Efficient Packet Payload Feature Extraction Using the BIGBIRD Model

Son A. Pham[1,*] and Yasuhiro Nakamura[2]

[1]Graduate School of Science and Engineering, National Defense Academy, Yokosuka, Japan
[2]Department of Computer Science, School of Electrical and Computer Engineering, National Defense Academy, Yokosuka, Japan
Email: pisonnda@gmail.com (S.A.P.); yas@nda.ac.jp (Y.N.)
*Corresponding author

Manuscript received February 20, 2024; revised March 22, 2024; accepted June 17, 2024; published January 9, 2025

Abstract—**In recent years, the rise in cyber-attacks on the Internet has become a major concern. Addressing these threats requires continuous monitoring and analysis of communication patterns in cyberspace. However, the large volume and diverse nature of incoming packets and payloads present a challenge for simultaneous processing. Preliminary clustering of payloads is essential for subsequent analysis and interpretation. Previous studies have explored the use of natural language processing models such as N-gram, Word2Vec, and Bidirectional Encoder Representations from Transformers (BERT) to identify and categorize payloads, aiming to extract payload features for distinguishing between benign and malicious content. However, these models overlook the sequential order and byte-level positioning within payloads, thus limiting their effectiveness in capturing the intrinsic characteristics of payload content. This study introduces a novel model, which effectively extracts comprehensive features of payload content by considering the sequential ordering of byte sequences. Comparative experiments demonstrate that the clustering rate and clustering accuracy of the proposed method surpass those of other text feature extraction models such as N-gram, Word2Vec, and BERT, even when using the same clustering models. Moreover, the practical applicability of the proposed model is validated through its adaptation to actual observed data. This research significantly contributes to the field of cybersecurity and is expected to lead to future advancements and applications in this domain.**

Keywords—**clustering, features extraction, computer network and communications, network observation**

I. INTRODUCTION

In conjunction with the expansion of the Internet and the proliferation of IoT devices, there is a continuous increase in the frequency of cyber-attacks and data breaches [1, 2]. This situation represents a significant and serious issue, necessitating urgent measures for defense and prevention.

To prevent and detect these attacks, identifying early signs and risks at the initial stages is of paramount importance. According to the Cyber Kill Chain model [3], the first phase of an attack involves reconnaissance. During this phase, attackers send a large volume of packets to numerous IP addresses, including unused IP addresses, to investigate operational status and detect vulnerabilities. Based on this foundation, a variety of observation systems have been deployed to investigate these reconnaissance activities. Among these, systems that collect packets sent to unused IP addresses are known for their cost-effectiveness and efficiency. Prominent examples of such models include deployments by National Institute of Information and Communications Technology (NICT) [4], Center for Applied Internet Data Analysis (CAIDA) [5], and Israel InterUniversity Computation Center/ InterDisciplinary Center (IUCC/IDC) Network Project [6], The Honeynet Project [7], Internet Storm Center [8].

The analysis of collected data has significantly contributed to various aspects of Cyber Security. Specifically, in the field of research on IoT Malware and Worm activities [9–13], studies related to the detection of DoS and DDoS attacks [14–16], and reports providing an overview of cyber space trends and conditions [17–19] have been notably beneficial. However, research to date has primarily focused on analyzing features within collected packet headers, with a very limited number of studies concentrating on the payloads of packets sent to unused IP addresses. By focusing solely on the characteristics of packet headers, these studies are limited to understanding trends and changes in quantity and are restricted in predicting the purpose of the transmissions and the actual targets of the dispatched packets.

By analyzing the payloads contained in these packets, we can discern the content of the packets and the intentions of the sender. Corresponding to the volume of packets sent across cyberspace, the number of payloads sent to unused IP addresses is vast, involving many different protocols, and is highly diverse in terms of type and data format.

Due to the overwhelming quantity and variety of payloads, analyzing all payloads poses significant challenges. An essential first step for effective analysis of these payloads is to cluster them into groups with similar characteristics. Previous studies have attempted to address this objective by adopting existing techniques such as N-gram, Word2Vec, BERT, and machine learning approaches. However, there are limitations in effectively extracting features for payloads with complex and diverse characteristics, including network communications and encrypted traffic, which continue to evolve alongside the advancement of the Internet.

Therefore, this study aims to clustering diverse payloads, including encrypted communications, and proposes an effective method for feature extraction using the BIG-BIRD model to address the challenges of payload feature extraction.

To demonstrate the effectiveness and practicality of the proposed method, Chapter II will review previous studies related to the extraction of payload features and payload clustering. Chapter III will introduce the proposed method, and Chapter IV will conduct experiments comparing it with previous models and studies, thereby proving the effectiveness of the proposed method. Chapter V will apply the proposed method to the clustering of real-observation data, demonstrating its practicality.

II. RELATED WORKS

Before introducing previous studies, it is essential to

DOI: 10.7763/IJCTE.2025.V17.1363

present some concepts and associated knowledge.

A. About N-gram

N-grams are a fundamental concept in linguistic analysis and natural language processing, creating “n” sequential items from text or speech. These items can be as small as phonemes or as large as words, and the length of the sequence can vary from unigrams (1-gram) to bigrams (2-gram), trigrams (3-gram), and beyond. The core operating principle is to break larger text and speech samples into smaller chunks to facilitate analysis and prediction. N-Grams offer simplicity, efficiency, and solid basic performance in many NLP tasks, and their flexibility allows them to be applied to a variety of language levels. Higher-order N-Grams can capture more context and improve the accuracy of language models. However, N-Grams also have drawbacks. The size of the model increases exponentially with “n” leading to scalability issues. Higher n-Grams may encounter data scarcity and generalization challenges, as certain sequences may be rare. In addition, the limited context window of ‘n’ items may miss longer-term dependencies, making the effectiveness of the model highly dependent on the training corpus and affecting its generalizability.

B. About Word2Vec

Word2Vec is a set of models developed by Tomas Mikolov’s team at Google, designed to represent words as high-dimensional vectors and capture context, semantic and syntactic similarity, and relationships with other words. The models are renowned for their ability to understand linguistic context and similarity, such as relating “king” to “man” and “queen” to “woman”. The model works with two architectures: the Continuous Bag of Words (CBOW) and Skip-Gram, where CBOW predicts words based on context and Skip-Gram predicts context from words. Training adjusts word vectors through a neural network to encode word relationships.

The advantages of Word2Vec include semantic comprehension ability, training efficiency, dense and informative word representations, and generality for different tasks and domains. However, there are limitations, including the inability to distinguish between multiple meanings of a word in different contexts (polysemy), the need for large data sets for effective training, the difficulty with non-lexical words, and the fact that training, despite being relatively efficient, is computationally intensive and time consuming.

1) About BERT (Bidirectional Encoder Representations from Transformers)

Developed by Google in 2018, BERT is a major advance in NLP by understanding the context of words in a sentence in both directions and considering the words before and after each word. It is based on the Transformer architecture, which uses a self-attention mechanism to weight the importance of each word in relation to other words; BERT training includes Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) to obtain rich linguistic understanding.

The advantages of BERT include deep contextual understanding, state-of-the-art performance on a variety of NLP tasks, fine-tuning versatility for different tasks, and the ability to learn transitions with minimal task-specific data. However, it requires considerable computational resources and time for training. BERT fine-tuning is delicate, requires careful parameter tuning, and is limited by the maximum word input (512 tokens), which limits its effectiveness for long texts.

2) About HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is an advanced clustering algorithm that extends the concepts of the classic DBSCAN by introducing a hierarchy and addressing some of its limitations. This algorithm is particularly adept at handling data with varying densities, a common challenge in real-world datasets. The fundamental concept of HDBSCAN, similar to DBSCAN, is based on the notion of density-based clustering. However, HDBSCAN does not require the specification of a global density threshold (ε in DBSCAN). Instead, it operates on the idea of varying density, which allows it to identify clusters of differing densities. This makes HDBSCAN more flexible and applicable to a broader range of datasets compared to DBSCAN.

