Spotlight on Video Piracy Websites: Familial Analysis Based on Multidimensional Features

Chenlin Wang[1], Yonghao Yu[1], Ao Pu[1], Fan Shi[2,3(✉)], and Cheng Huang[1,3]

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, China
{wangchenlin,yuyonghao,puao}@stu.scu.edu.cn,
codesec@scu.edu.cn

[2] College of Electronic Engineering, National University of Defense Technology, Hefei, China
shifan17@nudt.edu.cn

[3] Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei, China

Abstract. With the gradual increase in awareness of intellectual property protection in recent years, it has become imperative to strengthen the monitoring and regulation of digital piracy. The previous countermeasures suffer from low accuracy or passive data collection. Furthermore, the commonly adopted website clustering methods focus exclusively on a few attributes. The results obtained do not draw a comprehensive picture of the connections between websites within a family. In this paper, we aim to address the issue of digital piracy being challenging to identify, trace, monitor, and regulate in the current situation, utilizing video piracy websites targeting Chinese consumers as an example. The present architecture enables proactive discovery and detection of suspicious websites with a 96.2% accuracy, compensating for traditional digital piracy detection inadequacies. The proposed novel feature extraction method for clustering video piracy websites can synthesize multiple aspects in terms of layout, content, and infrastructure. The clustering results indicate that the websites belonging to the same family obtained by the proposed method show a more comprehensive similarity.

Keywords: Proactive discovery · Piracy detection · Multidimensional features · Feature serialization · Website clustering

1 Introduction

Along with the flourishing development of the Internet and the continuous improvement of network infrastructure, the number of Chinese Internet users is booming. According to The 48th Statistical Report on China's Internet Development [6], the amount of China's online video (including short video) users had reached 944 million by June 2021, accounting for 93.4% of all Internet users. At

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 272–288, 2022.
https://doi.org/10.1007/978-3-031-10989-8_22

the same time, the black market of pirated videos is overgrowing. According to iResearch's [11] definition, video piracy refers to the free or paid distribution of video resources on the Internet without the authorization of copyright owners and purchasers. Video Piracy Website (VPW) provides a staggering amount of pirated videos to Internet users. Operators of VPW continue to reach a high level of profitability by setting up VIPs, placing advertisements, and selling pirated videos. The operation mode of VPW is already well developed after going through stages of development such as the server storage model, P2P model, hotlink model, and cloud storage model. Currently, the development trend of VPW is from centralization to decentralization, bringing significant challenges to the fight against piracy.

Nowadays, several websites allow users to download pirated videos from streaming cyberlockers directly. Streaming cyberlockers have several things in common: first, they have no copyright checking policy; second, these websites often use circumvention tactics, such as additional settings on the homepage to make the website appear compliant and legitimate; and finally, they mostly disable search functions to prevent visitors from finding the resources they store. Streaming cyberlockers rely on third-party indexing websites to provide a searchable directory of video links and maintain a query function. The blocking of third-party indexing websites does not affect the streaming cyberlockers, ultimately creating a scenario in which streaming cyberlockers and third-party indexing websites operate in a symbiotic relationship.

As of now, there have been several studies on video piracy. Lyu et al. [15] investigated Chinese Internet users' attitudes toward video piracy and examined the factors affecting their attitudes. AliCloud [2] extracted the copyright information in the video DNA and uploaded it for deposition. They realized the integration and data exchange to build a blockchain-based copyright protection solution. Ibosiola et al. [10] analyzed the links to cloud storage on the websites and studied these links' characteristics and potential relationships. The above work suffers from the issue that they can only perform passive defense or detect a single category of VPW. We need the ability to fight against piracy more proactively and to analyze the characteristics of particular VPW families more comprehensively.

In summary, the primary contributions of this paper are listed as follows:

- We combine the incremental crawler and the BERT model to build a VPW detection model, which achieves timely proactive discovery and high detection accuracy of 96.2%.
- We design an encoding method to store multidimensional features in sequences. Combined with the string comparison algorithm, we can use each feature to divide VPWs into different groups efficiently.
- We cluster websites using layout, content, and infrastructure features. The websites belonging to the same cluster obtained by the presented method possess a more comprehensive similarity.

The rest of this paper is structured as follows: Sect. 2 discusses the related work. Section 3 shows the overall architecture and details the methods we used to

detect and cluster VPW. Section 4 provides the experiments and analysis related to this work. Section 5 concludes the paper and proposes future works.

2 Related Work

2.1 Features Extraction

Content. Babapour et al. [3] used natural language processing and text mining techniques such as TF-IDF and SVD to extract the content feature from webpage text. Maktabar et al. [16] employed the Bag-of-Words technique and Part-of-Speech tags to construct the content feature vector and segregate the fraudulent websites.

