applied sciences

MDPI

Article

Illegal Domain Name Generation Algorithm Based on Character Similarity of Domain Name Structure

Yuchen Liang, Yanan Cheng, Zhaoxin Zhang *, Tingting Chai * and Chao Li

Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China
* Correspondence: zhangzhaoxin@hit.edu.cn (Z.Z.); ttchai@hit.edu.cn (T.C.)

Abstract: Detecting and controlling illegal websites (gambling and pornography sites) through illegal domain names has been an unsolved problem. Therefore, how to mine and discover potential illegal domain names in advance has become a current research hotspot. This paper studies a method of generating illegal domain names based on the character similarity of domain name structure. Firstly, the K-means algorithm classified illegal domain names with similar structures. Then, put the classified clusters into the adversarial generative network for training. Finally, through a specific result verification method, the experiment shows that the average concentration of the generation algorithm is 23.82%, the effective concentration is 63.54%, and the expansion rate is 7.5. By comparing the results with the enumeration algorithm, the generation algorithm has greatly improved in terms of generation efficiency and accuracy.

Keywords: illegal domain names; K-means; generation algorithm; adversarial generative network; enumeration algorithm

Citation: Liang, Y.; Cheng, Y.; Zhang, Z.; Chai, T.; Li, C. Illegal Domain Name Generation Algorithm Based on Character Similarity of Domain Name Structure. *Appl. Sci.* **2023**, *13*, 4061. https://doi.org/10.3390/app13064061

Academic Editor: Adamu I. Abubakar

Received: 21 February 2023
Revised: 13 March 2023
Accepted: 17 March 2023
Published: 22 March 2023

1. Introduction

According to the "Facts and Figures" [1] released by the ITU, as of December 2021, 50.8% of Internet users have encountered harassment from a series of illegal websites, such as illegal gambling and pornographic websites. Illegal websites bring huge property losses, serious privacy leaks, and mental torture to many netizens.

The domain name is one of the essential identity characteristics of the website. Currently, illegal domain name mining is not accurate and efficient. Therefore, the primary purpose of this paper is to study a method for generating illegal domain names through illegal domain name structures.

This comprehensive generation algorithm aims to improve the number and efficiency of illegal domain name generation. Firstly, this paper uses the K-means algorithm to cluster the illegal domain names with similar structures to generate various illegal domain name clusters. Since the number of illegal domain names exceeds 10,000, this study optimizes the K-means algorithm, improving classification accuracy and efficiency. Finally, bring these clusters into a sequence confrontation generation network (SeqGAN) to obtain a batch of potential illegal domain names. The experiment shows that the average concentration of the generation algorithm is 23.82%, the effective concentration is 63.54%, and the expansion rate is 7.5. The following sections will introduce the related research, experimental design, and experimental results.

The domain name is one of the essential identity characteristics of the website. The main goal of our research is to use the characteristics of domain names to find undetected illegal websites. Currently, illegal domain name mining is not accurate and efficient. Therefore, the primary purpose of this paper is to study a method for generating illegal domain names accurately and efficiently through illegal domain name structures.

2. Related Research

In terms of websites, Luca Invernizzi et al. [2] proposed a method for effectively searching illegal web pages by analyzing the redirection behavior between illegal web pages and applying the heuristic search algorithm to web crawlers. Wang Qingguang et al. [3] actively discovered and identified pornographic, violent, reactionary, and other harmful websites in the network according to the link relationship between illegal webpages, combined with web crawler technology and content security filtering technology.

In terms of DNS, Sato et al. [4] observed DNS data and found that illegal domain names belonging to the same malware family were often resolved by the same host at the same time, and thus proposed an evaluation method based on the co-occurrence relationship of domain names. The degree of co-occurrence of listed domains with unknown domains, discovering new illegal domains that are not on the blacklist. Guerid et al. [5] found that botnets tend to have a community structure by analyzing DNS traffic, thus proposed a new botnet discovery method, identifying bot communities composed of hosts with similar malicious behaviors through DNS traffic, and correlating each community traffic to identify malicious servers controlling these hosts. Aiming at the unrobust problem of using local features of DNS data, Issa Khalil et al. [6] proposed a complementary method to discover and analyze the global association between domains, establish meaningful associations between domains, and infer whether unknown domains are illegal.