HDBSCAN begins by transforming the dataset into a hierarchical tree of clusters. It does this by first creating a minimum spanning tree of the data, then condensing this tree based on a density threshold, effectively creating a hierarchy of clusters. The next step involves a complex process of extracting the stable clusters from this hierarchical tree. This process includes pruning the tree and selecting clusters based on their persistence, a measure of cluster stability across the hierarchy.

One of the main advantages of HDBSCAN is its ability to identify clusters of varying shapes and sizes, much like DBSCAN, but with enhanced capability to handle varying densities. It’s particularly useful in scenarios where clusters may have different levels of sparsity. Additionally, HDBSCAN simplifies the parameter selection process. The primary parameter is the minimum cluster size (min_cluster_size), which intuitively defines the smallest size a cluster needs to be to be considered a separate cluster. Another significant benefit is its robustness to noise and outliers. Like DBSCAN, HDBSCAN can effectively separate noise from significant clusters, but it does so with a more refined approach due to its hierarchical nature.

HDBSCAN is a powerful and flexible clustering algorithm well-suited for complex datasets with varying densities. Its hierarchical approach, ability to handle different densities, and minimal parameter tuning requirements make it a popular choice in a wide range of applications, from data mining to image analysis and bioinformatics.

C. About Darknet

1) Definition of darknet and darknet observation systems

In the realm of the internet, there exist IP addresses not connected to any device (unused IP addresses). These collections of IP addresses are referred to by various names such as Network Telescope, Internet Sink Hole, or Darknet. In this study, we define these unused IP addresses collectively as the Darknet.

Darknet Observation Systems are systems designed to collect packets transmitted to this unused IP addresses. Furthermore, based on the actions taken upon the received packets, research [20] categorizes Darknet Observation

Systems into several types. This study will introduce the simplest and most widely implemented Observation system.

2) Stealth-type darknet observation systems and payload collecting method

- Stealth-type observation system:

Stealth-type observation systems are the most utilized due to their ease of installation and low cost. Fig. 1 illustrates a schematic of a stealth-type observation system. Such systems do not respond to incoming requests but merely record the incoming packets.

- Stealth-type observation system collecting payloads:

The schematic of a stealth-type observation system that collects payloads is shown in Fig. 2. This system operates similarly to a standard observation system by recording incoming packets. However, it responds to TCP connection request SYN packets with SYN+ACK packets, prompting the sender to transmit the initial payload. Subsequently, it sends an RST packet to terminate the session immediately. This approach enables the collection of initial payloads, which is not typically possible with general stealth-type observation systems.

- Distinction between General Payload and Darknet Payload:

In this study, darknet payloads are defined not as the typical fragmented and continuously transmitted payloads observed in standard internet device communications, but rather as the initial payload sent by the sender in the above-described stealth-type observation system.

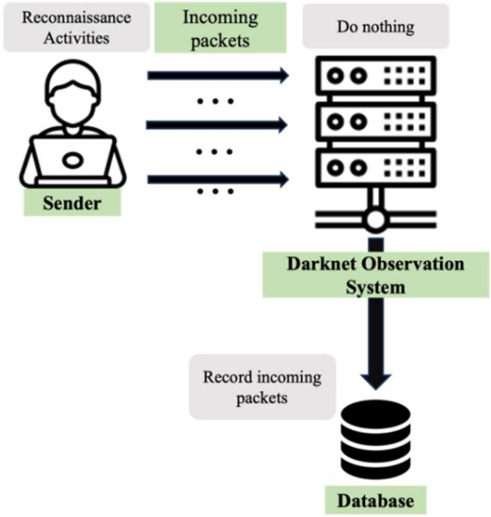

Fig. 1. Darknet stealth-type observation system.

D. Research on Payload Analysis

Numerous investigations have been undertaken regarding the challenges of both general payload and darknet payload clustering, broadly categorized into two principal areas: detection of abnormal and clustering payloads.

1) Researches on detection of abnormal payloads

Yamanaka [21] employs machine learning techniques to distinguish between normal and anomalous payloads in Modbus/TCP and BACNet protocols. This research treats each byte in the payload byte sequence as an individual word, utilizing BERT and Word2Vec for feature extraction from normal payloads. Utilizing these features, Yamanaka applies Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) in the machine learning process. The study leverages the results from this machine learning model to differentiate between normal and anomalous payloads. The findings indicate a high accuracy in distinguishing between these payload types, with features extracted from the BERT model yielding a higher detection rate compared to Word2Vec.

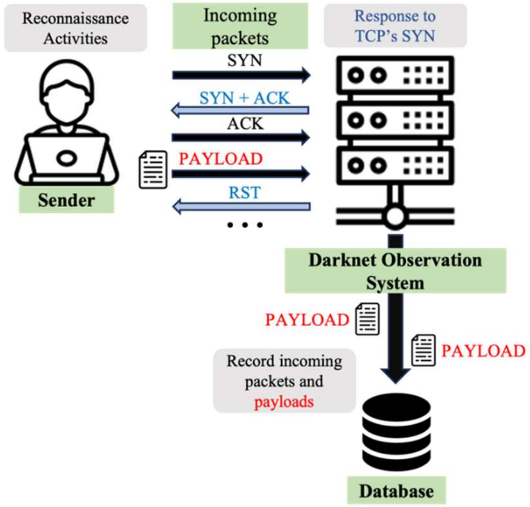

Fig. 2. Darknet stealth-type observation system collecting payloads.

Nakajima [22] transforms the content of payloads recorded during the communication between botnets and Command and Control (C&C) servers into ASCII format in two datasets: CTU-DATASET and BOS2018. It then employs N-gram for feature extraction. These features are fed into a Recurrent Neural Network (RNN) machine learning model. The machine learning outcomes are applied to differentiate between normal and anomalous payloads in cyber spaces. The study demonstrates the effectiveness of using 2-gram for data extraction in the proposed method from ASCII sequences converted from payloads, though its effectiveness in other datasets remains unverified.

Takahashi [23] focuses on validating the effectiveness of BERT in extracting payload features for use in detecting anomalous packets. Payload contents are converted into hexadecimal strings (ranging from 00 to FF), and these 256 tokens are fed into a deep machine learning model, a Variational Autoencoder (VAE), with tasks including Masked Language Modeling and Next Sentence Prediction. The results show that features extracted using BERT are effective in detecting anomalous packets in Modbus, BAC Net, and Ethernet IP protocols, but show very low effectiveness with HTTPS.

2)Researches on clustering of darknet payloads

Suzuki [24] utilizes Self-Organizing Maps (SOM) to cluster darknet payloads based on features extracted from fuzzy hashes. This method is based on the principle of generating hash codes from the blocks into which payloads are divided, and calculates payload similarity based on the derived hash codes. This method is effective when the payload length is large, but is less efficient when the payload length is short, especially when the payload is shorter than the length of one block.

Suzuki [25] also extracts all readable strings from the resulting darknet payload. Then, using the amount and frequency of these readable strings, it found that the string at the beginning of the payload has a significant impact on the payload characteristics.

Kajikawa [26] involved the detection of distributed scan groups and the clustering of attack payloads. By analyzing payloads sent to port 1723 (PPTP) and manually clustering them by protocol, it became clear that payloads sent to a particular port are not limited to the protocols traditionally associated with that port, but include payloads from a variety of protocols. The clustering by protocol is based on the manual work of the user.

Vincent Ghiette [27] focuses on clustering DNS packets, a UDP protocol, to identify scanning activity performed by a large number of IP addresses. The darknet payload is converted to a hexadecimal string and a distance matrix is computed using the Levenshtein distance as input for cluster generation. This matrix is processed with the HDBSCAN model to cluster the payloads. The clustering results show that at least 93.3% of the payloads were successfully clustered; based on the DNS packet clustering results, several templates were developed. Using these templates, 96.04% of payloads were correctly clustered, yielding an accuracy of 97.28%. Mapping payloads to these templates revealed temporal variations in the use of DNS payloads in network scans. However, it was noted that UDP packets are unreliable because the source IP may be spoofed. Furthermore, since DNS protocol packets contain primarily ASCII characters and domain names, clustering methods rely heavily on the domain names in each payload. The clustering results correspond closely to the frequency of domain occurrences in the DNS payload. Furthermore, the validity of this method has not yet been verified for the TCP protocol.

