Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4408987, 17 pages
https://doi.org/10.1155/2022/4408987

Research Article

Birds of a Feather Flock Together: Generating Pornographic and Gambling Domain Names Based on Character Composition Similarity

Yanan Cheng,[1] **Hao Jiang**,[1] **Zhaoxin Zhang**,[1] **Yuejin Du**,[2] **and Tingting Chai**[1]

[1]*Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China*
[2]*Beijing Qihoo Technology Co., Ltd, Beijing 100015, China*

Correspondence should be addressed to Zhaoxin Zhang; zhangzhaoxin@hit.edu.cn and Tingting Chai; ttchai@hit.edu.cn

Received 10 May 2022; Revised 5 June 2022; Accepted 23 June 2022; Published 11 July 2022

Academic Editor: Yuanlong Cao

Cybercriminals often register many pornographic or gambling domains (known as abusive domains) with similar character compositions in bulk to reduce their investment in buying domains and make it easier for clients to remember and spread them. Therefore, this study combines the ideas of text similarity and text generation and proposes an abusive domain generation model based on GRU for rapidly generating new abusive domain names from known ones. Additionally, we develop a two-layer detection system for pornography and gambling domains using fastText and CNN models to obtain an abusive domain dataset for model training and validation. In the end, our detection system identifies pornographic and gambling domains with 99% precision while balancing correctness and speed. By inputting 40,000 random keywords into the abusive domain generation model, we obtained 130,220 online domains that served web pages, of which about 66% were pornographic or gambling domains. The results show that by exploiting cybercriminals' behaviors in registering abusive domain names, such as bulk registration of similar domain names, we can prospectively acquire a large number of new abusive domains based on known ones. This study demonstrates that predicting new abusive domains not only expands the domain blacklist but also allows researchers to target the generated suspicious domains and dispose of them in time before they show abusive behavior.

1. Introduction

Cybercriminals are establishing more and more pornographic and gambling domains (or websites, collectively referred to as abusive domain names) in pursuit of profit. At the same time, with the growth of the Internet and social media, people are increasingly exposed to these abusive domain names, either intentionally or unintentionally, including children and minors. Pornographic videos and images hurt the physical and mental health of minors. Many gambling sites are fraudulent sites that cheat people out of their money [1]. At the same time, the current state of the global epidemic of COVID-19 has led to an even more rampant spread of pornographic and gambling domains on the Internet [2–7]. Therefore, the sooner governments, security institutions, and Internet entities can discover, block, and handle these pornographic and gambling domains, the more they can mitigate the harm caused by these domains [8]. Therefore, from a technical perspective, the first significant challenge for each Internet entity is how to quickly, accurately, and early discover pornography and gambling domains, which is the research objective of this paper.

Generally, much of the existing research in pornography and gambling domain discovery focuses on detection, where the website (domain) is entered into the detection model. Then, information about the website (e.g., text or images)

is used to determine whether the domain is pornographic or gambling. These detection methods are necessary to discover pornographic and gambling domains. However, detection methods cannot discover abusive domains earlier because they can only detect the domains that are entered into the models. In order to discover the abusive domains earlier, we need to adopt a new perspective to start with.

Through our empirical analysis of many pornographic and gambling domain names, we find that there are similarities in the composition of these domain names. These similar characteristics are mainly reflected in two aspects. On the one hand, to facilitate abusive domain management and memorization, cybercriminals register many domain names with similar compositions, such as *porn[0-9].com*, in bulk. On the other hand, many pornographic and gambling domain names have no special meaning but are just combinations of letters and digits. Because domains with meaningful word combinations are expensive to register, cybercriminals register many domain names with meaningless compositions in bulk to reduce the investment in malicious attack activities (Section 2).

Therefore, in this paper, we develop a two-layer detection system for pornography and gambling domains using fastText and CNN models (Section 3.1), which is able to identify abusive domains quickly and accurately. Meanwhile, using the compositional similarity features of many pornographic and gambling domains, we combine the ideas of text similarity and text generation and propose a novel abusive domain generation model based on GRU to generate new pornographic and gambling domains from existing ones (Section 3.2 and Section 3.3). Finally, our detection system identifies pornographic and gambling domains with 99% precision while balancing correctness and speed. By inputting 40,000 random keywords into the abusive domain generation model, we obtained 130,220 online domains that served web pages, of which about 66% were pornographic or gambling domains (Section 4).

In short, we make the following contributions:

(i) We develop a two-layer detection system using fastText and CNN models to identify pornographic and gambling domains. The system is capable of ensuring high detection efficiency while maintaining a high detection accuracy rate for abusive domain names. In addition, this method can exclude websites that contain only pornographic or gambling keywords in the text of the page

(ii) For the first time, using existing abusive domains, we propose a novel approach to generate many new and undiscovered abusive domains based on domain composition similarity. This method enables us to discover many pornographic and gambling domains earlier so that they can be blocked and handled in a timely manner

(iii) For the first time, we share a database (https://reurl.cc/0p27db, accessed on 6 May 2022, access password: nist@HIT) of manually labeled website snapshots of abusive domains, containing 18,428 pornography domains and 15,578 gambling domains. We hope that more security communities and researchers can use these samples for research on pornography and gambling domain detection or generation

In summary, this paper aims to discover a large number of pornographic and gambling domains as quickly, accurately, and early as possible. This paper is intended for audiences across Internet infrastructure, cybersecurity industries, and researchers.

2. Background and Related Work

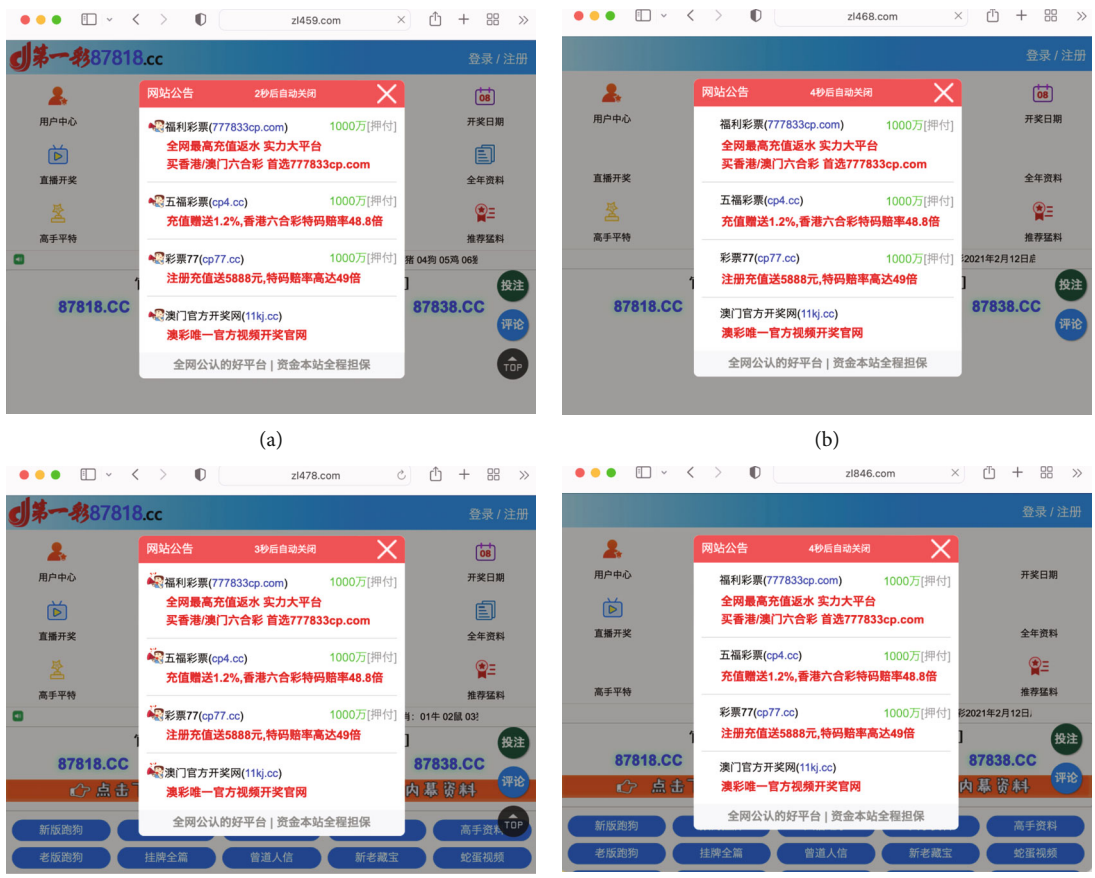
2.1. Background

2.1.1. Similarity in the Composition of Abusive Domains. One of the fundamental assumptions of this study is that a substantial number of abusive domain names share a common character composition, i.e., they follow the same composition rules. We provide some examples of domain names with similar character compositions. As shown in Figure 1, the four gaming domains display the exact same web page content, and their domain name character composition rules conform to the rules *zl* ∗∗∗ *.com*. The same situation exists for pornographic domains, as shown in Figure 2, which all conform to rule ∗∗∗∗ *av.com*.