Layout. Bozkir et al. [5] proposed a ranking approach that considers visual similarities among webpages by using layout-based and vision-based features. Experimental results showed that their approach promisingly simulates the average human similarity judgment. Balogun et al. [4] proposed a meta-learning model based on the functional tree for detecting phishing websites. Experimental results showed that the functional tree and its variants are superior to some existing phishing websites detection models. Mao et al. [18] focused on extracting features from CSS layout files. To extract and quantify CSS features, they transferred their property values into computable types by doing some simplified encoding.

Website Infrastructure. The technology of Cyberspace Surveying and Mapping (CSM) detects, analyses, and visualizes all kinds of cyberspace resources and their relationships [27]. By building a map of cyberspace resources, we can comprehensively describe and display cyberspace information. Cyberspace search engines adopt CSM technology. Unlike Web search engines, cyberspace search engines can obtain the critical information of the target and conduct a comprehensive analysis and display [14]. Many cyberspace search engines have been well developed, such as Shodan, Zoomeye, Censys and FOFA.

2.2 Website Detection

The main approaches to detecting websites can be broadly divided into two categories, traditional machine learning approaches [13,17,24,25] and deep learning approaches [9,20,21,28]. Researchers do not need to extract features manually or have extensive knowledge while using deep learning methods. The latest studies in website detection have focused on deep learning models in recent years. Du et al. [9] proposed an intelligent classification schema based on the deep neural network using mixed featured extractors consisting of Text-CNN Feature Extractor and Bidirectional GRU Feature Extractor. Patil et al. [20] proposed a deep neural network to predict structural similarity between 2D layouts using

Graph Matching Networks (GMN). Their network's retrieval results are more consistent with human judgment of layout similarity. Yang et al. [28] proposed a multidimensional feature phishing detection approach. Under the control of a dynamic category decision algorithm, the speed and accuracy are all improved. Rajaram et al. [21] proposed a Convolutional Neural Network framework with 18 layers and transfer learning to classify websites using screenshots and URLs of phishing websites and legitimate websites. They achieved the integration of the visual-similarity-based and the character-based approach.

2.3 Websites Clustering

Jie et al. [12] proposed a label to class clustering analysis method and label category similarity attribute to compare the relevance of different types of website addresses. The clustering results showed that darknet sites can be gathered together unsupervised. Nagai et al. [19] proposed a new malicious website identification technique by clustering the WS-trees. Experiment results verified that the proposed approach identifies malicious websites with reasonable accuracy. Rentea et al. [22] proposed a clustering algorithm to cluster an extensive collection of URLs and selected centroids from each cluster, such that each URL in the cluster is similar with at least a centroid. The proposed algorithm is fast and scales well on large datasets. Drew et al. [8] proposed an optimized combined clustering method by extracting key website features, including text, HTML structure, file structure, and screenshots of websites. The results showed that their method more accurately groups similar websites together than existing general-purpose consensus clustering methods.

3 Methodology

In this section, we provide a detailed description of the methods we used. We divide our research into four parts: proactive discovery, VPW detection, VPW feature extraction, and familial clustering. The overall architecture is shown in Fig. 1.

3.1 Proactive Discovery

We employ incremental crawlers in conjunction with automatic keyword and external link extraction. By observing the layouts of VPWs, we found that the majority of them would include external links at the bottom of the webpages. Since developers typically index the links pointing to the resources inside the website with relative paths, another characteristic of the external links is that the links start with "http(s)". Our crawler can locate and extract the external links more efficiently based on the above findings. We use a record table to avoid duplicate crawling of an existing one.

The encoding on a portion of webpages does not perfectly match how it claims to be encoded. We replace these garbled characters with spaces since

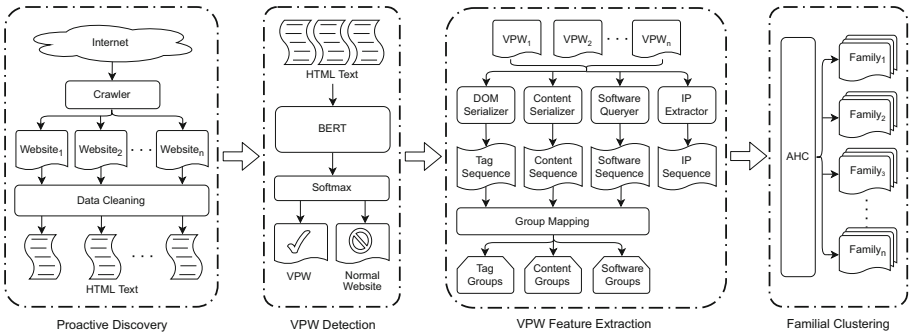

Fig. 1. Overall architecture.

they usually do not contribute to the characterization of the page. Additionally, the comments on websites are also meaningless, and we eliminate this part in the same manner. Then we extract the text from HTML files of all website homepages and record the mapping relations.