3) *Limitations of related works*

This research represents an important milestone in payload clustering. Research using BERT, Word2Vec, and N-grams by Yamanaka [21], Nakajima [22], and Takahashi [23], as well as clustering methodologies by Suzuki [24, 25], Kajikawa [26], and Vincent Ghiette [27], have made significant progress in this area. At the same time, however, these studies have also revealed certain limitations and areas in need of improvement, especially regarding feature extraction from payload contents.

While Vincent Ghiette [27] is pioneering in its application to UDP packets, empirical verification in the context of the TCP protocol is notably lacking. This omission illustrates an important limitation: by ignoring TCP, the fundamental protocol for network communications, one may miss the behavior and characteristics of payloads in a large fraction of network traffic.

Research by Yamanaka [21], Nakajima [22], Takahashi [23], and colleagues using BERT and Word2Vec highlights the potential of machine learning techniques to decipher the complex characteristics of payloads. However, while robust, these methodologies often have difficulty fully understanding the complex and diverse nature of network payloads in a constantly evolving cyber threat landscape. Furthermore, Takahashi's research has observed that the effectiveness of these models is variable across different network protocols.

Similarly, BERT and Word2Vec are effective for certain payload types, but may not capture the subtle characteristics required for more complex or encrypted traffic. Also, N-gram based methods, while effective in pattern recognition, often have difficulty adapting to the scalability and dynamic nature of network traffic. In the area of clustering methods, the main challenge is to accurately cluster diverse payloads when dealing with extremely variable data streams, including encrypted data streams.

To address these identified deficiencies and challenges, this research proposes an effective method for extracting payload features, leveraging the BIG BIRD model for efficient and effective payload feature extraction specifically for payload clustering.

III. Proposed Method

This chapter presents a method for effectively extracting features of payloads sent to the darknet observation system. First, the features and principles of BIG BIRD are described, and then a method for extracting payload features is proposed.

A. About BIG BIRD [28]

BIG BIRD is a sophisticated variant of transformer-based models such as BERT and GPT, specifically designed for Natural Language Processing (NLP). BIG BIRD was developed to address the limitations of standard transformer models when processing long data sequences. Traditional transform models are limited by quadratic dependencies on sequence length, and their large memory and computational requirements limit their ability to process long documents. BIG BIRD solves this by introducing a sparse attention mechanism, allowing longer sequences to be processed efficiently made.

The core of BIG BIRD's innovation lies in its modified attention mechanism, which allows for linear scaling with sequence length. BIG BIRD combines three types of attention: global, window, and random. Global attention ensures that important parts of the sequence, such as special tokens, receive comprehensive attention. Window attention provides a local view, allowing each token to focus attention on a fixed-size window of neighboring tokens. Random attention introduces randomness and allows connections between distant tokens. This hybrid attention model allows BIG BIRD to efficiently manage longer sequences while maintaining contextual understanding.

BIG BIRD offers significant advantages, especially in its ability to handle long text sequences, surpassing the capabilities of standard transformers. BIG BIRD maintains a high level of context recognition and understanding of relationships in the data, even as the sequences grow in length. Moreover, its versatile applications extend beyond Natural Language Processing (NLP) and have proven useful in a variety of fields where handling long sequence data is essential.

B. Proposed Method

The proposed clustering method is shown in Fig. 3 Specificaly, it consists of three steps:

- Step 1—Collect Payloads: Collect packets from the darknet (or from darknet's packet capture file) and extract the payload from the packets.
- Step 2—Conversion to hexadecimal string: Convert the payloads into a hexadecimal string.

- Step 3—Feature Extraction: Assume that each byte is a word, and input all words into the feature extraction model BIG BIRD to extract features of the payload. It is imperative to emphasize that the BIG-BIRD model utilized in this study is not subjected to re-training or fine-tuning processes; Here, we employ the model originally proposed by Zaheer *et al.* [28].

The features obtained from the BIG BIRD can be input into a clustering model to cluster the payloads.

To prove the effectiveness of the proposed method for extracting payload features, several comparison and validation experiments are conducted in Chapter IV.

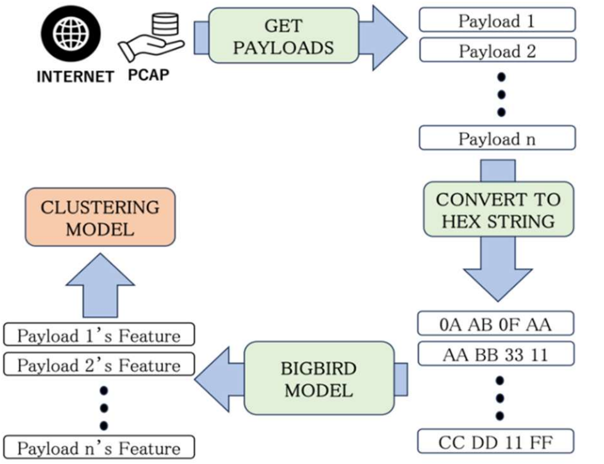

Fig. 3. Proposed method—payload's feature extraction with BIG BIRD.

IV. Experiments

In this chapter, to assess the efficacy of the proposed method in feature extraction, an experiment was undertaken to compare the clustering outcomes of features extracted by the proposed method against those derived from N-Gram, Word2Vec, BERT, and the approach referred to as Levenshtein & HDBSCAN of Vincent Ghiette [27].

A. About the Comparative Experiment Data

For this experiment, 1590 TCP payloads from the darknet, encompassing encrypted payloads and spanning various content types, were gathered. These payloads were sourced from 9 distinct payload groups across multiple protocols observed through the National Defense Academy of Japan's Darknet observation system. This rigorous collection process aimed to ensure accurate and impartial validation and assessment. Specifically, the payloads were clustered as shown in Table 1, detailing the content and quantity of each protocol's payloads.

Table1. Contents and quantity of payloads for each protocol

Highest Protocol	Quantity	A Sample of payload's contents (HEX)
BitTorrent	100	13 42 69 74 54 6f 72 72 65 6e…
TDS	200	12 01 00 29 00 00 00 00 00 00…
HTTP	400	50 4f 53 54 20 2f 20 48 54 54…
Bitcoin	190	f9 be b4 d9 76 65 72 73 69 6f…
SMB	100	00 00 00 85 ff 53 4d 42 72 00 …
Binary on Port 80	100	83 f7 da f1 1d 21 dc 30 ba 20…
SSH	100	53 53 48 2d 32 2e 30 2d 50 55…
SSH2	200	00 00 00 14 06 01 00 00 00 0b…
TLS	200	16 03 00 00 69 01 00 00 65 03…

Since the clusters of the experimental data were known, the extracted features were input into the KMeans clustering model (with K = 9) to check how well each feature extraction model could cluster the original group.

B. Experimentation with KMeans (K = 9)

1) Experimental procedure and evaluation metrics:

Fig. 4 illustrates the experimental procedure. Specifically:

- Step 1—Conversion to hexadecimal string: For all payloads, convert the contents to hexadecimal string format. In text format, each byte (in hexadecimal format) is assumed to be one word, with a space between words.
- Step 2—Feature Extraction: Input all text strings obtained in Step 1 into the feature extraction models (N-gram with n = 2, Word2Vec, BERT, BIG BIRD and Levenshtein Distance) and store the output feature F.
- Step 3—Clustering: Using the KMeans model (K = 9), input the features F and store the clustereds as groups G_n(n = {0, 1, 2, 3, 4, 5, 6, 7, 8}).
- Step 4—Evaluate the efficiency of the extraction algorithm: The evaluation metric adopted in this study is clustering accuracy (p). To determine the clustering accuracy, we first calculate the clustering accuracy for the payloads of each protocol.