In terms of domain name generation, after a lot of research, domain name structure is one of the critical factors affecting the effectiveness and accuracy of domain name generation. Considering the factors of domain name structure, the primary process of illegal domain name generation includes data collection, structure analysis, classification, and generation of several steps. Yuan Chen et al. [7] proposed a method to generate DGA variant samples for the illegal domain name data collection and structure analysis process. This method defines the domain name encoder and decoder based on the ASCII encoding method, trains the generative confrontation network to construct the domain name character generator, and thus predicts the DGA variant samples belonging to the same family. Regarding the illegal domain name classification, Yanan Cheng et al. [8] analyzed illegal domain name data sets, customized illegal domain name similarity rules, and classified illegal domain names using the structural similarity between illegal domain names. In terms of illegal domain name generation, Yanan Cheng et al. [9] proposed a method to proactively discover illegal domain names. Samuel Marchal et al. [10] use natural language modeling techniques to build a proactive blacklist and effectively learn illegal domain names in the network. Bi Xiaotao [11] by extracting 56 data features in five categories, including domain name text and lexical grammar, proposed a data optimization method based on the degree of feature anomaly and constructed a variety of differentiated unlabeled real-network DNS traffic data sets. Aiming at the slow detection speed of mainstream illegal domain name detection algorithms, Zhang Weiwei et al. [12] proposed a lightweight domain name detection algorithm based on morpheme features. According to the structural characteristics of illegal domain names, only mining domain name morpheme features can quickly detect illegal domain names.

According to relevant research, there are many ways to discover illegal domain names. There are also studies on the discovery and generation of illegal domain names based on domain name structure. In terms of generation efficiency and accuracy, the results are not ideal. Furthermore, the existing illegal domain name generation methods only generate specific types of illegal domain names and cannot be applied to other kinds of illegal domain name generation. Therefore, this paper designs and implements a set of illegal domain name generation algorithms based on similar names, which are applied to generate various types of illegal domain names. The comparison of existing approaches and proposed approaches was shown as Table 1.

Table 1. Comparison of existing approaches and proposed approaches.

No.	Improvement Direction	Existing Approaches	Proposed Approaches
1	Mining for different types of illegal domain names	The existing illegal domain name mining method is only for the specific illegal type of domain name mining.	The illegal domain names with similar names are gathered in the same cluster by clustering. Then different clusters are trained so that illegal domain names with different characteristics can be better mined.
2	The coverage of illegal domain names mined is not broad enough	The obtained illegal domain name is limited to a certain area of the domain name space. Therefore, global information about illegal domain names cannot be obtained. Therefore, the obtained illegal domain name cannot be reused and further mined.	The potential illegal domain names scattered in many corners of domain name space can be mined effectively by using the adversarial generation network method to obtain more comprehensive illegal domain names.
3	Algorithm efficiency	In the existing illegal domain name enumeration algorithm, although the number of generations is large, the effective concentration is low, and the generation time is too long.	The clustering algorithm is used to form different types of illegal domain name clusters so as to reduce the generation of invalid domain names in the generation process. At the same time, the adversarial generation network is used to reduce the gradient disappearance or gradient explosion caused by feature extraction and thus reduce the training time.

3. Algorithm Design and Process

To solve the problem that the existing illegal domain name generation algorithms are only practical for specific types, this chapter designs a comprehensive generation algorithm. The algorithm realizes the generation of new domain names from unknown space through domain name structure similarity.

This algorithm includes two stages: illegal domain name clustering and illegal domain name generating. The algorithm first uses the Mini-Batch K-means algorithm to cluster illegal domain names with high feature similarities. Then use these clusters as the training data to design and train SeqGAN. Finally, a batch of potentially illegal domain names can be obtained by bringing actual data into the generated SeqGAN model. This is shown in Figure 1.