3.2 VPW Detection

The detection model in this paper is implemented based on the BERT model. BERT [7] stands for bidirectional encoder representations from transformers. Transformers can handle sequential input data and do not need to process the data in order based on the self-attention mechanism [26]. Self-attention can calculate interrelations between all words in a sentence. These interrelations then adjust the weight of each word to obtain their new expressions. These new expressions contain semantic meanings and interrelations. Therefore, the vector obtained by the self-attention mechanism has a more global expression than the traditional mechanism.

To further enhance the semantic representation of the model, we select the Chinese Wikipedia corpus as a pre-training task and then input HTML text into the model for targeted training.

3.3 VPW Feature Extraction

Layout Sequence. Researchers have conducted many studies on the layout similarity of webpages, primarily focusing on image-based comparisons. For example, in the research of phishing webpages, Abdelnabi et al. [1] proposed a method for comparing visual similarity using screenshots. Unfortunately, the image-based layout comparison method is too computationally expensive. To reduce the overhead while accurately recording the layout features of webpages, we serialize the layout into a one-dimensional sequence. Since some HTML tags can only uniquely represent the nested relationships when the start and end

positions are determined, we need to record them entirety. We selectively serialize tags frequently found in VPWs, further reducing the overhead. The encoding table is shown in Table 1.

Table 1. Encoding table.

Tag	Code	Tag	Code
a	A	li	G
div	B	input	H
link	C	form	I
img	D	p	J
script	E	table	K
ul	F	span	L

Here we give an example of webpage layout serialization. There are two simple webpages in Fig. 2, Page 1 will be encoded as "EEFGGGGGGFLLC", and Page 2 will be encoded as "FGGGGGGFCLL". It is worth noting that we only encode the tags in the encoding table.

```
<html>
<head>
    <title>page 1</title>
    <script>some code</script>
</head>
<body>
    <ul type="some type">
        <li>some text</li>
        <li>some text</li>
        <li>some text</li>
    </ul>
    <span>some text</span>
    <link rel="xxx" href="a link"/>
</body>
<html>
```

(a) Page 1

```
<html>
<head>
    <title>page 2</title>

</head>
<body>
    <ul type="some type">
        <li>some text</li>
        <li>some text</li>
        <li>some text</li>
    </ul>
    <link rel="xxx" href="a link"/>
    <span>some text</span>
</body>
<html>
```

(b) Page 2

Fig. 2. Example of HTML code for two similar webpages.

Content Sequence. Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical method to assess the importance of a word for one of the documents.

Term Frequency (TF) represents the number of times a word appears in a document. The importance of the word w in a document can be expressed as:

$$TF_{w,d} = \frac{n_{w,d}}{\sum_i n_{i,d}} \tag{1}$$

This formula calculates the importance of word w to the document d. $n_{w,d}$ indicates the number of times word w appears in document d. $\sum_i n_{i,d}$ is the sum of the occurrences for all words in document d.

Inverse Document Frequency (IDF) measures the ability of words to distinguish between categories. The ability of word w can be expressed as:

$$IDF_w = log\frac{|D|}{|\{d : n_w \in d_i\}|} \tag{2}$$

$|D|$ is the number of all documents. $|\{d : n_w \in d_i\}|$ is the number of documents containing word w.

TF-IDF represents the weight of words. The weight of word w in document d can be expressed as:

$$TF\text{-}IDF_{w,d} = TF_{w,d} \times IDF_w \tag{3}$$

We first extract all the Chinese corpus in HTML and train the TF-IDF model. Since the TF-IDF score's decile is the most significant number in this score, we take all the TF-IDF scores' deciles of a particular website and join them into a sequence. We give a few simple examples, as shown in Table 2.

Table 2. Example of content sequences for three websites.

Website	TF-IDF scores	Content sequence
1	0.21, 0.33, 0.00, 0.10, 0.56	23015
2	0.23, 0.37, 0.10, 0.17, 0.51	23115
3	0.42, 0.33, 0.00, 0.00, 0.22	43002

Website Infrastructure. We aggregate data from various cyberspace search engines, maximizing the combination of each search engine to make our statistics more comprehensive and reliable. We gather information about the server software to determine the habits of the operating entity building the website and IP address to determine the network segment in which the server is located. We also serialize the two features acquired above to evaluate the similarity of the infrastructure between websites in the subsequent step. We stitch together all the server software for the website into a sequence so that the sequences between websites with similar software is also alike. An example is shown in Table 3.