In addition, we checked the domain name certificate of the gambling domain *lh1769.com*, as shown in Figure 3, and found 120 domain names with similar character composition to this gambling domain. Once again, it is proven that pornography or gambling sites use a large number of domain names with similar character composition. This case not only facilitates distribution but also makes it easy for the viewer to remember the domains of sites, and when a domain name is not available to access the site, the viewer uses a new domain name to access it.

By observing many abusive domain names of pornography and gambling types, we found two main characteristics of these abused domain names. First, they are mainly composed of pure numbers or a mixture of numbers and letters with no real meaning. Second, the similarity is primarily shown by the fact that some characters (numbers or letters) in the domain name stay the same, but many other characters in their next or previous positions change. Therefore, we can design methods to discover abusive domain names of similar composition as soon as possible using these characteristics.

On the other hand, many pornography and gambling domains consist of meaningless letters and numbers, as described above. Figure 4 shows the frequency of letters and digits at various positions in popular domains (Alexa top 1 million, http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m.csv.zip, accessed on 6 May 2022) and abusive domains (domain length less than 16). We can find that the frequency of characters in abused domain names is different from popular domain names, especially a large number of numeric characters that appear in pornography and gambling domain names.



(a) (b) (c) (d)

Figure 1: Gambling domain names with similar character composition. (a) Domain *zl459.com*. (b) Domain *zl468.com*. (c) Domain *zl478.com*. (d) Domain *zl846.com*.

In general, the similarity in character composition and nonsense of many pornography and gambling domain names provides a practical basis for generating new abusive domain names.

2.1.2. Abusive Domains in Disguise. Pornography and gambling websites have apparent textual features, such as many keywords (https://github.com/mrcheng0910/reporting_abusive_domains/blob/main/abusive_keywords.txt, accessed on 6 May 2022) related to pornography or gambling. Therefore, high detection accuracy can be achieved by designing a text-based classifier. Yang et al. designed and implemented an SVM-based classifier to achieve 99% accuracy in detecting online gambling websites [1]. Therefore, we refer to text-based related methods to filter gambling and pornographic websites from the textual perspective.

On the other hand, miscreants from online underground economies regularly exploit website vulnerabilities and inject fraudulent content into web pages to promote illicit goods and services. Adversaries often manage to inject content stealthily by obfuscating the description of illegal products and/or the presence of defacements to make them undetectable [9]. As shown in Figure 5, gambling-related keywords are maliciously embedded in the title, description, and keyword tags of the normal website, respectively. However, the page displayed to the users in the browser is benign.

As a result, such sites are easily misclassified as abusive domain names through text-based classifiers. In view of this situation, this paper implements an image-based abusive domain name detection tool in addition to developing a text-based filter. The text-based filter is fast, consumes fewer resources, and can filter out abusive domain names from a large number of websites as quickly as possible. The image-based classifier further detects the filtered abusive domain names to improve the final detection accuracy. In this paper, we use convolutional neural networks (CNNs) to detect website snapshots to find pornographic or gambling domains. We describe both methods in detail in Section 3.1.

2.2. Related Work

2.2.1. Abusive Domain Detection. Srinivasan et al. [10] present DeepURLDetect (DURLD), a method that extracts features from character level embedding using hidden layers in deep learning architectures and then uses a nonlinear activation function to predict the likelihood of the URL is malicious or not. Lison et al. [11] established a model for detecting domain generation algorithm (DGA) domains using recurrent neural networks. This model was capable of making predictions only based on domain names, without the need for human participation or access to external

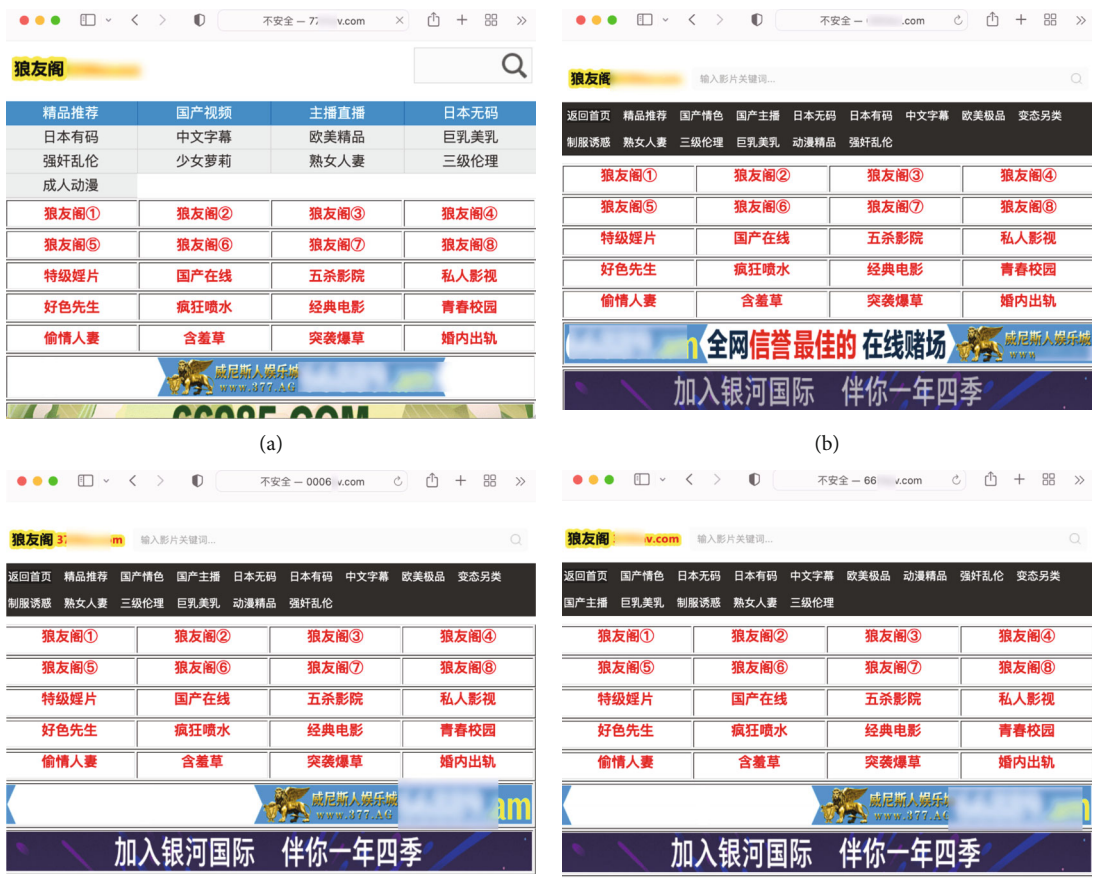


Figure 2: Pornographic domain names with similar character composition. (a) Domain *7757av.com*. (b) Domain *0004av.com*. (c) Domain *0006av.com*. (d) Domain *6687av.com*.

Figure 3: Multiple abusive domain names share one certificate. (a) Domain *lh1769.com*. (b) List of domains that use the domain *lh1769.com*'s certificate.

resources. Curtin et al. [12] provided a novel machine learning system built partially on recurrent neural network (RNN) that is capable of classifying DGA-generated domain names even from families traditionally understood as difficult. Xu et al. [13] proposed a novel n-gram combined character-based domain classification (n-CBDC) model using n-grams and a deep convolutional neural network. This model operates end-to-end and does not require manually extracted features or DNS context information; it only requires the domain name itself as input and can

Figure 4: The frequency of characters at each position in the popular and abusive domains.

Figure 5: Gambling-related keywords are maliciously embedded in the benign website.

automatically assess the probability that the domain name was formed using DGAs. Bharathi et al. [14] proposed to take a string of characters as the input given in the domain names and classify them as either benign or malicious domain names using deep learning architectures such as Long-Short-Term Memory (LSTM) and bidirectional LSTM. Ren et al. [15] applied a deep neural network model with an attention mechanism (ATT-CNN-BiLSTM) for the detection and classification of DGA domain names. The main thought behind their ensemble model is that the validity of the context inherent in domains could contain sufficient information with which to distinguish DGA domain names, especially the wordlist-based ones. The majority of traditional approaches focus on a particular feature of these pornographic and gambling websites, which leaves out more nuanced and problematic scenarios. Chen et al. [16] developed an automatic detection system for pornographic and gambling websites based on visual and textual content using a decision process to address this issue. Similarly, Zhao et al. [17] proposed Porn2Vec, a robust end-to-end framework for detecting pornographic websites using contrastive learning. This framework, in particular, models pornographic websites as a heterogeneous network composed of websites, web pages, images, and text, as well as their interaction relationships, and formalizes pornographic website identification as a node classification problem on the graph. Additionally, the model employs a novel contrastive learning-based heterogeneous graph embedding method to learn the high-level representation of web pages by combining image-based, text-based, and structure-based information

concurrently. Finally, the learned website characteristics are sent into a neural network to train an automatic pornographic website detection model.