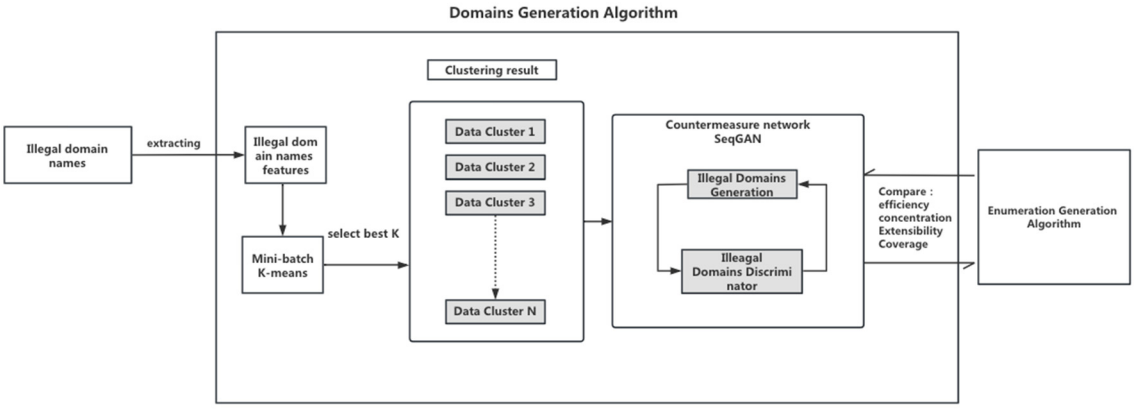

Figure 1. Generation algorithms process.

This chapter discusses the implementation process of the illegal domain name generation algorithm. Furthermore, sets the enumeration method as the control group to verify the effectiveness of this generation algorithm.

3.1. Domain Name Coding and Decoding Method

To convert the domain name and the input and output forms accepted by the network, the encoding and decoding method of the domain name is designed according to the structural characteristics of the domain name. The specific implementation steps are as follows:

(1) Coding method

Encoding is the process of converting a domain name into a network-acceptable domain vector form. Each domain name level consists of letters a~z, numbers 0~9, and a hyphen -. All the top-level domains and empty characters appearing in the data set form a domain name character table, and then number each character in the table. Complete the domain name with empty characters to the specified length, and convert all the characters in the completed domain name into character numbers to obtain a domain name vector composed of character numbers.

(2) Decoding method

Decoding is converting the domain name vector output by the generator into a domain name. Find the corresponding character in the domain name character table according to the character number in the domain name vector, obtain a domain name string of a specified length, and remove the null character at the end of the string to obtain the domain name.

3.2. Illegal Domain Name Clustering

Illegal domain names of the same type have similarities in structure. To divide illegal domain names belonging to different name types, the Mini-Batch K-means algorithm [13] is a practical choice for clustering illegal domain names. Mini-Batch K-means is a derivative algorithm of K-means, which usually appears in large-scale data of more than 10,000 [14].

Extracting the features of illegal domain names is an essential step before clustering. This experiment uses different methods to extract top-level and second-level domain name features.

- Second-level Domain Name Characterization

For the second-level domain, string s obtains all substrings with a length of n, and counts the frequency of the substrings. After removing substrings with low frequency, the remaining substrings construct into a substring list S. Initialize a zero vector $\vec{Z}$ of the length of the substring list, and set the i^{th} component Z_i in the vector $\vec{Z}$. Z_i is the number of occurrences of the i^{th} substring (i^{th} substring $\in$ S) in the string s. The vector $\vec{Z}$ is the n-gram feature of the string s. After many tests, the 3-g feature can cause the best clustering effect.

- Top-level Domain Name Characterization

Extract all top-level domain names to form a top-level domain list T after deduplication. For the top-level domain name t, its feature vector $\vec{Z}$ $\left(length[\vec{Z}] = length[T]\right)$ is initialized as a zero vector. Set Z_i to 1 and the other components to 0. ($t = T[i]$) $\vec{Z}$ is the feature vector of t which is encoded by one-hot.

The clustering purpose of the Mini-Batch K-means algorithm is to find the set C of cluster centers by minimizing the objective function. Its calculation formula is shown in Formula (1):

$$I = \sum_{x \in T} \|f(C, x) - x\|^2 \quad (1)$$

In the formula, means to return the cluster center $c \in C$ closest to the feature vector of the illegal domain name.

Euclidean distance is used as the distance calculation formula between domain name feature vectors.

$$d_{ij} = \sqrt{\sum_{l=1}^{m} (x_i^l - x_j^l)^2} \quad (2)$$

In the formula, d_{ij} represents the Euclidean distance between the i^{th} domain name feature vector and the j^{th} domain name feature vector.

The optimal K value can be selected by reference to the sum of the squares of the distance between the sample and the nearest cluster center. In this chapter, the elbow method is used to select the optimal number of clusters K. The core idea is that as the number of clusters K increases when K is smaller than the real number of clusters, the increase in K will greatly increase. Increase the degree of aggregation of each cluster, so the sum of the squares I of the distance between the sample and the nearest cluster center will decrease greatly, and the K value I will continue to increase, and the K value will be the best when I is the smallest.

$$I = \sum_{i=1}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right) \quad (3)$$

In the formula, I is the sum of the squares of the distance between the sample and the nearest cluster center, n is the total number of samples, x_i is the feature vector of the i sample, and μ_j is the cluster center of cluster C.

