, V. Yegneswaran, K. H. Lee, J. Patel, S. Jha, Y. Kwon, D. Xu, and X. Zhang, "Trace: Enterprise-wide provenance tracking for real-time apt detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4363–4376, 2021.

[9] X. Chen, H. Irshad, Y. Chen, A. Gehani, and V. Yegneswaran, "{CLARION}: Sound and clear provenance tracking for microservice deployments," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3989–4006.

[10] S. Wang, Z. Wang, T. Zhou, H. Sun, X. Yin, D. Han, H. Zhang, X. Shi, and J. Yang, "Threatrace: Detecting and tracing host-based threats in node level through provenance graph learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3972–3987, 2022.

[11] 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.

[12] Y. Wu, M. Zhao, A. Haeberlen, W. Zhou, and B. T. Loo, "Diagnosing missing events in distributed systems with negative provenance," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 383–394, 2014.

[13] H. Wang, G. Yang, P. Chinprutthiwong, L. Xu, Y. Zhang, and G. Gu, "Towards fine-grained network security forensics and diagnosis in the sdn era," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 3–16.

[14] J. Bi, S. He, F. Luo, W. Meng, L. Ji, and D.-W. Huang, "Defense of advanced persistent threat on industrial internet of things with lateral movement modelling," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 9, pp. 9619–9630, 2023.

[15] J. Liu, Y. Shen, M. Simsek, B. Kantarci, H. T. Mouftah, M. Bagheri, and P. Djukic, "A new realistic benchmark for advanced persistent threats in network traffic," *IEEE Networking Letters*, vol. 4, no. 3, pp. 162–166, 2022.

[16] A. Spyros, I. Koritsas, A. Papoutsis, P. Panagiotou, D. Chatzakou, D. Kavallieros, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Ai-based holistic framework for cyber threat intelligence management," *IEEE Access*, 2025.

[17] C. Do Xuan, L. Van Duong, and T. V. Nikolaevich, "Detecting c&c server in the apt attack based on network traffic using machine learning," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 5, 2020.

[18] C. Do Xuan, "Detecting apt attacks based on network traffic using machine learning," *Journal of Web Engineering*, pp. 171–190, 2021.

[19] A. Alzu'bi, O. Darwish, A. Albashayreh, and Y. Tashtoush, "Cyberattack event logs classification using deep learning with semantic feature analysis," *Computers & Security*, vol. 150, p. 104222, 2025.

[20] Z. Wang, Y. Zhou, H. Liu, J. Qiu, B. Fang, and Z. Tian, "Threatinsight: Innovating early threat detection through threat-intelligence-driven analysis and attribution," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[21] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," in *Technical report*. The MITRE Corporation, 2018.

[22] C. Xiong, T. Zhu, W. Dong, L. Ruan, R. Yang, Y. Cheng, Y. Chen, S. Cheng, and X. Chen, "Conan: A practical real-time apt detection system with high accuracy and efficiency," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 551–565, 2020.

[23] S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, and D. S. Kim, "Harmer: Cyber-attacks automation and evaluation," *IEEE Access*, vol. 8, pp. 129 397–129 414, 2020.