2.2.2. Prediction or Generation Based on RNN. RNN is one of the most promising tools for deep learning, which has been applied to speech recognition, machine translation, music generation, and text generation in a large number of previous studies [18–24]. For the first time, we applied it to domain name generation based on the idea of text generation.

The RNN is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data. Therefore, many studies use RNN for prediction or generation tasks. Wang et al. [18] proposed a novel attention-based LSTM [25] model for song iambic generation. Specifically, they encoded the cue sentences by a bidirectional LSTM model and then predicted the entire iambic with the information provided by the encoder, in the form of an attention-based LSTM that can regularize the generation process by the fine structure of the input cues. Sturm et al. [19] applied the LSTM model to music transcription modeling and composition. They built and trained the LSTM network using approximately 23,000 music transcriptions expressed using a high-level vocabulary (ABC notation) and then used them to generate new transcriptions. For the purpose of generating Chinese classic poetry, Luo et al. [21] introduced a novel text steganography technique based on the RNN encoder-decoder structure. They employed a keyword to construct the first line of a quatrain and then generated the subsequent lines one by one using the LSTM model. Additionally, they used a template-based generating method and established a word-choosing strategy based on inner-word mutual knowledge to combat poetry's dramatic decline in quality. Accurate and real-time traffic flow prediction is important for traffic control. Fu et al. [22] used LSTM and gated recurrent units (GRU) neural network methods to predict short-term traffic flow. Unlike prior template-based systems, Liu et al. [23] showed a system for generating Chinese classical poetry dubbed Deep Poetry that utilizes neural networks trained on over 200 thousand poems and 3 million pieces of ancient Chinese prose. This technology can generate Chinese classical poetry from plain text, images, or aesthetic notions. More importantly, this method allows users to engage in the process of composing poetry. Bartoli et al. [24] proposed and assessed a system for automatically generating restaurant reviews suited to the desired rating and restaurant category using LSTM. They trained the neural network on a set of authentic restaurant reviews in order to produce text that appears to be a restaurant review.

To summarize, the numerous existing approaches for detecting abusive domains listed above each rely on a single type of feature, such as the domain character, the URL, textual, or visual features. In comparison to these single-feature detection methods, hybrid feature-based methods perform better and offer broader development prospects. Therefore, this study combined the textual and visual features of the website to detect gambling and pornographic domains. Additionally, many of the different types of tasks (e.g., classical poetry and criticism) predicted or generated are carried out using the RNN model and perform well. Therefore, we generate new abusive domain names based on the RNN model.

In particular, because the purpose of the research described in this paper is to generate or predict new abusive domain names based on existing abusive domain names, the accuracy of detecting abusive domain names should be high enough. In addition, considering the significant resource and time consumption of image-based detection, therefore, we first filter out many suspected gambling and pornographic domains with a text-based detection method and then use an image-based approach for further verification. In this way, the accuracy and efficiency of domain name abuse detection meet the requirements.

3. Methodology

In this section, we design methods for generating more new abusive domains based on the abusive domain samples that have been acquired, as shown in Figure 6. Thus, our method consists of three major stages: acquiring abusive domain name samples; clustering abusive domain names based on similar composition rules; generating new abusive domain names based on these clusters.

3.1. Obtaining Abusive Domains. As shown in Figure 6, the work in this stage is mainly to build a database of abusive domains for generating new abusive domain names. This stage contains three main tasks: one is to obtain the web content of a large number of domain names, including HTML source code and snapshots; two is to detect pornographic and gambling domains based on HTML source code and snapshots, respectively; three is to discover more abusive domain names based on the certificate features of pornographic and gambling websites.

3.1.1. Crawling Web Content. First, we downloaded over 260 million domain names from Domain Monitor [26]. These domains come from 1500 zones, which indicates that these domains have DNS records. Second, we developed two types of web crawlers to crawl web content.

(i) Requests-based web crawler. This crawler uses the Python-requests [27] package to crawl the HTML source code of domains. Then, we extract the title, keywords, and description tags from the source code. This text information is used to determine if the domain name is pornographic or gambling

(ii) Selenium-based web crawler. This crawler uses the Selenium webdriver [28] to get the snapshots of the specific domains. We use these snapshots to detect pornographic and gambling domains. Compared to fetching web page source code, fetching web page snapshots is slower and consumes more computing resources

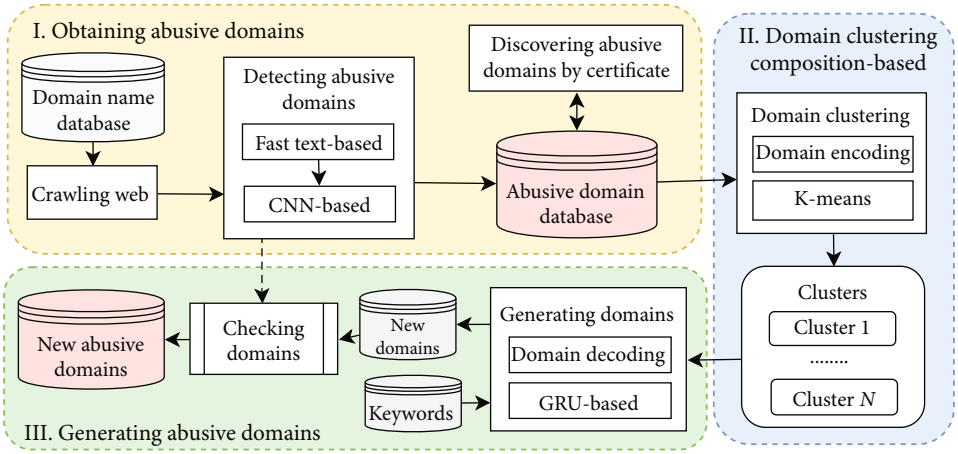


Figure 6: The process of generating new abusive domain names based on GRU-RNN.

We need to emphasize two points. On the one hand, in order to get as many websites with the same composition rules as possible, we get their web content in order based on the initial order of the domain names. On the other hand, we only need to obtain the web content of a small number of domains out of 260 million domains to meet our needs.

3.1.2. Detecting Abusive Domains Based on fastText. As introduced in Section 2, many previous studies detected the web page source codes of domains to determine whether they are pornographic or gambling. Also, both acquiring web page source code and abusive domain detection based on text are faster than acquiring web page snapshots and abuse detection based on the images. Therefore, text-based detection is the optimal solution when a large number of domains need to be detected while ensuring high efficiency and accuracy.

The fastText [29] is a natural language processing (NLP) library generally used for text representations and classification. The fastText does not need to rely on feature engineering like machine learning models for classification, and the classification effect does not depend on the selection of effective features. At the same time, although text classification based on deep learning can achieve good results and does not require feature engineering, the training speed is slow and the training conditions are high, so it cannot be used in large-scale text classification tasks. Therefore, the fastText model is widely used in text-based classification tasks because of its fast speed and good effect. Finally, this paper builds an abusive domain classifier with fastText based on HTML source codes. The process of detecting pornography and gambling domain names based on text is shown in Figure 7.

(1) Training and test sets. The training and test sets contain text samples that have been labeled as abusive or benign types. The text comes from the requests-based web crawler that gets the HTML source code of a large number of domains and extracts the key HTML tags content, i.e., title, description, and keywords. On the one hand, we obtain the source code of websites with a high traffic ranking in China from Alexa [30], and these text messages are labeled as benign. On the other hand, we obtain the source code of the web pages of the domains provided by Domain Monitor introduced above and filter out the text content matching pornographic and gambling keywords. These keywords (shared in GitHub [31]) are the more frequent Chinese words in pornography and gambling websites that we collected manually in the early stages, such as 做爱 (sex), 成人电影 (adult movies), and 澳门娱乐场 (Macau casinos). In addition, for the initially obtained benign and abusive texts, we manually filtered them again to ensure that the texts were indeed pornographic or gambling. In the end, we get a total of 31,667 benign and 177,963 abusive texts as training and test sets, which we will describe in detail in Section 4.1

(2) Text preprocessing. The task of text preprocessing for our dataset takes a few steps to convert the data into a convenient form that we can feed into the fastText classifier. First, since Chinese sentences are not separated by spaces like English sentences, we use the open-source tool Jieba [32] to split Chinese sentences into words. Jieba Chinese text segmentation is the best Python Chinese word segmentation module, and a lot of research relies on its excellent results. For example, the pornographic Chinese text "亚洲成人片不卡无码⊠天看片免费高清观看⊠国内自拍视频在线" (watch Asian adult movies and selfie porn videos for free every day) is divided into "亚洲 (Asian) 成人片 (adult videos) 不卡 (fluency) 无码 (codeless) 天天 (every day) 看片 (AV) 免费 (free) 高清 (high definition) 观看 (watch) 国内 (domestic) 自拍 (selfie) 视频 (videos) 在线 (online)." Second, in this step, we remove the repeated words after the sentence is divided. Also, remove meaningless words or symbols from the set of words, including stop words and special symbols, like #, ∗, and &, etc.