The basic steps of the Mini-Batch K-means algorithm are as follows:

(1) Randomly extract a fixed-size collection of illegal domain names from the data set T to form a small batch and cluster the small batches through the K-means algorithm to construct initial K clusters;
(2) Continue randomly extracting a fixed-size set of illegal domain names from the data set T to form a small batch;
(3) For each illegal domain named in the small batch, calculate the Euclidean distance between d and each cluster center, assign d to the nearest cluster C, and calculate the mean value of d and other illegal domain names d in C to update the cluster C class center;
(4) Use the elbow method to select the optimal number of clusters K for this algorithm;
(5) Steps (2) and (4) are iterated in a loop until the center point is stable or the number of iterations is reached, and the calculation operation is stopped.

3.3. Illegal Domain Name Generating

The problem of illegal domain name generating can be viewed as a sequence prediction problem. This paper uses the SeqGAN [15] to train the generation model. The structural diagram of the SeqGAN is shown in Figure 2.

SeqGAN is a model generator as stochastic policies in reinforcement learning. It uses Monte Carlo search to pass the reward signal output by the discriminator back to the intermediate state step. This method can bypass the generator differentiation problem and directly execute the gradient update strategy so that the network can generate discrete sequences well. Therefore, SeqGAN generates illegal domain names that exist in the unknown domain name space. The SeqGAN in this section comprises a generator based on the gated recurrent unit (GRU) layer and a discriminator based on the character-level language model [16]. The generator aims to generate fake samples close to real samples, and GRU is a variant network of the recurrent neural network [17] (RNN). Adding a GRU layer can improve the sequence prediction ability of the generator. The discriminator aims to identify whether an input sample is a real sample or not. The character-level language model uses the convolutional neural network (CNN) to extract the features of the sequence at the character level to achieve classification, and the use of the character-level language model can improve the ability of the discriminator to classify language samples.

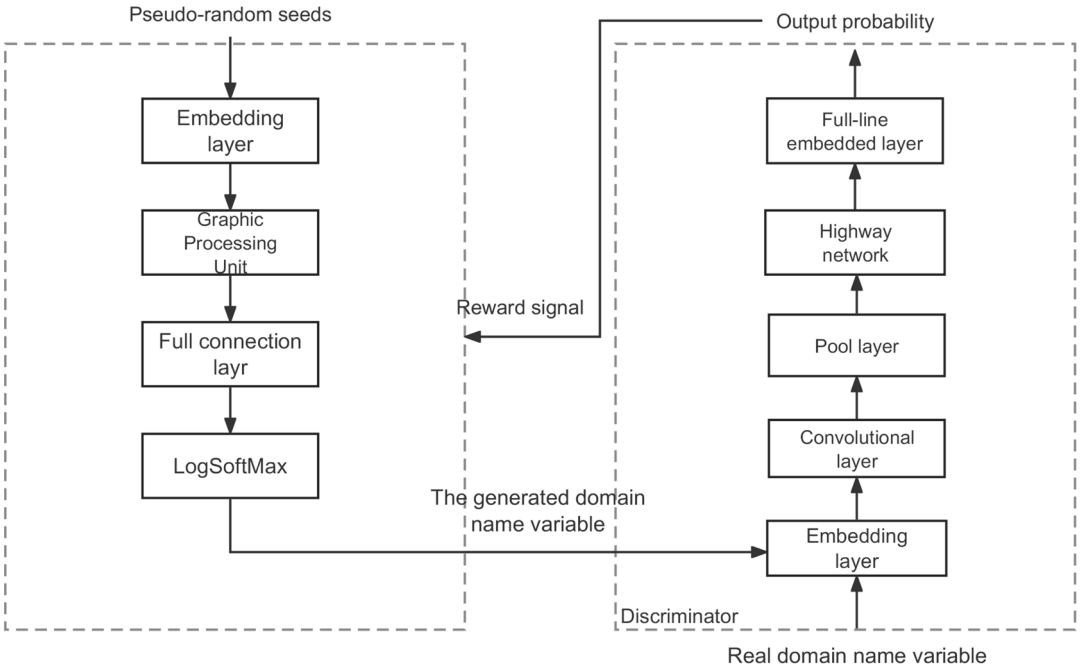

Figure 2. SeqGAN structure.