The characteristics of a clustered group are dependent on the protocol that constitutes the majority (principal component) of payloads within that group, while the payloads of other protocols are considered as noise. To identify the principal component protocols, first count the payloads for each protocol in each group obtained in Step 3. The protocol with the highest number of payloads is assumed to be the principal component protocol of that group.

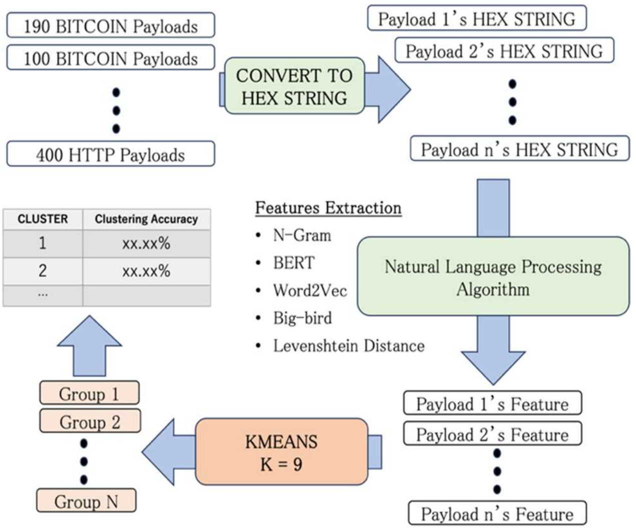

Fig. 4. Experimental procedure with KMeans (K = 9).

The clustering accuracy $P(type_T)$ for a given protocol type T is calculated using the following formula:

$$P(type_T) = \frac{C_i + C_j + \cdots}{A(type_T)}$$

In this formula i, j are the numbers of groups in which protocol type T is main component. C_i, C_j are the count of payload of protocol type T (payload type T) in groups i and j. $A(type_i)$ is the total count of payload type T.

Based on the individual protocol clustering accuracies, the overall clustering accuracy (p) of the clustering model is calculated as follows:

$$p = \frac{\sum_{i=1}^{9} P(type_i) \times A(type_i)}{S}$$

Here, $P(type_i)$is the clustering accuracy of the payload type i. $A(type_i)$ is the total count of payload type i and S is the total count of payloads in the experimental data.

2) Result of experimentation with KMeans (K = 9)

Figs. 5 and 6 show the N-gram clustering results in two and three dimensions. Figs. 7 and 8 show the Word2Vec clustering results in two and three dimensions. Figs. 9 and 10 show the BERT clustering results in two and three dimensions. Figs. 11 and 12 show the BIG BIRD clustering results in two and three dimensions. Figs. 13 and 14 show the clustering results of Levenshtein Distance in two and three dimensions.

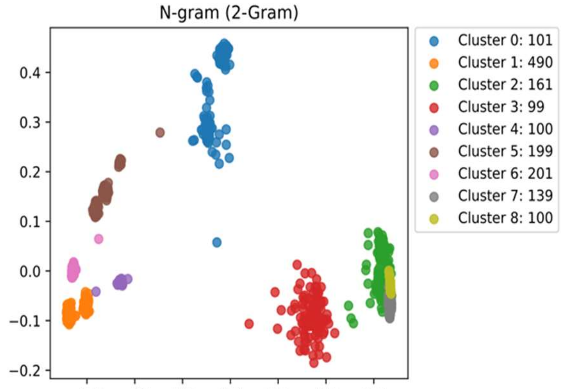

Fig. 5. Clustering result of N-gram (n=2) (2D).

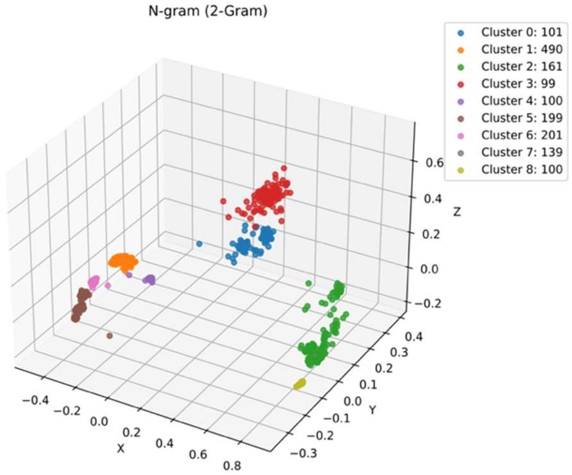

Fig. 6. Clustering result of N-gram (n=2) (3D).

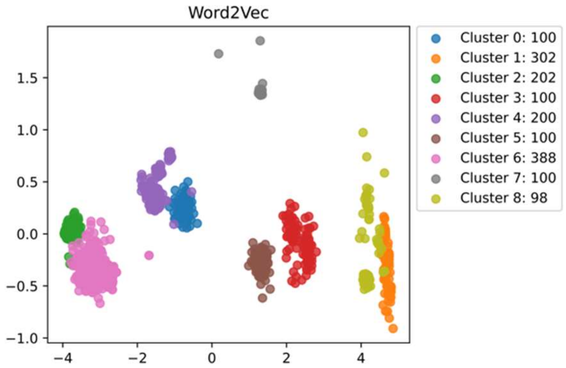

Fig. 7. Clustering result of Word2Vec (2D).

Tables 2 and 3 show the payload clustering results and clustering accuracy from the KMeans algorithm using features extracted from the N-gram, Word2Vec, BERT, Levenshtein Distance, and BIG BIRD algorithms. A comparison of the clustering accuracy (p) results is shown in Fig. 15. The obtained results show that among the four NLP algorithms (N-gram, Word2Vec, BERT, and BIG BIRD), the BIG BIRD algorithm yields the highest clustering accuracy of 100%, followed by Word2Vec with 87.92%, N-gram with 81.64%, BERT with 67.55% and Vincent Ghiette [27] which shows the lowest result of 50.75% when specified to cluster into 9 clusters.

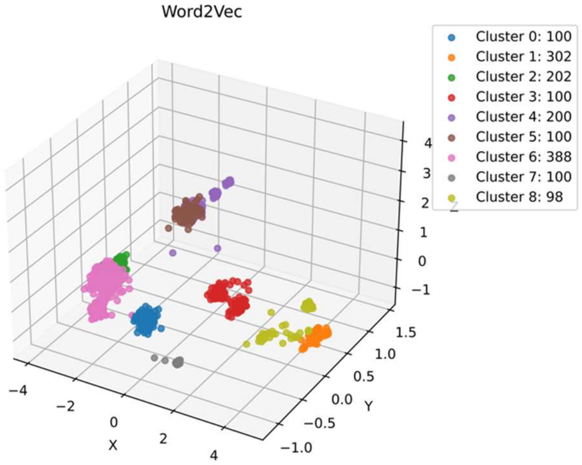

Fig. 8. Clustering result of Word2Vec (3D).

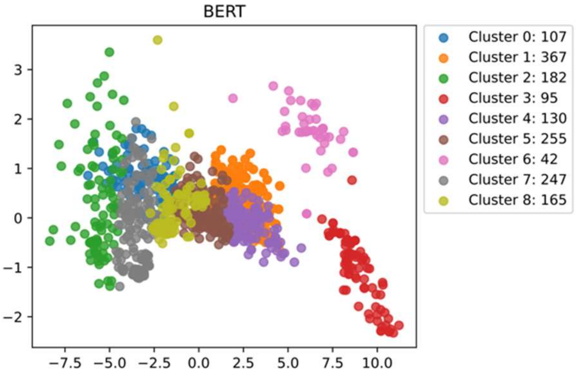

Fig. 9. Clustering result of BERT (2D).