Table 3. Example of three websites using similar server software.

Website	Software
1	Nginx, jQuery-3.3.1, Bootstrap
2	CNZZ, Nginx, jQuery-3.2, Bootstrap
3	Swiper Slider, Nginx, jQuery-1.11.3, Bootstrap

We intercept the first three segments for the website's IP address, make up each segment into three digits, and then join the three parts into a nine-digit number. These nine-digit sequences can be used later in the clustering step to indicate the distance between network segments in which different websites are located. An example of the process is shown in Fig. 3.

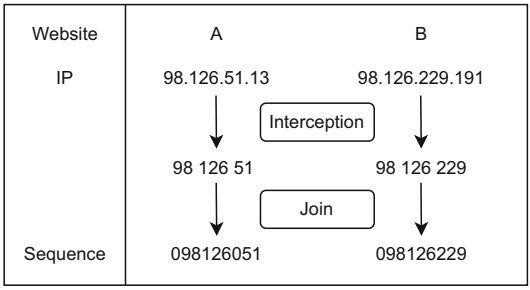

Fig. 3. Example of IP sequence processing.

Group Mapping. After the above extraction and encoding, we can obtain sequences of each webpage's layout, content, and software. We use a sequence comparison algorithm to calculate the similarity between two webpages in the above three aspects. The core formula of the algorithm is as follows.

$$similarity = \frac{Lsum - Ldist}{Lsum} \tag{4}$$

$Lsum$ is the sum of the lengths of two sequences. $Ldist$ is the edit distance, i.e., the minimum number of edit operations required to convert from one to the other between two sequences. Both insert and delete operations add 1 to the value of $Ldist$ and replace operations add 2. The edit distance of two sequences a and b is denoted as $Ldist_{a,b}(|a|, |b|)$, which can be calculated as follows.

$$Ldist_{a,b}(i,j) = \begin{cases} max(i,j) & \text{if } min(i,j) = 0 \\ min \begin{cases} Ldist_{a,b}(i-1,j) + 1 \\ Ldist_{a,b}(i,j-1) + 1 \\ Ldist_{a,b}(i-1,j-1) + 2_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \tag{5}$$

$|a|$ and $|b|$ denote the lengths of sequences a and b respectively. $2_{(a_i \neq b_j)}$ is a characteristic function with the value of 2 when $a_i \neq b_j$ and 0 otherwise. $Ldist_{a,b}(i,j)$ denotes the distance between the first i characters of a and the first j characters of b. After obtaining the edit distances of two sequences, we perform a similarity calculation. Here we take the sequences obtained from the two webpages shown in Fig. 2 as an example. The edit distance between the sequence of Page 1 and Page 2 is 4 and further get the similarity as 0.83. The comparison between software sequences or content sequences goes the same way.

The IP sequence can be standardized and fed directly to the clustering algorithm. However, layout, content, and software sequences must be mapped first. The specific implementation is shown in Algorithm 1.

Algorithm 1: Group mapping using sequence similarity.

Input: List of websites, L
Output: Dict of websites and group IDs, D

$D \leftarrow$ *empty dict*
GID $\leftarrow$ *new GID*
while L *is not empty* **do**
  $Similarity_{old} \leftarrow 0$
  $Website_1 \leftarrow L.pop()$
  $Mate \leftarrow Website_1$
  for $Website_2$ *in* L **do**
    $Similarity_{new} \leftarrow Compare(Website_1, Website_2)$
    if $Similarity_{new} > Similarity_{old}$ **then**
      $Mate \leftarrow Website_2$
      $Similarity_{old} \leftarrow Similarity_{new}$
    end
  end
  if $D.get(Mate)$ *is not None* **then**
    $D[Website_1] \leftarrow D.get(Mate)$
  else
    $D[Website_1] \leftarrow GID$
    $D[Mate] \leftarrow GID$
    GID $\leftarrow$ *new GID*
  end
end

Algorithm 1 first initializes an empty dictionary D, generates a random *GID*. Then the following operations are performed for each website in the list L. The algorithm first pops up a website, assigns it to $Website_1$, and initializes the *Mate* of $Website_1$ to itself and $Similarity_{old}$ to 0. Then, $Website_1$ compares with each remaining $Website_2$ in list L. If the $Similarity_{new}$ between $Website_1$ and $Website_2$ is higher than the $Similarity_{old}$, the algorithm sets the *Mate* of $Website_1$ to $Website_2$ and assigns the $Similarity_{new}$ to $Similarity_{old}$. After $Website_1$ finishes comparing with all the websites in list L, the algorithm then tries to find out if the *Mate* is already grouped in the dictionary. If yes, the

algorithm then assigns the *Mate*'s group ID to $Website_1$. Otherwise, the algorithm assigns *GID* to both $Website_1$ and *Mate*, generating a new *GID*. Finally, Algorithm 1 divides two websites corresponding to the highest sequence similarity into a group. After the algorithm execution, each website is assigned three group IDs according to its layout, content, and software.