SeqGAN is a brand-new network born from the combination of GAN and reinforcement learning ideas. GAN has two problems when dealing with discrete sequences. One is that the loss gradient obtained by the discriminator cannot effectively update the generator's parameters, and the other is that it cannot evaluate the quality of the current vocabulary and its impact on subsequent vocabulary generation. SeqGAN solves the first problem by introducing the idea of Policy Gradient, using the reward of each step as feedback to update the generator parameters, bypassing the gradient calculation. The Monte Carlo search method is used to obtain the reward of the generated sequence at each step, which solves the second problem.

The goal of SeqGAN is to generate a sequence close to the sample's true distribution, and its generator's goal is to maximize the expectation of the reward obtained by generating the sequence. The expression of the objective function is shown in Formula (4):

$$J(\theta) = \sum_{y_t \in v} G_\theta(y_t | Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) \tag{4}$$

G_θ: Generator;
$D_\varnothing$: Discriminator;
V: Vocabulary;
y_t: Word generated in the t^{th} step;
$Y_{1:t-1}$: Sequence formed by the words generated in the previous t − 1 step.
$G_\theta(Y_t - Y_{1:T-1})$: Probability value of generating the t^{th} word as y_t when the previous t − 1 words have been generated.
$Q_{D_\varnothing}^{G_\theta}(Y_{1:t-1}, y_1)$: Existing sequence $Y_{1:t-1}$.

$$Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n), Y_{1:T}^n \in MC^{G_\theta}(Y_{1:t}; N) & for\ t < T \\ D_\phi(Y_{1:t}) & for\ t = T \end{cases} \tag{5}$$

For the generated sequence $Y_{1:t-1}$, if the next word y_t is the last word in the sequence, the output probability of the discriminator for the input sequence $Y_{1:t}$ is $Q_{D_\varnothing}^{G_\theta}(Y_{1:t-1}, y_1)$. Otherwise, start sampling from the next position of the current position through a Monte Carlo search, complete the content after y_t, and obtain all possible sequences $Y_{1:t}$, calcu-

late the output probabilities of these complete sequences through the discriminator and calculate the average as $Q_{D_\varnothing}^{G_\theta}(Y_{1:t-1}, y_1)$. See Formula (5) for the calculation:

N: Number of completed complete sequences.
$D_\varnothing$: Discriminator.
G_θ: Generator.
T: Length of the complete sequence.
$Y_{1:t}^n$: Complete sequence at the n^{th} completion.
$MC^{G_\theta}(Y_{1:t}, N)$: Possible complete sequences obtained by Monte Carlo search.
$Y_{1:T-1}$: Sequence of words generated in the previous t − 1 steps.

3.3.1. Generator Design

The generator [18] is designed based on the GRU layer and consists of a three-layer neural network, including an embedding layer with a dimension of 32, a GRU layer with an initial hidden state of 32, and a fully connected layer whose output dimension is the size of a domain name character list.

The generator aims to generate domain name vectors close to real domain name vectors so that the discriminator cannot distinguish real domain name vectors from generated domain name vectors. The first layer of the embedding layer accepts an integer domain name character number input and outputs a fixed-size and meaningful embedding vector. Each vector component is a constant real value, describing the domain name from different aspects. The second layer of GRU is a variant network of recurrent neural networks (RNN), which can effectively solve the problem of long-distance dependence that RNNs cannot handle. At the same time, it accepts the input of the feature vector at the last moment and outputs the character information of the domain name at the next moment. The third fully connected layer converts the output of the second layer into a vector output of the size of the domain name character table; finally, the actual number vector output by the third layer is converted into a logarithmic probability distribution output through LogSoftMax [19].

The structure diagram of the generator is shown in Figure 3.