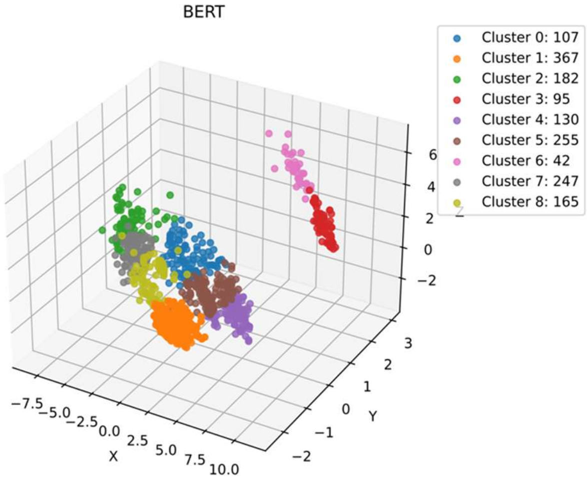

Fig. 10. Clustering result of BERT (3D).

Table 2. Experimentation results with KMeans (K = 9) of N-gram, Word2Vec, BERT, BigBird and Levenshtein Distance

Group	N-gram	Word2Vec	BERT	Big Bird	Levenshtein Distance
0	BitTorrent (100) Binary on Port 80 (1)	SSH (100)	HTTP (175) BitTorrent (50) Binary on Port 80 (24)	BITCOIN (190)	TDS (200) SSH2 (200) BitTorrent (100) SSH (100)
1	SSH2 (200) Bitcoin (190) SSH (100)	HTTP (302)	SSH2 (114) TDS (16)	HTTP (400)	HTTP (123) Binary on Port 80 (10)
2	HTTP (161)	TDS (200) SSH2 (2)	Binary on Port 80 (39) HTTP (3)	SSH2 (200)	HTTP (20) Binary on Port 80 (15)
3	Binary on Port 80 (99)	BitTorrent (100)	Bitcoin (186) TLS (177) Binary on Port 80 (4)	SMB (100)	Bitcoin (190) SMB (98) TLS (63) Binary on Port 80 (9)
4	SMB (100)	TLS (200)	HTTP (129) BitTorrent (34) SSH (7) Binary on Port 80 (12)	BITTORENT (100)	HTTP (76) Binary on Port 80 (25)
5	TLS (199)	Binary on Port 80 (100)	HTTP (94) Binary on Port 80 (1)	TDS (200)	HTTP (1)
6	TDS (200) TLS (1)	SSH2 (198) Bitcoin (190)	SMB (100) BitTorrent (16) Bitcoin (4) SSH (2) TLS (22) Binary on port 80 (20)	Binary on Port 80 (100)	TLS (79) HTTP (62) Binary on Port 80 (22)
7	HTTP (139)	SMB (100)	TDS (171) SSH (24) SSH2 (59) TLS (1)	TLS (200)	HTTP (113) SMB (2) Binary on Port 80 (19) TLS (58)
8	HTTP (100)	HTTP (98)	SSH (67) TDS (13) SSH2 (27)	SSH (100)	HTTP (5)

Table 3. Clustering Accuracy of N-gram, Word2Vec, BERT, BigBird and Levenshtein Distance with KMeans (K= 9)

Protocol	N-gram	Word2Vec	BERT	Big Bird	Levenshtein	Vincent Ghiette [27]
P(BitTorrent)	100	100	0	100	0	94
P(TDS)	100	100	85.5	100	100	100
P(HTTP)	100	100	99.25	100	84.5	86.75
P(Bitcoin)	0	0	97.98	100	100	100
P(SMB)	100	100	100	100	0	99
P(Binary on Port 80)	99.5	100	39	100	0	0
P(SSH)	0	100	67	100	0	100
P(SSH2)	100	99	57	100	0	100
P(TLS)	99.5	100	0	100	39.5	99
Clustering Accuracy of Model (p)	**81.64%**	**87.92%**	**67.55%**	**100%**	**50.75%**	**89.81%**

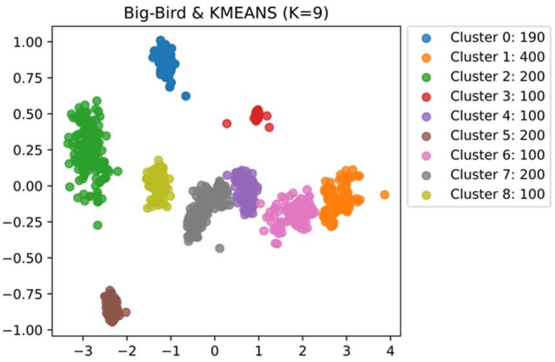

Fig. 11. Clustering result of Proposed Model (2D).

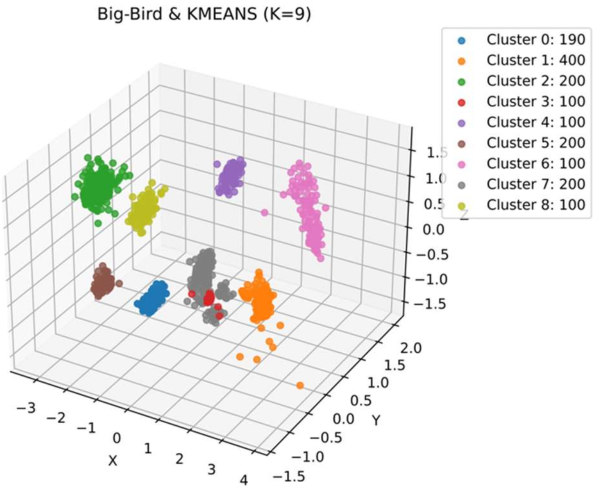

Fig. 12. Clustering result of Proposed Model (3D).

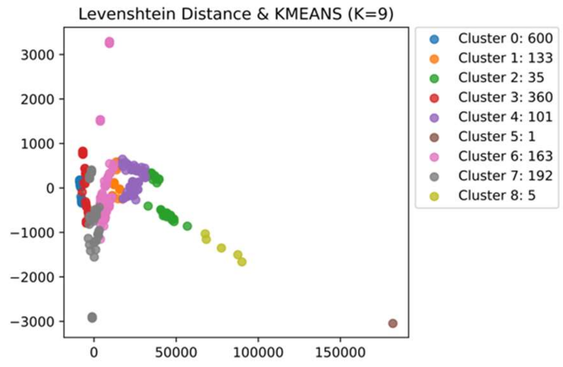

Fig. 13. Clustering result of Levenshtein Distance (2D).

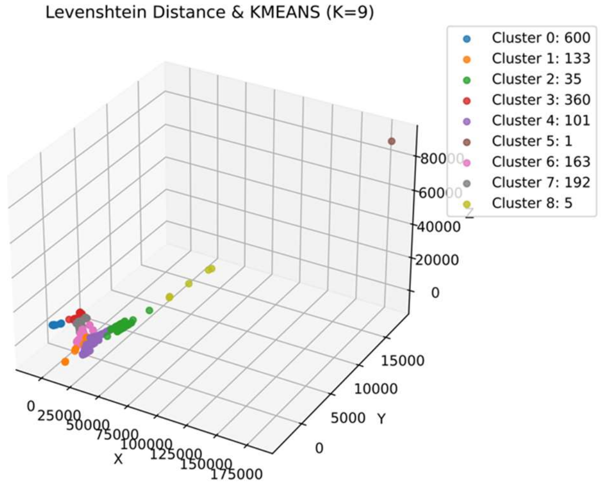

Fig. 14. Clustering result of Levenshtein Distance (3D).

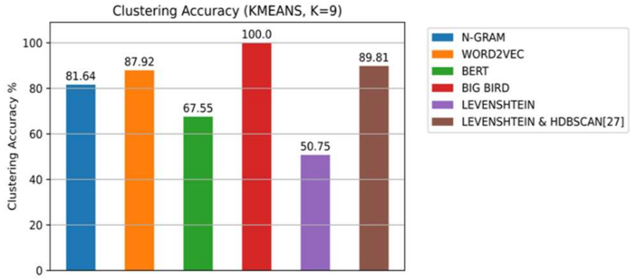

Fig. 15. Comparison results of Clustering Accuracy between the proposed method and other feature extraction models.