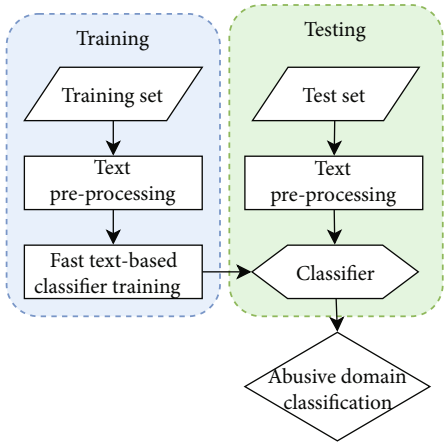


Figure 7: The process of detecting pornography and gambling domains based on fastText.

Finally, the fastText requires the labeling of each line in the database to be in a particular format like this: "__label__<label_name> <text>." For example, for pornographic text, it is "__label__ abusive 亚洲成人片不卡无码天天看片免费高清观看国内自拍视频在线." Another example, of benign text, is "__label__benign 百度一下⊠你就知道."

(3) FastText-based classifier training and classification. Text classification mainly uses a classifier to label the text of an unknown category, so the most important part of classification is the selection of the classification algorithm. The classification task proposed in this paper uses the fastText library by Facebook. Since the method needs the classification of texts with our prelabeled dataset, we used the supervised technique for text classification

Overall, the detection model we built based on fastText can detect a large number of domains containing pornographic and gambling texts. The detection model has the advantages of high accuracy, low resource consumption, and high efficiency, as detailed in Section 4.1.

3.1.3. Detecting Abusive Domains Based on CNN. As explained in Section 2.1.2, while some of the site's HTML information (title, description, and keywords) are related to pornography or gambling, the actual content presented on the site does not. As a result, we need to conduct additional research to determine whether the website is abusive. This section investigates the algorithm for detecting abusive domain names based on web page snapshots. Most pornographic or gambling websites have significantly different front-end design styles from benign websites. Therefore, features of both pages are automatically extracted by CNN, which are used to detect whether a domain name is being abused or not.

Abusive domain name detection based on web snapshots can be tried as an image binary classification problem, i.e., domain name snapshots are classified into benign and abusive results. In this paper, CNN is used to train the image recognition model, and the schematic diagram of the convolutional neural network structure is shown in Figure 8.

We input the web page snapshot into the convolutional neural network after it has been converted into an RGB 3D tensor of size 1600∗1000∗3. ReLU is applied nonlinearly to the output of the convolutional layer and the penultimate fully connected layer. The last fully connected layer uses a Sigmoid function to map the output to between 0 and 1 to obtain the probability that the input snapshot is legal for binary classification. In the training of the model, the loss function is a binary cross-entropy loss function, as shown in Equation (1).

$$\text{Loss} = \frac{-\sum_{i=0}^{n}(y_i \log(f_i(x_i))) + (1-y_i)\log(1-f_i(x_i))}{n}, \tag{1}$$

where n denotes the total number of output nodes, y_i is the real label corresponding to the i_{th} category, and $f_i(x)$ is the output value of the corresponding model.

Due to the large resolution of the input images, we increase the depth and width of the network in order to allow the neural network to extract its deep abstract features, reduce the number of parameters, and improve the classification accuracy. That is, the convolutional kernel width is 3 for each convolutional layer, and the convolutional kernel width is 2 for the pooling layer.

In order to improve the detection efficiency and accuracy of detecting pornographic and gambling domains, we combine two methods based on text filtering and image-based detection. That is, we first get a large number of suspected pornography and gambling websites by text-based filtering methods, then get snapshots of these websites, and finally detect whether these websites are really pornography and gambling websites by image-based methods.

3.1.4. Discovering Abusive Domains by Certificate. As described in Section 2.1.1, many abusive domains' certificates contain other abusive domains with a fairly similar character composition that can share these certificates. As a result, we developed tools to extract abusive domains from the certificates. Additionally, we extracted the primary domain name portion of the fully qualified domain names (FQDNs). For instance, for the domain https://www.abusive-domain.com, abusive-domain.com is the portion on which we concentrate our efforts. This enables us to acquire the maximum number of domain name composition rules feasible.

3.2. Domain Clustering Based on Composition. As described above, there are character compositional similarities between the different domain names in the set of abusive domain names. In other words, there are many different malicious domain names, but they come from various forms of character composition. In order to distinguish the abusive domain names belonging to different composition forms, we first cluster the abusive domain names based on the composition similarity of domain characters in order to provide training data for the domain name generation model.

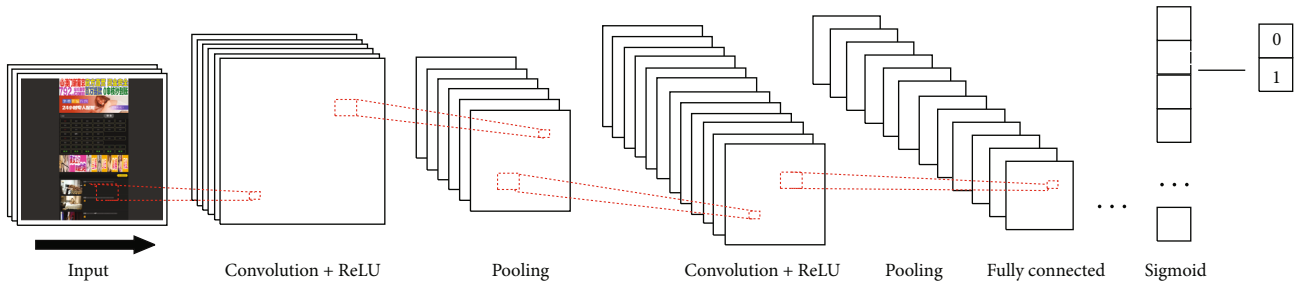


Figure 8: The process of identifying abusive website snapshots using convolutional neural networks.

3.2.1. Domain Encoding and Decoding. In order to achieve similar domain name clustering and interconversion with neural network acceptable input and output forms, encoding and decoding rules for domain names are constructed based on domain name composition features.

(i) *Domain Encoding*. Domain names are formed by the rules and procedures of the Domain Name System (DNS). The first-level set of domain names is the top-level domains (TLDs), including the generic top-level domains (gTLDs), such as .COM, .ORG, and .NET, and the country code top-level domains (ccTLDs). Below these top-level domains in the DNS hierarchy are the second-level and third-level domains (2LD and 3LD) that are typically open for reservation by end-users who wish to connect local area networks to the Internet, creating another publicly accessible Internet resource, or run websites. For example, for the domain name http://google.com, the top-level domain is .COM and the second-level domain is Google. We divide the domain name into two parts, i.e., 2LD.TLD, and encode the top-level domain and the second-level domain of the domain name, respectively. The second-level domains are all composed of letters a-z, numbers 0-9, and the ligature "-," while the top-level domains as a whole act as a specific unit due to their nondetachable nature. Therefore, we number all top-level domains and individual characters in the abusive domain dataset to form a domain-numbered dictionary. When encoding a domain name, first, we convert all characters and top-level domains in the domain name to their corresponding numbers, forming a domain-numbering vector of variable length. Then, the domain number vector is filled to the specified length with null characters to obtain a domain number vector of definite length composed of numbers.

(ii) *Domain Decoding*. Domain name decoding is the process of converting the domain name number vector output from the neural network into a domain name character vector. The domain name is obtained by looking up the corresponding characters and top-level domain names according to the character numbers in the domain character dictionary, obtaining a domain name character vector of definite length, and then removing the trailing null characters

3.2.2. Abusive Domains Clustering. In this paper, we use the K-means algorithm [33] to cluster domain names and divide the abusive domain names with similar character compositions into the same cluster set. The basic idea of the K-means algorithm is as follows.