3.4 Familial Clustering

Clustering is the task of dividing a dataset into different clusters according to specific criteria. The data object is more similar to other data objects in the same cluster and dissimilar to data objects in other clusters. Hierarchical clustering is a method that decomposes a dataset hierarchically until certain conditions are met. Traditional hierarchical clustering algorithms are divided into two main algorithms: divisive clustering and agglomerative clustering. Divisive clustering uses a top-down strategy. This algorithm splits a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data is split into singleton clusters. Agglomerative clustering uses a bottom-up strategy. This algorithm treats each data as a singleton cluster at the outset. It then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

We use the three group IDs and IP sequences as input features to cluster the VPWs. We choose Agglomerative Hierarchical Clustering (AHC) as our clustering method. And we use StandardScaler to standardize features by removing the mean and scaling to unit variance.

4 Experiment

In this section, we use collected data to train the BERT model and perform VPW detection. We then conduct a familial clustering and visualize the results. We analyze two cases to illustrate the superiority of using multidimensional features for clustering.

4.1 Dataset

The experimental dataset are collected by the methods mentioned in Sect. 3.1 which contain 22363 pieces of data. To train the BERT model, we label some data with 1, meaning the VPW (blacklist) and 0, meaning the normal websites (whitelist). After the manual screening, the training dataset has 1752 whitelist data and 1336 blacklist data. Each piece of data includes the URL, domain name, title, update time, Etc. Then we download all websites' HTML files and extract the text using the methods mentioned in Sect. 3.1.

4.2 Evaluation Metrics

We use weighted Accuracy, Precision, Recall, and F1-score to evaluate the performance of the BERT model. Accuracy represents the percentage of correct predictions. The Precision represents the accuracy of predicting positive samples. The Recall represents the probability of predicting positive samples in the actual positive samples. The F1-score comprehensively reflects Precision and Accuracy.

We use the silhouette coefficient [23] to evaluate the performance of AHC. The silhouette coefficient is a measure of clustering validity, and its core formula is as follows.

$$SilhouetteCoefficient_i = \frac{(b_i - a_i)}{max(a_i, b_i)} \tag{6}$$

In the formula, i denotes object i. a_i is the average intra-cluster distance, i.e., the average distance between i and the other objects within a cluster. b_i is the average inter-cluster distance, i.e., the average distance between i and all objects in other clusters. The silhouette coefficient ranges from –1 to 1. A large positive value indicates that the cluster's intra-dissimilarity a_i is much smaller than the smallest inter-dissimilarity b_i, and thus object i is well-clustered.

4.3 Experimental Settings

VPW Detection. The experiment uses the BERT model, choosing the Voting Ensemble model (consisting of Logistic Regression, Random Forest, Gradient Boosting, and Xgboost) and RNN model as a comparison. We randomly split our dataset into train and validation sets using a 90/10 cut and take HTML text as model input. The size of the BERT embedding vector is set to 768. The learning rate is $5e^{-5}$, and the batch size is 8.

Familial Clustering. We divide this subsection into two parts: pre-processing and parameter setting.

Pre-processing. The layout sequence preserves the layout information of webpages well and makes it possible to compare a large number of webpages. We divide the tag sequence into four segments of 250 characters each and compare the similarity of the characters in each of the four segments between every two webpages. The stacked histogram of statistical results is shown in Fig. 4.

Four colors indicate the similarity distribution in the four segments shown in the stacked histogram. The heights of the bars in different colors represent the number of a certain similarity in each segment. The trend of the similarity distribution of layout sequences for different segments is about the same. It is noteworthy that many samples with similarities of 0 and 1 appear in the statistics for characters 500–750 and 750–1000. This situation occurs because some websites have layout sequences less than or equal to 500 in length. If we compare an empty sequence with a non-empty sequence, the similarity is 0. If we