C. Experimental Comparision between the Proposed Method and Vincent Ghiette

The clustering method Levenshtein & HDBSCAN of Vincent Ghiette [27] achieves high clustering accuracy in payload clustering. In addition, the proposed method employs HDBSCAN, a clustering method that does not require the specification of the number of clusters. In order to objectively evaluate the proposed method and the method by Vincent Ghiette [27], it is necessary to apply the features extracted by the proposed method to other classifier that do not require specification of the number of clusters, as in Vincent Ghiette [27], and compare and evaluate the clustering results. In the next experiments, the proposed method is adapted to HDBSCAN (used by Vincent Ghiette [27]) and compared with the clustering results of Vincent Ghiette [27].

1) Comparative experiment: comparision between Model (Proposed Method & HDBSCAN) and Model (LEVENSHTEIN & HDBSCAN) by Vincent Ghiette [27]

In this experiment, features extracted from BIG BIRD are used in the HDBSCAN clustering model. Here, the parameter 'min_cluster_size' is varied from 2, 5, 10, 20, 30.... to 100, and these results were compared with the clustering results of the model by Vincent Ghiette [27]. The clustering results will be compared with 2 metrics:

1. Clustering Accuracy (p): This metric, consistent with the calculation method used in previous experiment, measures the precision of the clustering process. It assesses how accurately the clustering algorithm has grouped similar payloads together. High clustering accuracy indicates that payloads within each cluster are very similar to each other, and payloads in different clusters are distinct.

2. Clustering Ratio (r): Defined as the ratio of the number of payloads successfully clustered to the total number of payloads, this metric evaluates the efficacy of the clustering algorithm in categorizing the data. A high clustering ratio suggests that the algorithm is capable of effectively clustering a large proportion of the payloads, while a lower ratio may indicate that many payloads remain unclustered or are incorrectly grouped.

The results of adapting the method by Vincent Ghiette [27] to the experimental data are as follows:

- Clustering Accuracy (p): 89.81%
- Clustering Ratio (r): 89.94%

Subsequently, these results will be compared with the results from Model (Proposed Method & HDBSCAN).

2) Result of comparative experiment

Fig. 16 presents the comparative results for the Clustering Accuracy metric between the BIG BIRD & HDBSCAN model and the model by Vincent Ghiette [27]. These results show that when min_cluster_size is less than 60, the BIG BIRD model consistently achieves higher clustering accuracy than Vincent Ghiette [27]. When min_cluster_size exceeds 60, the Clustering Accuracy (p) drops sharply. The reason for this is that there are many groups of protocol payloads in the dataset that consist of 100 payloads, and these groups are considered as noise when the value of the parameter min_cluster_size is around 100.

Fig. 17 presents a comparison regarding the Clustering Ratio. Based on these findings, it is evident that the BIG BIRD model initially exhibits a very high ratio, which gradually decreases as min_cluster_size increases. Notably, when min_cluster_size exceeds 60, the Clustering Ratio significantly diminishes and falls below that of the model by Vincent Ghiette [27]. The underlying reason for this phenomenon aligns with the factors contributing to the reduction in Clustering Accuracy.

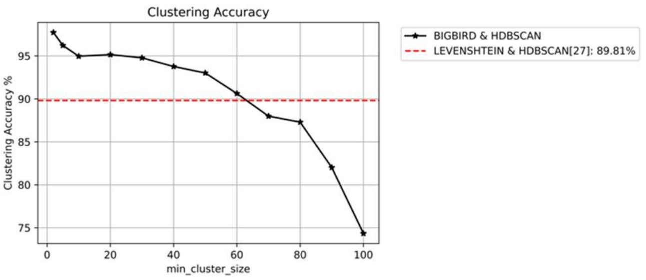

Fig. 16. Comparison results of Clustering Accuracy between the proposed method with HDBSCAN and the method by Vincent Ghiette [27].

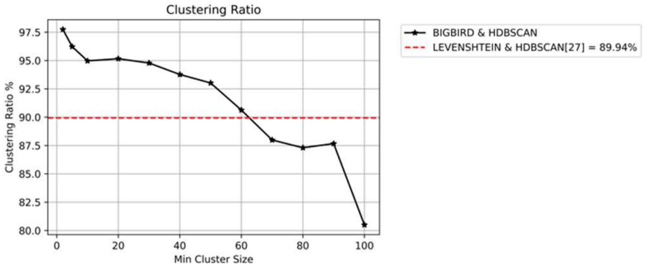

Fig. 17. Comparison results of Clustering Ratio between the proposed method with HDBSCAN and the method by Vincent Ghiette [27].

Based on the comparative results obtained:

- When min_cluster_size is ≤60:
  - ➢ p (Proposed method & HDBSCAN) > p (Vincent Ghiette [27])
- When min_cluster_size is ≤60:
  - ➢ r (Proposed method & HDBSCAN) > r (Vincent Ghiette [27])

From these results, we conclude that the Proposed method & HDBSCAN model consistently yields better outcomes across all 2 metrics:

1. Clustering Accuracy—p
2. Clustering Ratio—r

When min_cluster_size is ≤60.

The results show that by setting optimal parameter min_cluster_size, the Model (Proposed method & HDBSCAN) can cluster payloads with similar content features with higher accuracy (p) than Vincent Ghiette [27]. Furthermore, the percentage of clustered payloads (r) is also higher.

V. Applying the Proposed Model to Darknet Observation Data

In this chapter, we demonstrate the effectiveness of the proposed method by applying the (Proposed Method & HDBSCAN) Model used in Experiment 2 to data from National Defense of Academy's Darknet observation system. The validation of the clustering results is evaluated by calculating the similarity between payloads in the same group with a Cosine distance.

A. Experiment Environment

Given the considerable amount of data accumulated in recent years and the huge amount of computing time and memory required, this experiment will use one day's worth of data observed at port 80 of the darknet system on September 1, 2017. The specific quantities of data files and details of the data employed in this experiment are shown in Table 4.

Table 4. Information of data used in the experiment

	2017-09-01	2017-09-01 Port 80
File Size	14 Gb	225 Mb
Data Size	12 Gb	197 Mb
Packets	130,886,271	1,763,301
Uniq Payloads	732,940	27,100

B. Experiment Result

For this experiment, min_cluster_size = 5 was selected as the parameter for the HDBSCAN algorithm. The clustering of the 27,100 payloads sent to port 80 resulted in 64 groups, with 116 payloads in Group(-1) considered noise. This result indicates that approximately 99.57% of the total payloads were clustered.

C. Verification of Clustering Results

To assess the clustering results, a validation process is conducted. However, due to the absence of labels for individual payloads in darknet data, the validation of the experimental clustering results necessitates an extensive investigation into the similarity between payloads within the same group, to verify the accuracy of the clustering.

The metric employed for this evaluation is the average cosine distance between payloads within each group. A cosine distance approaching 1 within a group indicates a higher degree of payload similarity, thus reflecting the efficacy of the clustering model. Conversely, a cosine distance approaching 0 suggests a lack of similarity, indicating a potential misclassification of payloads. To provide an overarching assessment of the overall results, an example is presented to elucidate the method of evaluation.

1) An example of comparing similarity between payloads in Group X

Fig. 16 serves as an illustrative example of a heatmap diagram, depicting the similarity levels among the 7 payloads of Group X. The intensity of each cell at position (i, j) on the heatmap corresponds to the degree of similarity between Payload i and Payload j. The more similar Payload i and Payload j are, the lighter the color of the cell at (i, j) will be. Conversely, a greater difference between Payload i and Payload j results in a darker color at their respective position in the heatmap. At position (i, i), we do not perform any comparison; hence, all values are zero, resulting in black cells along the diagonal of the heatmap. In this experiment, we evaluated the similarity among payloads within the same group by calculating the correlation values for all positions in the heatmap (excluding position (i, i)). The string Group X (7) = 0.72485 displayed above the hit map diagram implies that Group X has 7 payloads, and the average cosine distance of the Group is 0.72485.