(i) K domain feature vectors are selected from the dataset as the clustering centers

(ii) For each other domain feature vector, calculate its Euclidean distance [34] from all the cluster centers and assign it to the cluster center with the closest distance

(iii) Update the cluster centers of all cluster sets to the mean value of all domain feature vectors in the cluster set and calculate the squared distance sum $J(C)$ (as shown in Equation (2)) values of all samples to their category cluster centers

(iv) Finally, determine whether the clustering center and $J(C)$ value have changed; if they have, return to the second step to continue the iteration, and vice versa, end the algorithm

$$J(C) = \sum_{k=1}^{K}\sum_{i=1}^{n} d_{ki}\|x_i - y_k\|^2,$$
$$\text{where } d_{ki} = \begin{cases} 1, \text{if } x_i \in c_k \\ 0, \text{if } x_i \notin c_k \end{cases}, \tag{2}$$

where C is the set of clusters, K is the number of clusters, n is the number of sample data, x_i is the data point, and y_k denotes the cluster center of the k_{th} cluster set. Finally, we aggregate the abusive domain names with similar character compositions into K clusters. After that, new abusive domain names are generated based on each cluster.

3.3. Generating New Abusive Domains. The new abusive domain name generation problem can be viewed as a character sequence prediction problem. As described in Section 2.2, LSTM is a special kind of recurrent neural network that has been successful in dealing with machine translation and

sequence problems. However, the algorithm is slow to converge because of its large number of parameters. Therefore, in this paper, GRU is used to build the neural network, which works on the same principle as the LSTM layer but with computational simplifications making the operation less expensive, while the difference in model performance is not significant.

3.3.1. Generating Domains Based on GRU

(1) Building Generation Model

The schematic diagram of the structure of the GRU-based malicious domain name generation neural network is shown in Figure 9.

The first embedding layer accepts an integer domain number vector of definite length as input and will output a meaningful embedding vector of definite length. Each component of the vector consists of floating-point numbers, which can describe the relationship between the characters of the domain name in a specific way. The second layer of the GRU layer accepts the feature vector input of the previous layer and outputs the information of the character of the domain name at the next moment. The third layer of the fully connected layer acts as a classifier, which converts the input of the second layer into a vector of the size of the domain-numbered dictionary and outputs it. Finally, the output vector of the third layer is converted into a logarithmic probability distribution by LogSoftMax and output.

The steps of generating a batch of domain name vectors based on the GRU-based malicious domain name generation neural network are as follows:

(1) First calling the pseudo-random generator to select a batch of first character numbers from the number set corresponding to the set of letters and numbers and forming the corresponding domain number vector, that is, a tensor of 1∗batch-size, and input to the embedding layer, where batch-size is customized by the user during the generation process

(2) The embedding layer converts each input character number into an embedding-dim dimensional embedding vector to obtain the embedding feature in the shape of 1∗batch-size∗embedding-dim and outputs it to the GRU layer. During this experiment, embedding-dim is 128

(3) The GRU layer converts the input of embedding features into a tensor in the shape of 1∗batch-size∗hidden-dim, which contains the information to predict the character number of the next batch, and inputs it to the fully connected layer. Hidden-dim is the size of the hidden layer of the GRU layer. During the experiment, hidden-dim is 256

(4) The fully connected layer converts the input tensor into a tensor in the shape of batch-size∗char-count and outputs it. LogSoftMax converts each line to a probability distribution of the next character number to be generated, where char-count is the size of the domain-numbered dictionary. During this experiment, char-count is 627

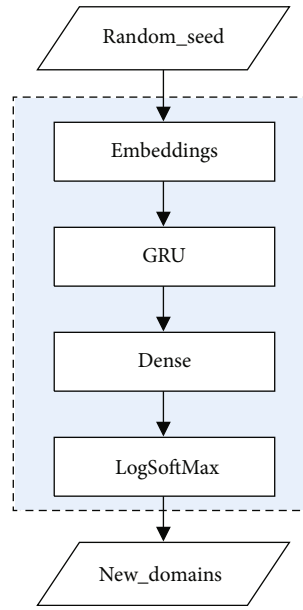


Figure 9: The structure of the GRU-based malicious domain name generation neural network.

(5) For each row, the character number with the highest probability is selected as the number of the next batch of characters generated. The character number of the next batch is used as the new input to the neural network

(6) Repeat steps (2) to (5) until the length of the domain name reaches the maximum length or the generated character number belongs to the top-level domain character numbers, then stop generating. The initial input character number and the domain name number generated in each step form a domain name number vector in the generation order. The domain name number vector is decoded into the corresponding domain name characters to obtain a batch of generated domain name vectors

(2) Training Generation Model

The steps of training GRU-based malicious domain name generation neural networks are as follows:

(1) Read model configuration information and related parameters

(2) Based on the clustering results, all the original domain name data in one of the categories that have not yet participated in model training are read as the model training dataset

(3) The original domain name data is encoded and converted into a domain name number vector, and the length of the domain name number vector is filled

to max-length. During this experiment, max-length is 50

(4) Configure the Adam optimizer to adapt to gradient changes to adjust different learning rates according to parameter changes

(5) Get a batch of training data shaped as max-length∗ batch-size

(6) Each column of training data is a domain name number vector. The first element to the max-length-1 element of each domain name number vector is combined into an input domain vector named input-vector. A batch of input-vector is combined into a tensor in the shape of (max-length-1) ∗ batch-size. The second element to the max-length element of each domain name number vector is combined into a target vector named target-vector, and a batch of target-vector is combined into a target tensor in the shape of (max-length-1) ∗ batch-size

(7) The model is learned and fitted on the input tensor and the target tensor. The loss value between the prediction result of the model under the input tensor and the target tensor is calculated using cross-entropy as the loss function. The trainable parameters are updated by backpropagation. The network parameters are updated by the optimizer

(8) Repeat steps (5) to (7) until the maximum number of iterations is reached or the network loss value stabilizes, and save the network parameters when the loss value is below the set threshold during the training process

(9) Repeat steps (2) to (8) until the original domain name data of all categories in the clustering results are used as the model training dataset to participate in the neural network training

3.3.2. Checking Domains. In this step, we first check if the generated domains are configured with IP addresses. For domains with configured IP addresses, we then try to obtain the web content of these domains. Finally, using the method, we devised (described in Section 3.1) to detect whether these domains are abusive or not.

3.4. Evaluation Metrics. In this paper, we use standard accuracy (*Acc*, Equation (3)), precision (*P*, Equation (4)), recall (*R*, Equation (5)), and F1-score (*F*1, Equation (6)) as the classification evaluation metrics to evaluate the performance of abusive domain name detection. The specific formulas are as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \tag{3}$$

$$P = \frac{TP}{TP + FP}, \tag{4}$$

$$R = \frac{TP}{TP + FN}, \tag{5}$$

$$F1 = \frac{2 * P * R}{P + R}, \tag{6}$$

where *TP* is true positives, *TN* is true negatives, *FN* is false negatives, and *FP* is false positives, respectively. We define the abusive domain names as positive and the benign ones as negative.

As we described above, pornographic and gambling domains detected using the fastText-based classifier are again detected using the CNN-based classifier. Therefore, we use Equation (7) to evaluate its joint precision (P_{joint}).

$$P_{\text{joint}} = 1 - (1 - P_{\text{text}}) * \left(1 - P_{\text{image}}\right), \tag{7}$$

where P_{text} denotes the precision of detecting abusive domains based on text, and P_{image} denotes the precision of detecting abusive domains based on the image.

4. Experimental Results

In this section, we mainly show the abusive domain detection performance and the results of the generated abusive domains.

4.1. Abusive Domain Database

4.1.1. Performance of fastText-Based Detection. In order to evaluate the performance of the fastText-based detection models, we used a 10-fold cross-validation strategy over the dataset. First, the dataset is split into 10 folds, then the seven folds are trained, and the remaining one is used for testing. This process is repeated ten times. Finally, all the metrics on the validation folds are averaged and a better estimate of the performance is achieved. Based on the method introduced in Section 3.1, we obtained the text information of benign and abusive domain names as shown in Table 1.

We trained and tested the model on Apple's Mac mini M1 version (8 CPU cores and 16 GB RAM). Under our normal conditions of using the device, such as opening the browser, PyCharm software, the model training took only 4.3 seconds. In addition, the model took only 1.2 seconds to complete the classification of 62,751 domain names. The values of *P*, *R*, *F*1, and *Acc* of the model in detecting abusive domain names are 0.98, 0.97, 0.98, and 0.96, respectively. Thus, the abusive domain name detection model base on fastText we built has very good performance with high efficiency and low resource consumption.