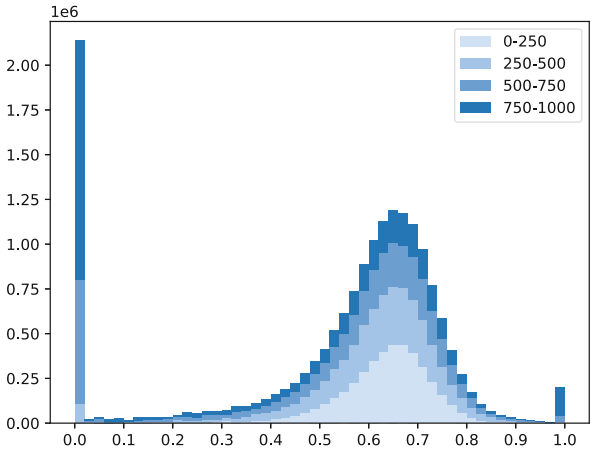

Fig. 4. Similarity distribution calculated using different segments of the layout sequence.

compare two empty sequences, the similarity is 1. To reduce the effect of extreme values on the detection results, we take the first 500 characters of each webpage's layout sequence to compare. In addition, sequence truncation emphasizes the top of the webpage, which significantly impacts the webpage's layout. It reduces the effect of the bottom of the webpage on the similarity calculation, which generally only affects the number of resources provided by the webpage.

The processing of content and software sequences does not need to be as complex as layout sequences. We can easily obtain these two sequences and use Algorithm 1 to process them into different groups, respectively. We obtain the IP sequence of each website with the process shown in Fig. 3. IP sequences do not need to be mapped to groups but only standardized. The standardized mapping results of the four aspects mentioned above are shown in Fig. 5.

Hyperparameter Tuning. In order to select a suitable hyperparameter, We use the silhouette coefficient for evaluation. The variation of the silhouette coefficient and the number of clusters with the distance threshold is shown in Fig. 6. The line indicates the silhouette coefficient, and the bars are the number of clusters. We chose 0.10 as the distance threshold for clustering, and the silhouette coefficient is 0.712.

4.4 Results

Detection Results. The results of the VPW detection are shown in Table 4.

The experimental results show that the BERT model used in this article can improve the detection effect of VPW. With the BERT model, we automatically classify the remaining unlabelled websites collected in Sect. 4.1. Finally, the model classifies 10978 VPWs and 11385 normal websites for our subsequent familial clustering.

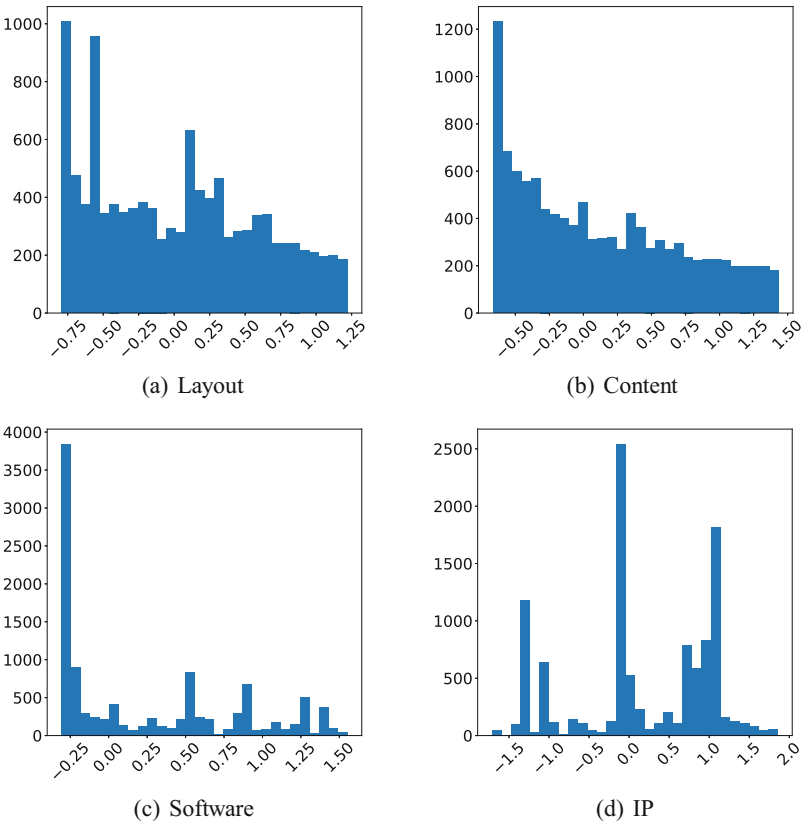

Fig. 5. The number of samples corresponding to different groups of each feature.