Furthermore, due to the large number of payloads in a single group and for ease of tracking, in subsequent heatmap diagrams, we will omit the correlation values and indexes at each position (i, j). Specifically, in future diagrams, Fig. 18 will be replaced with Fig. 19, which simplifies the representation while retaining the essential information regarding payload similarities within the group.

2) Verification results

The validation results regarding the similarity of payloads within the same group are presented in Fig. 20. Remarkably, all the groups demonstrated high average similarity indices. Among the 64 groups, two groups exhibited average cosine distances above 0.93, and 60 groups had indices exceeding 0.98. Collectively, these 62 groups represented approximately 92% of the total payloads analyzed.

The remaining two groups registered lower similarity indices, with one (group 2) having an index of 0.76 and the other an index of 0.2529.

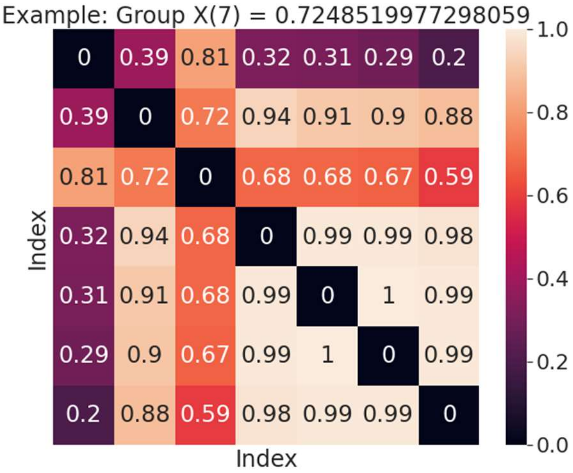

Fig. 18. Heatmap representing cosine distances within Group X.

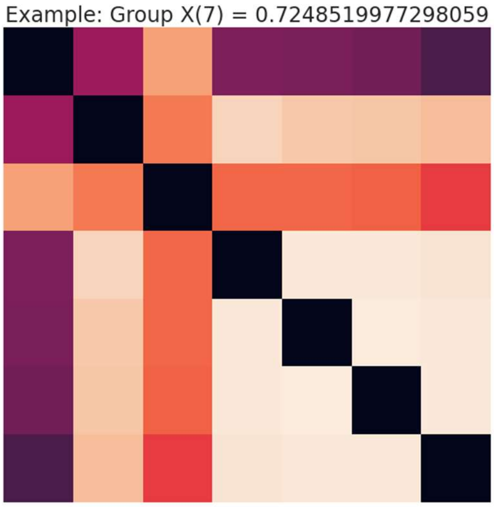

Fig. 19. Heatmap representing cosine distances within Group X (omited the corelation values and indexes).

3)Consideration of Groups with Low Similarity and Noise Group

First, upon examining the contents of payloads in Group 5 which comprised 2079 payloads and exhibited the lowest average cosine distance of 0.2529, it was discovered that the payloads in this group were not associated with any specific protocol but were rather irregular binary data. The characteristics of this group were markedly different from other groups, leading to the conclusion that unrelated irregular binary data were clustered into a single group. This group was categorized as a cluster of irregular binary data not corresponding to HTTP requests on port 80. In a sense, this can be considered a positive outcome.

When analyzing the payloads in Group 0, which had an average cosine value of 0.76 and consisted of 6 payloads, it was found that despite the payloads being distinct, their short length and features like line breaks, spaces, slashes, and the consistent presence of “HTTP1.0” at the end, likely contributed to their incorrect clustering into a single group.

Upon investigating the payloads within Group −1, identified as the noise group, it was observed that many of the payloads were either unique HTTP requests in small quantities, failing to reach the minimum cluster size threshold, or were associated with various other protocol types. Reducing the min_cluster_size could result in these payloads forming a new group; however, this might lead to their reclassification into other groups and a rapid increase in the number of clusters.

VI. Conclusion and Future Works

A. Conclusion

In this study, we introduced a novel feature extraction method utilizing BIG BIRD to analyze various payload contents, including encrypted payloads, and addressing diverse challenges encountered in payload clustering tasks. Our proposed approach showcased superior efficacy compared to other text feature extraction models like N-gram, Word2Vec, and BERT in extracting features from payloads. Specifically, through comparative experiments utilizing the KMeans clustering model, our method yielded a remarkable 100% clustering accuracy when using features extracted from various payloads of protocols. Notably, the post-clustering payload groups precisely aligned with the pre-clustering groups, a precision that gradually diminished with Word2Vec, N-gram, and BERT. In comparison to the Levenshtein Distance Method, our method demonstrated a notable capacity in clustering payloads with similar content features. Furthermore, when integrated with the HDBSCAN classifier, it achieved high clustering accuracy while maintaining a substantial proportion of clustered payloads.

Furthermore, when our proposed method was applied to actual darknet data clustering, it successfully clustered approximately 99.57% of the payloads. Moreover, we verified that over 92% of these payloads had a high degree of similarity within their clustered groups.

B. Limitation and Future Work

There are still several limitations to the proposed method. Firstly, in this study our focus is solely on the analysis of initial payloads. Looking ahead, within the context of detecting cyber attacks and malicious communications, it is conceivable that there are scenarios where immediately discernible malicious communications exist within initial payloads, while in other cases, subsequent payloads may follow without clear indicators of malicious intent. This suggests the difficulty in relying solely on initial payloads for determination. However, by integrating the clustering result of this study with additional information such as transmission times, sending frequencies, and sending methods, there is potential to enhance the detection of malicious communications and attacks.

Secondly, a significant limitation observed during the experimental phase of this study pertains to the feature extraction phase utilizing the BIG BIRD algorithm, which consumed considerable time. In the future, solutions to address this issue will be devised, potentially leveraging parallel processing to expedite feature extraction.

Finally, the clustering model HDBSCAN relies on the hyperparameter min_cluster_size. It necessitates exploring various values to determine the optimal value for a given dataset. To mitigate this requirement, we aim to explore models that can automatically compute the optimal value without relying on hyperparameters.

These limitations underscore areas for future improvement and refinement of the proposed method. Efforts to address these challenges will contribute to enhancing the efficiency and applicability of the methodology in practical scenarios.