4.1.2. Performance of CNN-Based Detection. Based on text detection of pornography and gambling domains, we filtered out a large number of pornography and gambling websites. We used the selenium-based crawler to obtain a total of 37,266 web snapshots of pornography and gambling domains. In addition, we obtained 27,132 web snapshots of domains provided by Alexa as benign samples. In the experiment, this paper divides the original domain snapshots into

Table 1: The summary of the training and test datasets for fastText-based detection.

Category	Training	Test
Abusive domains	124,686	53,277
Benign domains	22,193	9,474
Total	146,879	62,751

Table 2: The summary of the training, verification, and test datasets for CNN-based detection.

Category	Training	Verification	Test
Abusive domains	26,085	3,725	7,456
Benign domains	18,992	2,713	5,427
Total	45,077	6,438	12,883

training set, validation set, and test set according to the 7 : 1 : 2. The dataset division is shown in Table 2.

The GPU graphics card used in this experiment is the Quadro GV100 with 32 GB of video memory. TensorFlow (https://www.tensorflow.org, accessed on 6 May 2022), Matplotlib, Keras (https://keras.io, accessed on 6 May 2022), and other services are configured in the operating system (OS) system Ubuntu 20.04.3LTS environment. In this paper, the GPU-accelerated convolutional neural network was built with the Keras framework as the core. The evaluation metrics of model classification include training accuracy, training loss, validation accuracy, and validation loss, where validation accuracy and loss are used to determine whether the model is overfitted during the training iteration and to evaluate the generalization ability of the model. The training accuracy and loss values are used to evaluate the model's performance on the training set. The variation of the classification accuracy and loss values of the convolutional neural network with the number of training epochs during the model training are shown in Figure 10.

As shown in Figure 10, the model's classification accuracy went from 64% to 92% after 26 epochs of training. The accuracy increases rapidly in the first 10 epochs, fluctuates slightly between 11 and 26 rounds, and starts to grow slowly after 26 epochs. This indicates that the training of the model has converged. The trend of the model training loss value in Figure 10 is basically opposite to the trend of the model training accuracy. The loss value of the model decreases from 6.6 to 2.6 after 26 epochs of training. The loss value of the model starts to fluctuate after 26 and basically tends to be constant.

The variation of the validation classification accuracy and the validation loss value of the convolutional neural network with rounds during the model training are shown in Figure 11.

The accuracy and loss trends in Figure 11 are generally consistent with Figure 10. The validation accuracy values in Figure 11 start to fluctuate when the rounds reach 26, indicating that the fit has converged. After 26 epochs of training, the final accuracy increases from 0.77 to 0.91. The validation loss value tends to level off after 26 epochs of training, with only a slight vibration, which indicates the strong generalization ability and high accuracy of the model.

Finally, the experimental results show that the model built in this paper has an accuracy of 0.95 on the training set, 0.90 on the validation set, and 0.91 on the test set. Based on the test set, the model's values of P, R, and $F1$ in detecting abusive domain names are 0.92, 0.93, and 0.92, respectively.

Furthermore, as we introduced in the above section, we use CNN-based identification in the set of pornographic or gambling domains discovered by the fastText-based model. That is, we use the built CNN-based model to filter out benign domains that are misclassified as gambling or pornography from the set of abusive domains in this step. After evaluation, the CNN-based model achieves an accuracy value of 0.98 in detecting benign domains in this abusive domain set. Based on Equation (7), the joint model built in this paper achieves an accuracy (P_{joint}) of 0.99 in detecting pornographic and gambling domains.

4.2. Generating New Abusive Domains

4.2.1. Domain Clustering. When using the K-means algorithm to cluster domain names that are similar in composition, the first step is how to determine the appropriate K value. The elbow method [35] is proposed to explain and verify the consistency of clustering analysis to assist in the determination of the optimal number of clusters in the dataset. The core idea of the elbow method is that if the value of K is much smaller than the optimal number of clusters, as the value of K increases, which will greatly increase the degree of aggregation of each cluster, so the value of $J(C)$ will decrease sharply. When the value of K increases close to the optimal number of clusters, the decrease of the $J(C)$ value slows down as the value of K increases. And when the value of K reaches the optimal number of clusters, continuing to increase the value of K will cause $J(C)$ to level off. Therefore, $J(C)$ decreases sharply and then flattens out as the K value increases, and the optimal K value is the K value at the inflection point.

Therefore, we determine the correct number of clusters according to the elbow method. Its calculation formula is shown in Equation (2). We selected different K values to cluster the abusive domain name dataset while calculating the $J(C)$ values, as shown in Figure 12. It can be seen that when the curve has an inflection point at K =5 or 6.

The silhouette coefficient or silhouette score [36] is a metric used to calculate the goodness of a clustering technique. The silhouette score ranges from -1 to 1. The negative value indicates that the sample is assigned to the wrong set of clusters, and the assignment is not satisfactory. When the value is positive, the larger the value, the smaller the distance between samples of the same category, and the larger between samples of different categories, the better the clustering effect. The expression of the sample silhouette coefficient s is shown in Equation (8).

$$s = \frac{b - a}{\max(a, b)}. \tag{8}$$

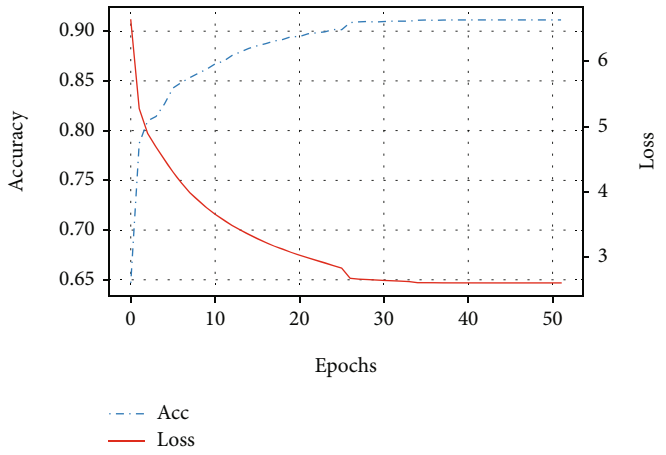


Figure 10: The accuracy and loss of model classification with epochs during model training.

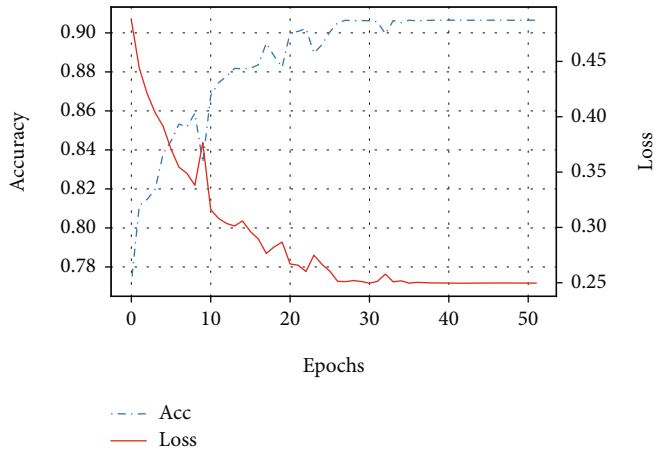


Figure 11: The accuracy and loss of model classification with epochs during model testing.

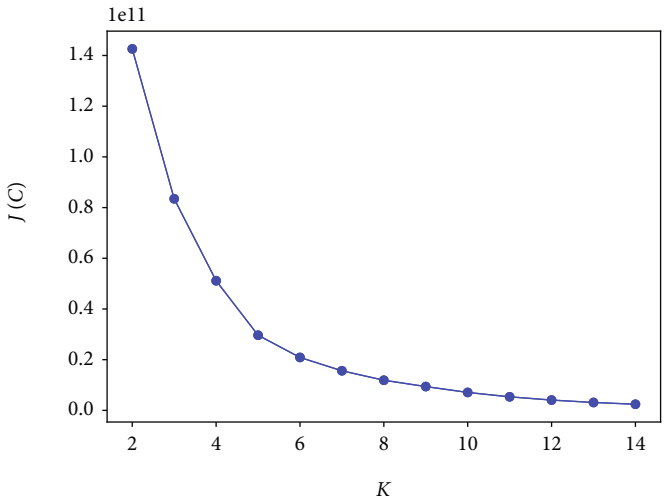


Figure 12: The decline curve of $J(C)$ as the K value increasing.