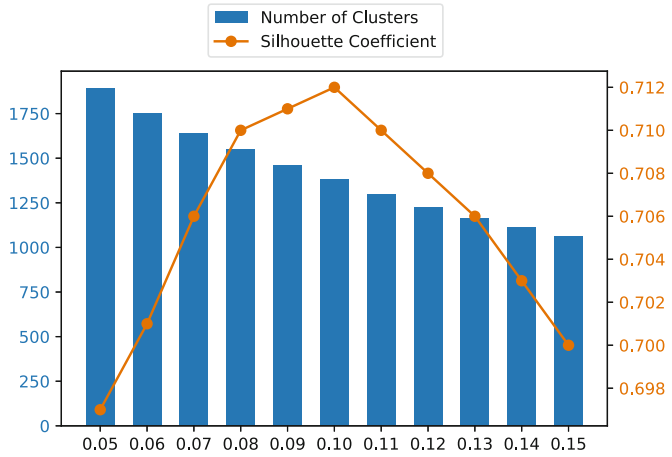

Fig. 6. Variation of silhouette coefficient and the number of clusters with distance threshold.

Table 4. Comparison of Voting Ensemble, RNN and BERT.

Model	Accuracy	Precision	Recall	F1-score
Voting Ensemble	87.97%	88.20%	87.97%	88.03%
RNN	92.78%	92.86%	92.78%	92.80%
BERT	**96.20%**	**96.21%**	**96.20%**	**96.20%**

Clustering Results. After the familial clustering, we visualize the clustering results as a graph. The graph of partial clustering results is shown in Fig. 7. The large nodes represent cluster labels. The small nodes associated with the large node represent the VPWs belonging to the cluster.

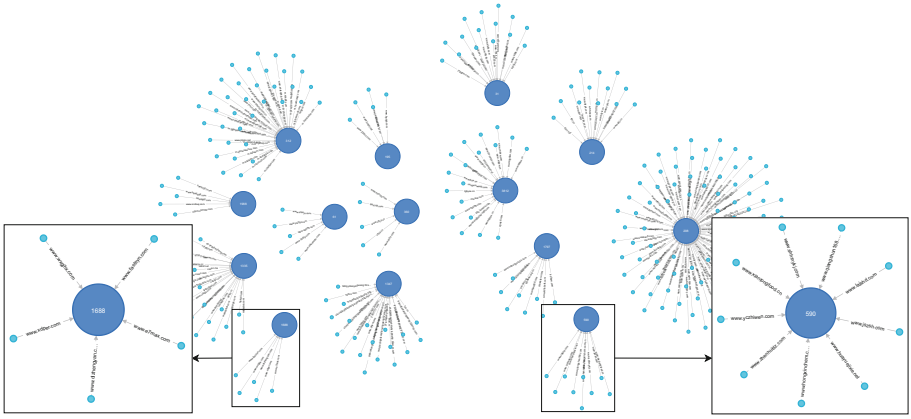

Fig. 7. Partial clustering results.

4.5 Case Study

We provide two cases to illustrate the clustering results. We call the group corresponding to the largest proportion of a feature in a cluster the dominant group. We further show the percentage of the dominant group within the cluster for each feature. For the cluster labeled 590 in Fig. 7, the statistical results are shown in Table 5. For the cluster labeled 1688 in Fig. 7, the statistical results are shown in Table 6. The same cluster of websites obtained by our method is more similar in layout, content, and infrastructure. Our clustering results allow a more comprehensive familial analysis of VPWs.

Table 5. VPWs labeled 590.

Feature	Percentage
Layout	88.89%
Cotent	77.78%
Software	100.00%
IP	66.67%

Table 6. VPWs labeled 1688.

Feature	Percentage
Layout	100.00%
Cotent	60.00%
Software	100.00%
IP	80.00%

5 Conclusion and Future Work

In this paper, we collected video piracy websites through proactive discovery and achieved high detection accuracy using the BERT model. We proposed a serialization method that encodes the website's features in terms of layout, content, and infrastructure into sequences. Our group mapping algorithm accurately grouped websites with similar sequences in the aforementioned aspects. By feeding the processed features into the AHC, we created a hitherto unseen VPW clustering method based on multidimensional features. The VPWs in the same cluster were almost identical, demonstrating the reliability of our clustering method. Our future work will focus on increasing efficiency and expanding the application of website familial analysis.

Acknowledgements. This research is funded by the National Key Research and Development Program of China (No. 2021YFB3100500), Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation (No. CSSAE-2021-001).