[24] K. Kim, Y. Shin, J. Lee, and K. Lee, "Automatically attributing mobile threat actors by vectorized att&ck matrix and paired indicator," *Sensors*, vol. 21, no. 19, p. 6522, 2021.
[25] Y. Kim, I. Lee, H. Kwon, K. Lee, and J. Yoon, "Ban: Predicting apt attack based on bayesian network with mitre att&ck framework," *IEEE Access*, 2023.
[26] Y. Shin, K. Kim, J. J. Lee, K. Lee *et al.*, "Focusing on the weakest link: A similarity analysis on phishing campaigns based on the att&ck matrix," *Security and Communication Networks*, vol. 2022, 2022.
[27] Z. Song, Y. Tian, and J. Zhang, "Similarity analysis of ransomware attacks based on att&ck matrix," *IEEE Access*, 2023.
[28] M. T. Alam, D. Bhusal, Y. Park, and N. Rastogi, "Looking beyond iocs: Automatically extracting attack patterns from external cti," in *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, 2023, pp. 92–108.
[29] Y. Zhang, T. Du, Y. Ma, X. Wang, Y. Xie, G. Yang, Y. Lu, and E.-C. Chang, "Attackg+: Boosting attack graph construction with large language models," *Computers & Security*, vol. 150, p. 104220, 2025.
[30] I. Lee and C. Choi, "Camp2vec: Embedding cyber campaign with att&ck framework for attack group analysis," *ICT Express*, 2023.
[31] S. Shin, I. Lee, and C. Choi, "Anomaly dataset augmentation using the sequence generative models," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 1143–1148.
[32] P. Wang, S. Li, F. Ye, Z. Wang, and M. Zhang, "Packetcgan: Exploratory study of class imbalance for encrypted traffic classification using cgan," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7.
[33] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," in *Pacific-asia conference on knowledge discovery and data mining*. Springer, 2022, pp. 79–91.
[34] D. Biesner, K. Cvejoski, B. Georgiev, R. Sifa, and E. Krupicka, "Generative deep learning techniques for password generation," *arXiv preprint arXiv:2012.05685*, 2020.
[35] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative deep learning to detect cyberattacks for the iot-23 dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2021.
[36] E. Gionanidis, P. Karvelis, G. Georgoulas, K. Stamos, and P. Garg, "Evaluating text augmentation for boosting the automatic mapping of vulnerability information to adversary techniques," in *2022 IEEE Secure Development Conference (SecDev)*. IEEE, 2022, pp. 23–29.
[37] H. Kim, H. Kim *et al.*, "Comparative experiment on ttp classification with class imbalance using oversampling from cti dataset," *Security and Communication Networks*, vol. 2022, 2022.
[38] C. Liu, B. Li, J. Zhao, Z. Zhen, X. Liu, and Q. Zhang, "Fewm-hgcl: Few-shot malware variants detection via heterogeneous graph contrastive learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[39] P. Kuehn, M. Schmidt, and C. Reuter, "Threatcrawl: A bert-based focused crawler for the cybersecurity domain," *arXiv preprint arXiv:2304.11960*, 2023.
[40] Y.-F. Li, Y. Gao, G. Ayoade, L. Khan, A. Singhal, and B. Thuraisingham, "Heterogeneous domain adaptation for multistream classification on cyber threat data," *IEEE Transactions on Dependable and Secure Computing*, 2022.
[41] Y. Yu, W. Yang, W. Ding, and J. Zhou, "Reinforcement learning solution for cyber-physical systems security against replay attacks," *IEEE Transactions on Information Forensics and Security*, 2023.
[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.
[43] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.
[44] "Apt & cybercriminals campaign collection." [Online]. Available: https://github.com/CyberMonitor/APT_CyberCriminal_Campagin_Collections
[45] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated retrieval of att&ck tactics and techniques for cyber threat reports," *arXiv preprint arXiv:2004.14322*, 2020.
[46] "The center for threat-informed defense, tram is an open-source platform designed to advance research into automating the mapping of cyber threat intelligence reports to mitre att&ck." [Online]. Available: http://github.com/center-for-threat-informed-defense/tram/
[47] "Lazarus targets defense industry with threatneedle." [Online]. Available: https://ics-cert.kaspersky.com/publications/reports/2021/02/25/lazarus-targets-defense-industry-with-threatneedle/
[48] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.
[49] "MITRE ATT&CK Enterprise Tactics," accessed: 2023-10-04. [Online]. Available: https://attack.mitre.org/tactics/enterprise/
[50] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

Insup Lee (S'20) received the B.E. degree in Cyber Defense from Korea University, Seoul, Republic of Korea, in 2018, where he is currently pursuing his Ph.D. degree in cybersecurity. From 2018 to 2023, he was a researcher at the Cyber Technology Center, Agency for Defense Development, Republic of Korea. He is currently a cyber officer at the Ministry of National Defense, Republic of Korea. His research interests include generative models, AI-driven security, adversarial machine learning, and secure communications.

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), Daejeon, South Korea, as a Senior Researcher. Since 2025, he has been with the Department of Cyber Defense at Sejong University, Seoul, South Korea, where he is currently an Associate Professor. His research interests include AI-based cybersecurity, generative models, proactive cyber defense technologies, and digital image forensics.