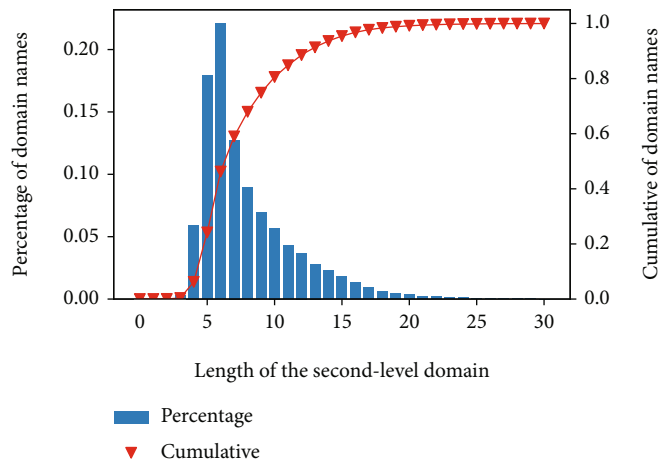


Figure 13: Distribution of second-level domain length.

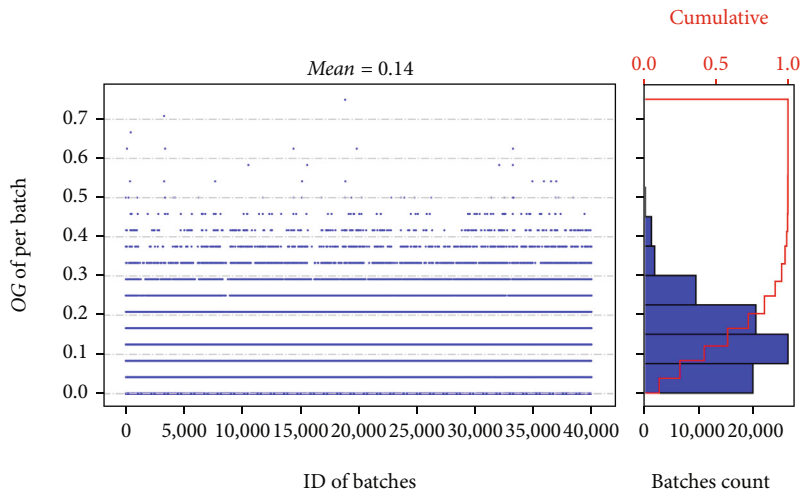


Figure 14: Distribution of new online domain names (2022-03-23).

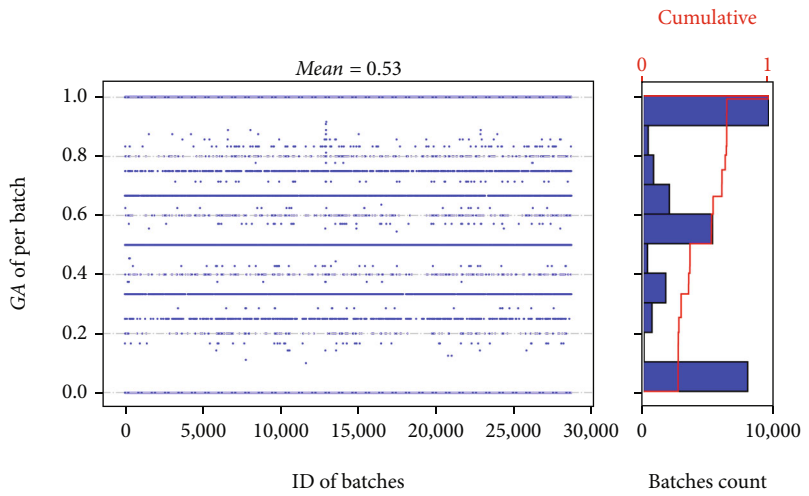


Figure 15: Distribution of new abusive domain names (2022-03-23).

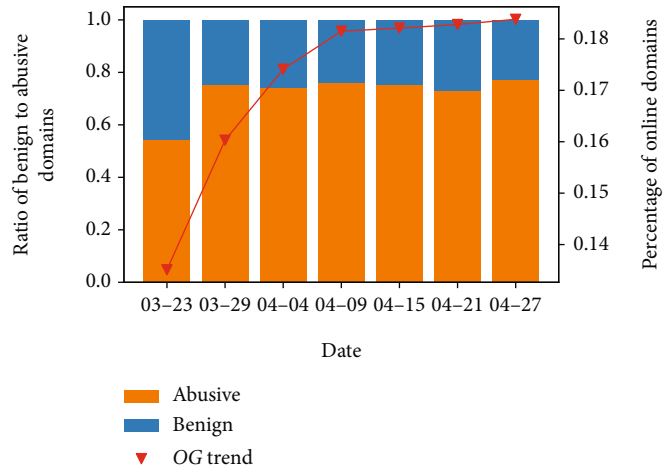


Figure 16: Increasing *GA* and *OG* values for new domain names generated over time.

In Equation (8), a is the average distance between the sample and other samples in the same cluster, and b is the minimum average distance between the sample and not in the same cluster samples.

The silhouette coefficient S of the sample set is the average of the silhouette coefficients of all samples, and its expression is as in Equation (9).

$$S = \frac{\sum_{i=1}^{n} s_i}{n}, \tag{9}$$

where n is the total number of samples, and s_i is the value of the i_{th} sample silhouette coefficient.

Finally, the values of the silhouette coefficients of the sample set are 0.866 and 0.831 when K =5 and 6, respectively. When K=5 indicates that the clustering effect is relatively satisfactory, the abusive domains within the uniform cluster sets are more similar. As a result, we divided the abusive domain names into five categories separately to generate new abusive domain names.

4.2.2. Generating Domain Names. The goal of abusive domain name generation is to generate domains that are as online (with web content) and as abusive as possible. We measure the performance of the generation model in terms of two metrics: the online rate of generating domain names (*OG*, as shown in Equation (10)) and the generation rate of abusive domain names (*GA*, as shown in Equation (11)).

$$OG = \frac{OGD}{GD}, \tag{10}$$

$$GA = \frac{AOD}{OGD}. \tag{11}$$

OG indicates the percentage of domains with web content to these generating domains. *GA* indicates the percentage of online domains that are pornographic or gambling domains. *GD* refers to the total number of generated domains. *OGD* refers to the number of online generated domains. *AOD* is the number of online domains that are abusive.

The neural network in the domain name generation model is built based on PyTorch (https://pytorch.org, accessed on 6 May 2022), and the model is configured and generated as described in Section 3.3. During the training process, the neural network weight parameters with model loss values below 0.3 are saved. In addition, for the abusive domains that have been divided into 5 clusters, 10 neural network weights are saved under each cluster, i.e., we end up with a total of 50 neural network weights.

Then, the keywords that compose the domain names are fed to the trained neural network model, and the model can generate the domain names with the maximum probability of abuse based on those keywords. Therefore, based on the configuration of the domain name generation model, 50 domain names can be generated for each keyword (or called batch), and the same batch does not contain the same domain name.

In the experiments, we determine the length of input keywords based on the length of the domainâ€™s second-level domain (e.g., the domain http://google.com, whose second-level domain is Google). As shown in Figure 13, which shows the distribution of the second-level domain length of benign and abusive domains we collected, it can be seen that more than 95% of the domains have a second-level domain length of less than 15. Therefore, we input 40,000 random strings (batches) (26 letters and ten digits) with a length less than 15 into the domain generation model. Finally, we got a total of 964,112 new domain names.

We use the requests-based crawler (introduced in Section 3.1) to crawl the web content of the domains in order to check whether the generated domains are online or not. The percentage of online domains in each batch of 50 generated domains is shown in Figure 14. We can discover that the average *OG* value is about 0.14, i.e., about 130,220 of the generated 1 million domains are with web content. In addition, the maximum and minimum values of *OG* for the batches of generated domains are 0.68 and 0.06, respectively. We can also find from the cumulative graph that the *OG* for each batch is concentrated between 0.06 and 0.3. This indicates that the number of online domains is related to the keywords entered into the generation model. By

selecting appropriate keywords, the number of online domains can be enhanced.

Next, we analyze how many of the generated online domains are malicious. Figure 15 shows the distribution of the percentage of generated online domains with pornographic or gambling websites in each batch. The average *GA* value was 0.53 on March 23, 2022, which means that about 70,318 domains were pornographic or gambling. Thus, this suggests that the domain name generation model we built can discover a large number of new pornographic and gambling domains in order to expand the list of abusive domains.

In addition, during our detection of all newly generated domains for more than one month, we found that more and more new domain names were gradually coming online, and most of the online domains were pornographic or gambling domains, as shown in Figure 16. The *OG* value of generated new domains improved from 0.14 to 0.18, i.e., about 48,205 domains came online in a month. And the percentage of new online domains that are abusive increased from 0.54 to 0.78, which indicates that a large number of pornographic or gambling domains came online over time and then spread malicious information. The experiments show that the domain generation model used in this paper can find a lot of pornographic and gambling domains in advance.