References

1. Abdelnabi, S., Krombholz, K., Fritz, M.: Visualphishnet: zero-day phishing website detection by visual similarity. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1681–1698 (2020)
2. Alicloud (2022). https://www.aliyun.com/solution/blockchain/bcpp
3. Babapour, S.M., Roostaee, M.: Web pages classification: an effective approach based on text mining techniques. In: 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 0320–0323. IEEE (2017)
4. Balogun, A.O., et al.: Improving the phishing website detection using empirical analysis of function tree and its variants. Heliyon **7**(7), e07437 (2021)
5. Bozkir, A.S., Sezer, E.A.: Layout-based computation of web page similarity ranks. Int. J. Hum. Comput. Stud. **110**, 95–114 (2018)
6. CNNIC: The 48th statistical report on China's internet development. Technical report, China Internet Network Information Center (2021)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

8. Drew, J.M., Moore, T.: Optimized combined-clustering methods for finding replicated criminal websites. EURASIP J. Inf. Secur. **2014**(1), 1–13 (2014). https://doi.org/10.1186/s13635-014-0014-4
9. Du, M., Han, Y., Zhao, L.: A heuristic approach for website classification with mixed feature extractors. In: 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), pp. 134–141. IEEE (2018)
10. Ibosiola, D., Steer, B., Garcia-Recuero, A., Stringhini, G., Uhlig, S., Tyson, G.: Movie pirates of the caribbean: exploring illegal streaming cyberlockers. In: Twelfth International AAAI Conference on Web and Social Media (2018)
11. iResearch: 2018 report of copyright protection in China's pan-entertainment industry. Technical report, iResearch (2018)
12. Jie, X., Haoliang, L., Ao, J.: A new model for simultaneous detection of phishing and darknet websites. In: 2021 7th International Conference on Computer and Communications (ICCC), pp. 2002–2006. IEEE (2021)
13. Kumar, J., Santhanavijayan, A., Janet, B., Rajendran, B., Bindhumadhava, B.: Phishing website classification and detection using machine learning. In: 2020 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–6. IEEE (2020)
14. Li, R., Shen, M., Yu, H., Li, C., Duan, P., Zhu, L.: A survey on cyberspace search engines. In: Lu, W., et al. (eds.) CNCERT 2020. CCIS, vol. 1299, pp. 206–214. Springer, Singapore (2020). https://doi.org/10.1007/978-981-33-4922-3_15
15. Lyu, Y., Xie, J., Xie, B.: The attitudes of Chinese online users towards movie piracy: a content analysis. In: Sundqvist, A., Berget, G., Nolin, J., Skjerdingstad, K.I. (eds.) iConference 2020. LNCS, vol. 12051, pp. 169–185. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43687-2_13
16. Maktabar, M., Zainal, A., Maarof, M.A., Kassim, M.N.: Content based fraudulent website detection using supervised machine learning techniques. In: Abraham, A., Muhuri, P.K., Muda, A.K., Gandhi, N. (eds.) HIS 2017. AISC, vol. 734, pp. 294–304. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76351-4_30
17. Malhotra, R., Sharma, A.: An empirical study to classify website using thresholds from data characteristics. In: Hu, Y.-C., Tiwari, S., Mishra, K.K., Trivedi, M.C. (eds.) Ambient Communications and Computer Systems. AISC, vol. 904, pp. 433–446. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-5934-7_39
18. Mao, J., et al.: Phishing page detection via learning classifiers from page layout feature. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–14 (2019). https://doi.org/10.1186/s13638-019-1361-0
19. Nagai, T., et al.: A malicious web site identification technique using web structure clustering. IEICE Trans. Inf. Syst. **102**(9), 1665–1672 (2019)
20. Patil, A.G., Li, M., Fisher, M., Savva, M., Zhang, H.: Layoutgmn: neural graph matching for structural layout similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11048–11057 (2021)
21. Rajaram, J., Dhasaratham, M.: Scope of visual-based similarity approach using convolutional neural network on phishing website detection. In: Satapathy, S.C., Bhateja, V., Janakiramaiah, B., Chen, Y.-W. (eds.) Intelligent System Design. AISC, vol. 1171, pp. 435–452. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-5400-1_45
22. Rentea, R., Oprişa, C.: Fast clustering for massive collections of malicious URLs. In: 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 11–18. IEEE (2021)
23. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)

24. Shabudin, S., Sani, N.S., Ariffin, K.A.Z., Aliff, M.: Feature selection for phishing website classification. Int. J. Adv. Comput. Sci. Appl. **11**(4), 587–595 (2020)
25. Ubing, A.A., Jasmi, S.K.B., Abdullah, A., Jhanjhi, N., Supramaniam, M.: Phishing website detection: an improved accuracy through feature selection and ensemble learning. Int. J. Adv. Comput. Sci. Appl. **10**(1), 252–257 (2019)
26. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
27. Xu, R., et al.: Cyberspace surveying and mapping: Hierarchical model and resource formalization. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 68–72. IEEE (2019)
28. Yang, P., Zhao, G., Zeng, P.: Phishing website detection based on multidimensional features driven by deep learning. IEEE Access **7**, 15196–15209 (2019)