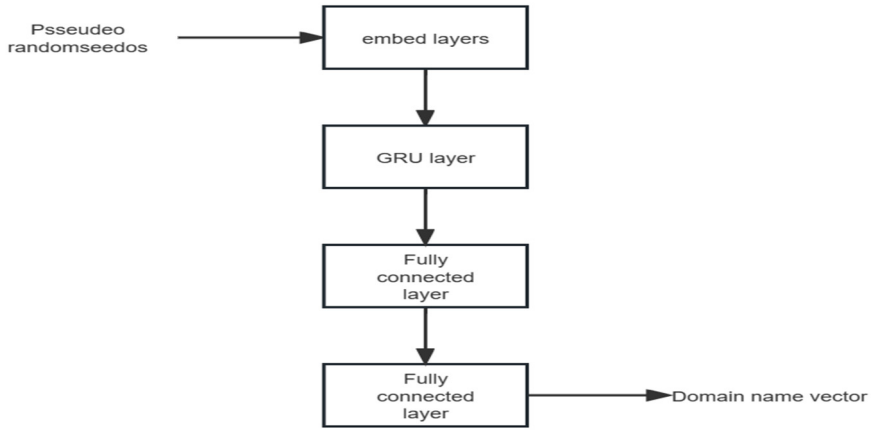

Figure 3. Generator design.

The specific steps for the generator to generate a batch of domain name vectors are as follows:

(1) First, a batch of initial character numbers is selected from the set of numbers corresponding to letters and numbers by pseudo-random seeds, and the batch size is batch_size, which is input to the embedding layer;
(2) The GRU layer obtains a 1 × batch_size × 32 tensor according to the input embedding feature, which contains information to predict the next batch of character numbers, and outputs it to the fully connected layer;

(3) The fully connected layer and LogSoftMax convert the above tensor into batch_size × num_chars output, take the index and convert it into the probability distribution of the next batch of character numbers, where num_chars num_chars is the size of the domain name character table;
(4) Each line of the above output is the probability distribution of the next character number, and the character number with the highest probability in each line is selected as the next character number to obtain the next batch of character numbers. This batch of character numbers is used as the new input of the generator;
(5) Repeat steps (2)~(5) until the current character number is the last character number of the specified domain name length, arrange the character numbers generated in each step in the order of generation to form a domain name vector, and obtain a batch of generated domain name vectors.

3.3.2. Discriminator Design

The discriminator is designed based on a character-level language model. The model first extracts the character-level features of the domain name through a convolution and pooling combination layer containing 20 2 × 2 filters and 10 3 × 3 filters [20], where the output dimension of the convolution layer is 32. Then a two-layer highway network is used to alleviate the problem of gradient disappearance or gradient explosion caused by feature extraction and deepening of the network. Finally, the classification of true and false domain name vectors is realized by two layers of fully connected layers.

The structural diagram of the discriminator is shown in Figure 4.

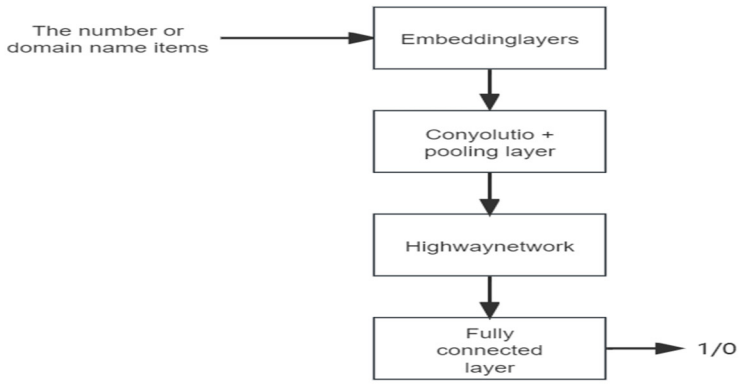

Figure 4. Structure diagram of discriminator.

The discriminator aims to distinguish whether or not the input domain name vector is a real domain name vector. The process of the discriminator to discriminate a batch of input domain name vectors is as follows:

(1) First input a batch of domain name vectors to the embedding layer, and obtain a batch_size × seq_len × 32 embedding tensor, where batch_size is the batch size, and seq_len is the maximum length of the domain name;
(2) The above-embedded tensor extracts the 2-g and 3-g features of the domain name through the convolution and pooling combination layer and outputs aleatch_size × num_features × 1 feature tensor, where num_features is the number of extracted features;
(3) Then slow down the gradient problem of the deep neural network through the highway network, and output a batch of domain name feature vectors;
(4) Finally, the domain name feature vector fully connected layer is used to obtain the probability output of this batch of domain name vectors. When the probability value exceeds 0.5, it is a real domain name vector. Otherwise, it is determined as a generated domain name vector to realize the identification of the authenticity of this batch of domain name vectors.