To summarize, we entered 40,000 keywords into the domain generation model and generated a total of 964,112 unique domain names. Eighteen percent of these domains (177,217 domains) serve web pages, with 127,596 domains serving pornographic or gambling sites. It turns out that with this domain name generation model, we can get a lot of new pornographic and gambling domain names with the same composition as the old domains.

5. Conclusion

The first step in blocking and dealing with pornographic and gambling domains is to discover them. The more quickly, precisely, and early these abusive domains are handled, the more harm they cause to people, particularly children and minors, can be mitigated. In this paper, we developed a two-layer detection system to quickly and precisely detect pornography and gambling domains using fastText and CNN models. In particular, in order to discover more abusive domains earlier, we proposed a domain generation model based on GRU for rapidly generating new abusive domain names from known ones. The experimental results demonstrate that our domain name detection and generation model is capable of discovering a large number of pornographic and gambling domains.

Moreover, it should be noted that the number of new gambling and pornographic domains that can be generated is more related to the sample of already existing pornographic and gambling domains. The larger the number of domains in the sample, and the more domains with similar character composition, the better the generated domains. Therefore, the limitation of this paper mainly comes from the number and quality of the sample domain names.

In the future, first, we should detect DGA domains using the idea of text-based generation and find domains generated using the same algorithm. Second, we should study the relationship between different keywords (length and composition) and the generated domain names that are more likely to be abusive. Finally, we would like to apply the detection techniques of this paper for pornography and gambling domains to discover phishing attack activities. Also, we would like to discover potential attacks by generating domain names that are similar to the target domain (e.g., http://google.com).

Abbreviations

CNN: Convolutional neural network
DGA: Domain generation algorithm
LSTM: Long-Short-Term Memory
GRU: Gated recurrent units
NL: Natural language processing
FQDN: Fully qualified domain name
DNS: Domain name system
TLD: Top-level domain
TP: True positive
TN: True negative
FN: False negative
FP: False positive
OG: Online rate of generating domains
GA: Generation rate of abusive domains
RNN: Recurrent neural network.

Data Availability

https://reurl.cc/0p27db, access password: nist@HIT.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Y.C. and Z.Z. contributed to the conceptualization; Y.C. and H.J. contributed to the methodology; Y.C. and H.J. were responsible for the software; Y.C. and H.J. contributed to the validation; T.C. contributed to the formal analysis; Y.C. contributed to the investigation; Y.C. was responsible for the resources; T.C. contributed to the data curation; Y.C. and H.J. contributed to the writing—original draft preparation; Y.C. and T.C. contributed to the writing—review and editing; Y.C. and T.C. contributed to the visualization; Z.Z. and Y.D. contributed to the project administration. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was funded by the Natural Science Foundation of Shandong Province [Grant No. ZR2020KF009] and the Young Teacher Development Fund of Harbin Institute of Technology [Grant No. IDGA10002081].

References

[1] H. Yang, K. Du, and Y. Zhang, "Casino royale: a deep exploration of illegal online gambling," in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 500–513, San Juan, PR, USA, December 2019.

[2] D. C. Hodgins and R. M. Stevens, "The impact of COVID-19 on gambling and gambling disorder: emerging data," *Current Opinion in Psychiatry*, vol. 34, no. 4, pp. 332–343, 2021.

[3] A. Hakansson, F. Fernandez-Aranda, and J. M. Menchon, "Gambling during the COVID-19 crisis – a cause for concern," *Journal of Addiction Medicine*, vol. 14, no. 4, pp. e10–e12, 2020.

[4] "Pornography is booming during the covid-19 lockdowns," May 2022, https://www.economist.com/international/2020/05/10/pornography-is-booming-during-the-covid-19-lockdowns.

[5] H. A. Awan, A. Aamir, M. N. Diwan et al., "Internet and pornography use during the COVID-19 pandemic: presumed impact and what can be done," *Frontiers in Psychiatry*, vol. 12, p. 220, 2021.

[6] V. Cerdan Martinez, D. Villa-Gracia, and N. Deza, "Pornhub searches during the Covid-19 pandemic," *Porn Studies*, vol. 8, no. 3, pp. 258–269, 2021.

[7] M. Brodeur, S. Audette-Chapdelaine, A. C. Savard, and S. Kairouz, "Gambling and the COVID-19 pandemic: a scoping review," *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, vol. 111, article 110389, 2021.

[8] Y. Cheng, Y. Liu, L. Wang, Z. Zhang, T. Chai, and Y. Du, "Evaluating the effectiveness of handling abusive domain names by internet entities," *Electronics*, vol. 11, no. 8, p. 1172, 2022.

[9] R. Yang, X. Wang, and C. Chi, "Scalable detection of promotional website defacements in Black Hat SEO campaigns," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*, pp. 3703–3720, Vancouver, B.C., Canada, 2021.

[10] M. Stamp, M. Alazab, and A. Shalaginov, *Malware Analysis Using Artificial Intelligence and Deep Learning*, Springer, Berlin/Heidelberg, Germany, 2021.

[11] P. Lison and V. Mavroeidis, "Automatic detection of malware-generated domains with recurrent neural models," 2017, https://arxiv.org/abs/1709.07102.

[12] R. R. Curtin, A. B. Gardner, and S. Grzonkowski, "Detecting DGA domains with recurrent neural networks and side information," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pp. 1–10, Canterbury, CA, United Kingdom, 2019.

[13] C. Xu, J. Shen, and X. Du, "Detection method of domain names generated by DGAs based on semantic representation and deep neural network," *Computers & Security*, vol. 85, pp. 77–88, 2019.

[14] B. Bharathi and J. Bhuvana, "Domain name detection and classification using deep neural networks," in *Proceedings of the International Symposium on Security in Computing and Communication*, pp. 678–686, Bangalore, India, 2018.

[15] F. Ren, Z. Jiang, and J. Liu, "Integrating an attention mechanism and deep neural network for detection of DGA domain names," in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 848–855, Portland, OR, USA, November 2019.

[16] Y. Chen, R. Zheng, A. Zhou, S. Liao, and L. Liu, "Automatic detection of pornographic and gambling websites based on visual and textual content using a decision mechanism," *Sensors*, vol. 20, no. 14, p. 3989, 2020.

[17] J. Zhao, M. Shao, H. Peng, H. Wang, B. Li, and X. Liu, "Porn2-Vec: a robust framework for detecting pornographic websites based on contrastive learning," *Knowledge-Based Systems*, vol. 228, article 107296, 2021.

[18] Q. Wang, T. Luo, and D. Wang, "Chinese song iambics generation with neural attention-based model," 2016, https://arxiv.org/abs/1604.06274.

[19] B. L. Sturm, J. F. Santos, and O. Ben-Tal, "Music transcription modelling and composition using deep learning," 2016, https://arxiv.org/abs/1604.08723.

[20] Y. Luo and Y. Huang, "Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry," in *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, pp. 99–104, New York, NY, USA, June 2017.

[21] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Piscataway, NJ, USA, November 2016.

[22] Y. Liu, D. Liu, and J. Lv, "Deep poetry: a Chinese classical poetry generation system," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13626-13627, New York, NY, USA, February 2020.

[23] A. Bartoli, A. De Lorenzo, and E. Medvet, ""' Best dinner ever!!!": automatic generation of restaurant reviews with LSTM-RNN," in *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 721–724, Omaha, NE, USA, October 2016.

[24] Y. Cao, R. Ji, L. Ji, G. Lei, H. Wang, and X. Shao, "l^2-MPTCP: a learning-driven latency-aware multipath transport scheme for industrial Internet applications," *IEEE Transactions on Industrial Informatics*, p. 1, 2022.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] "Download list of all domains," May 2022, https://domains-monitor.com.

[27] "Requests: HTTP for humans," May 2022, https://docs.python-requests.org/en/latest.

[28] "Selenium," May 2022, https://www.selenium.dev.

[29] "fasttext Â·PyPI," May 2022, https://pypi.org/project/fasttext/.

[30] "Alexa ranking _ Website traffic worldwide ranking _ Chinese website ranking," May 2022, http://www.alexa.cn/.

[31] "reporting_abusive_domains/abusive_keywords.txt at main Â·mrcheng0910/reporting_abusive_domains," May 2022, https://github.com/mrcheng0910/reporting_abusive_domains/blob/main/abusive_keywords.txt.

[32] "fxsjy/jieba: Jieba Chinese word segmentation," May 2022, https://github.com/fxsjy/jieba.

[33] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[34] "Euclidean distance - Wikipedia," May 2022, https://en.wikipedia.org/wiki/Euclidean_distance.

[35] F. Liu and Y. Deng, "Determine the number of unknown targets in Open World based on Elbow Method," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 986–995, 2021.

[36] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, pp. 25–71, Springer, Berlin, Heidelberg, Germany, 2006.