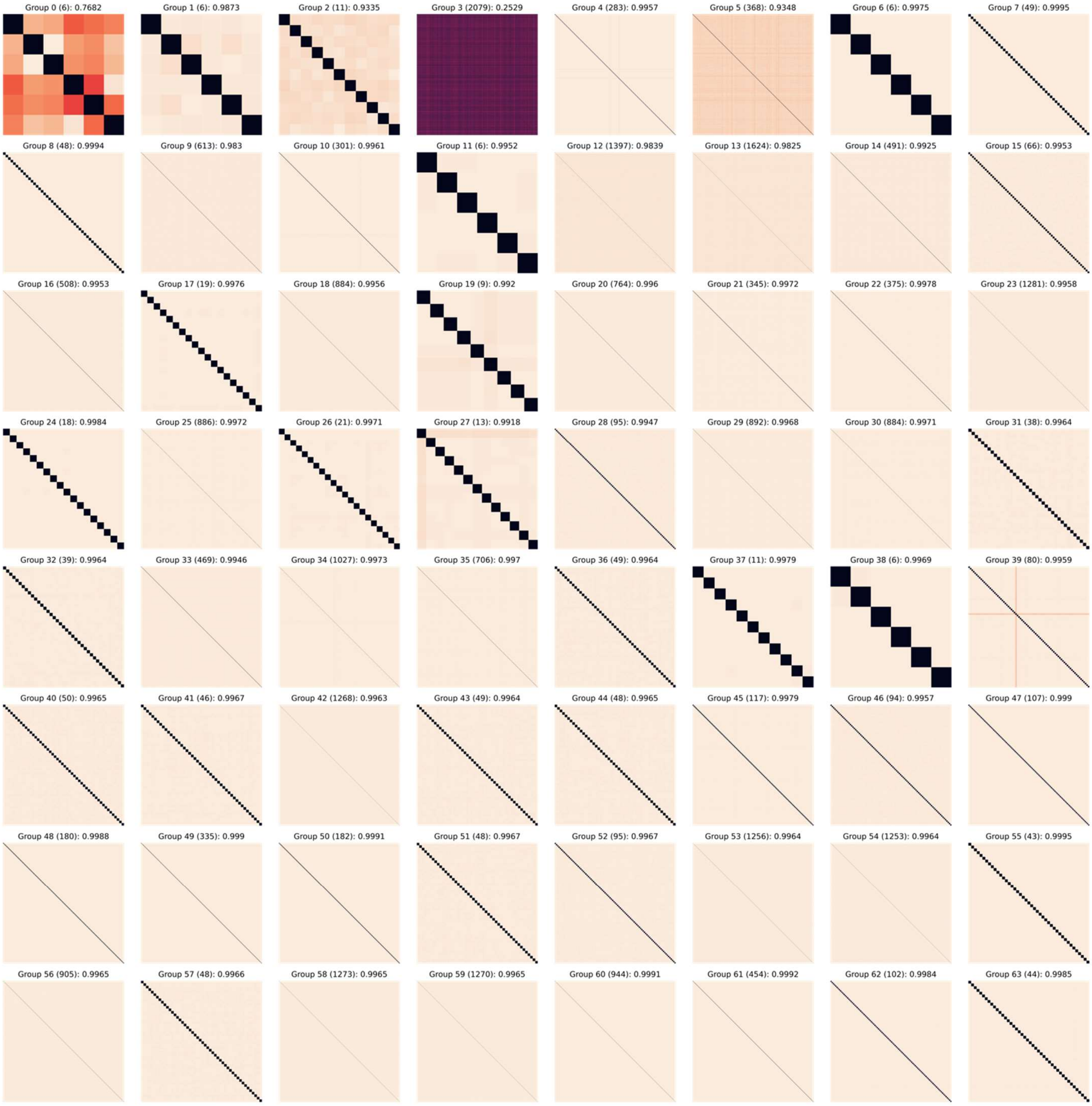

Fig. 20. Heatmap representing the validation results regarding the similarity of payloads within the same group.

Conflict Of Interest

The authors declare no conflict of interest.

Author Contributions

Son Pham Anh processed the data, prepared the experimental environment, conducted the experiments, and wrote the paper. Yasuhiro Nakamura collected the data, prepared the experimental equipment, and provided comments and advice during the paper's preparation. The authors reviewed and revised the final version of the paper, all authors had approved the final version.

Acknowledgment

We would like to express our gratitude to the National Defense Academy of Japan for providing the Darknet Observation System.

References

[1] Esentire. Cybersecurity ventures report on cybercrime. [Online]. Available: https://www.esentire.com/cybersecurity-fundamentals-defined/glossary/cybersecurity-ventures-report-on-cybercrime#:~:text=One%20of%20the%20key%20highlights,reaching%20%24265%20billion%20by%202031

[2] S. Morgan. (Oct. 2022). Cybercrime to cost the world 8 trillion annually in 2023. [Online]. Available: https://cybersecurityventures.com/cybercri-me-to-cost-the-world-8-trillion-annually-in-2023/

[3] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary

campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol 1, no. 1, p. 80, 2011.

[4] NICTER Project. National Institute of Information and Communications Technology of Japan. [Online]. Available: https://www.nicter.jp/en/project

[5] The UCSD Network Telescope. [Online]. Available: https://www.caida.org/projects/network_telescope/

[6] The IUCC/IDC Internet Telescope. [Online]. Available: https://nocvm.iucc.ac.il/research/telescope/

[7] The Honeynet Project. [Online]. Available: https://www.honeynet.org

[8] Internet Storm Center. [Online]. Available: https://isc.sans.edu/data/

[9] M. Antonakakis, T. April, and M. Bailey, *et al.*, "Understanding the mirai botnet," *26th USENIX Security Symposium (USENIX Security 17)*, pp. 1093–1110, 2017.

[10] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of Hajime a peer-to-peer IoT botnet," in *Proc. Network and Distributed Systems Security (NDSS) Symposium*, 2019.

[11] Y. Yao, H. Guo, G.Yu, and F. Gao, "Discrete-time simulation method for worm propagation model with pulse quarantine strategy," *Procedia Eng.*, vol 15, pp. 4162–4167, 2011.

[12] H. Berghel, "The code red worm," *Communications of the ACM*, vol 44, no. 12, pp. 15–19, 2001.

[13] D. Moore, C. Shannon, and K. C. Claffy, "Code-Red: A case study on the spread and victims of an Internet worm," in *Proc. the 2nd ACM SIGCOMM Workshop on Internet Measurement*, 2002, pp. 273–284.

[14] M. Ku¨hrer, T. Hupperich, C. Rossow, and T. Holz, "Hell of a handshake: Abusing TCP for reflective amplification DDoS attacks," *USENIX Workshop on Offensive Technologies (WOOT)*, 2014.

[15] R. Sommese, K. C. Claffy, R. van Rijswijk-Deij, A. Chattopadhyay, A. Dainotti, A. Sperotto, and M. Jonker, "Investigating the impact of DDoS attacks on DNS infrastructure," *ACM Internet Measurement Conference (IMC)*, Oct. 2022.

[16] A. Ali, A. Chaudhary, S. Sahana, "A review of defense against Distributed DoS attack based on Artificial Intelligence approaches," in *Proc. 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, 2022.

[17] Cyber Security Labo. (Feb. 2023). National Institute of Information and Communications Technology: NICTER Observation report 2022. [Online]. Available: https://csl.nict.go.jp/report/NICTER_report_2022.pdf (In Japanese)

[18] CTI League. CTI-League Darknet Report 2021. [Online]. Available: https://cti-league.com/wp-content/uploads/2021/02/CTI-League-Darknet-Report-2021.pdf

[19] T. Kasama, "Long-term Darknet Analysis in NICTER," *Journal of the National Institute of Information and Communications Technology*, vol. 63, no. 2, pp. 25–31, 2017.

[20] C. Fachkha and M. Debbabi, "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1197–1227, 2016.

[21] Y. Yamanaka, M. Yamada, T. Takahashi, and T. Nagai, "Utilizing BERT for feature extraction of Packet payload," in *Proc. the 35th Annual Conference of the Japanese Society for Artificial Intelligence*, 2021. (In Japanese)

[22] S. Nakajima and N. Torii, "Anomaly detection method through text clustering of payload information using recurrent neural networks (RNN)," in *Proc. the 83rd National Convention of Information Processing Society of Japan*, 2021. (In Japanese)

[23] T. Takahashi, Y. Yamanaka, T. Minami, and Y. Nakajima, "Performance evaluation of anomaly communication detection using BERT for feature extraction of packet payload," in *Proc. the 37th Annual Conference of the Japanese Society for Artificial Intelligence*, 2023. (In Japanese)

[24] Y. Suzuki and Y. Nakamura, "A self-organized clustering method of Darknet arrival packet payloads," in *Proc. Computer Security Symposium 2015*, 2015. (In Japanese)

[25] Y. Suzuki, Y. Goto, and Y. Nakamura, "Simplified honeypot-based darknet observation and proposal for payload clustering method," in *Proc. the 77th National Convention of Information Processing Society of Japan*, 2015. (In Japanese)

[26] K. Kajikawa and Y. Nakamura, "Detection of distributed scan group and clustering of attack payloads," in *Proc. the 16th Forum on Information Technology, 2017.* (In Japanese)

[27] V. Ghiette and C. Dörr, "Clustering payloads: Grouping randomized scan probes into campaign templates," in *Proc. 2022 IFIP Networking Conference–IEEE*, 2022, pp. 1–9.

[28] M. Zaheer, G. Guruganesh, and A. Dubey *et al.* "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, 2020. ArXiv./abs/2007.14062