3.3.3. SeqGAN Training

The clusters of illegal domain names obtained by illegal domain names clustering are used as experimental data. Divide 50,000 illegal domain names into 40,000 training data and 10,000 test data. The training data are used to train the generator and the discriminator to make it work better. The test data only show the change in the discriminator's discrimination ability during the training process and does not participate in the network training.

The basic steps of sequence confrontation generative network training are as follows:

(1) Initialize the generator and discriminator parameters;
(2) Minimize the maximum likelihood loss pre-training generator by Adam optimizer. Use the generator to generate some fake domain name vectors mixed with real domain name vectors and input them into the discriminator, and minimize the cross entropy through the AdaGrad optimizer to pre-train the discriminator;
(3) Use the generator to generate some fake domain name vectors and mix them with the actual domain name vectors and input them into the discriminator to obtain reward signals for these domain name vectors;
(4) Use Monte Carlo search to pass the reward signal output by the discriminator back to the intermediate state step to calculate the loss of these domain name vectors generated by the generator and update the network parameters of the generator through the Adam optimizer;
(5) Use the generator to generate fake domain name vectors, mix them with real domain name vectors, and input them into the discriminator. Using cross entropy as the loss function, calculate the loss of the discriminator to identify these domain name vectors, and update the network parameters of the discriminator through the AdaGrad optimizer;
(6) Repeat steps (3)~(5) until the maximum number of iterations is reached or the sequence confrontation network tends to be stable.

3.4. Illegal Domain Name Enumeration

The enumeration method is used as a comparison method in this chapter to evaluate the effectiveness of the generation algorithm.

The enumeration method introduces wildcards and indicators to define characters or strings in illegal domain names. The wildcards include the top-level domain wildcard "@" (which can enumerate the top-level domains in the top-level domain list), the alphabetic wildcard "*" (which can enumerate 26 lowercase letters), and the numeric wildcard "#" (which can enumerate the numbers 0~9), the valid character wildcard "$" (a~z0~9- can be enumerated). Indicators include the addition indicator "+" (indicating that a domain name inserts a character at any position in the second-level domain), the deletion indicator "-" (indicating that a character is deleted at any position in the second-level domain), and the continuation indicator "&" (indicating consecutive enumerations of the same character).

The domain name d is composed of the second-level domain p and the top-level domain t, p^i represents the i^{th} character of the second-level domain, and p^{ij} represents the substring formed from the i^{th} position to the j^{th} position in the second-level domain.

(1) If $t_1 \neq t_2$ and $p_1 = p_2$, then d_1 and d_2 are similar.
(2) If $t_1 = t_2$, length [p_1] = length [p_2] and $p_1^{ij} \neq p_2^{ij}$, each character of p^{ij} is the same, and $j - i + 1$ does not exceed 3, then d_1 and d_2 are similar.
(3) If $t_1 = t_2$, length [p_1] = length [p_2], and the number of characters in the corresponding positions is not more than 2, then d_1 and d_2 are similar.
(4) If $t_1 = t_2$, the lengths of the second-level domains differ by one, and they are the same after one editing operation, then d_1 and d_2 are similar.

3.5. Results Comparison Method

To better compare the effects of the illegal domain name generation algorithm and the enumeration method, the experiment introduces concentration and extension [21] to measure the pros and cons of the algorithm.

N: The number of initial illegal domain names;

M : The number of domain names that are generated through the illegal domain name generation algorithm;

V : The number of valid domain names;

I: The number of illegal domain names;

C: The concentration;

C': The effective concentration;

D: The expansion degree;

S: The sample silhouette coefficient;

Accuracy: Accuracy refers to the proportion of correctly classified samples in all samples.

Concentration refers to the proportion of illegal domain names in the generated domain names, reflecting the algorithm's generation effect. The calculation of the concentration is shown in Formula (6):

$$C = \frac{I}{M} \tag{6}$$

Effective concentration refers to the proportion of illegal domain names in the generated valid domain names. See Formula (7) for the calculation of effective concentration:

$$C' = \frac{I}{V} \tag{7}$$

The degree of expansion refers to the average number of new illegal domain names that can be obtained based on the existing illegal domain name. See Formula (8) for the calculation of expansion degree:

$$\mathrm{D} = \frac{I}{N} \tag{8}$$

The calculation of the sample silhouette coefficient is shown in Formula (9):

$$S = \frac{b - a}{max(a, b)} \tag{9}$$

Accuracy refers to the proportion of correctly classified samples in all samples, which reflects the quality of the classification results from an overall level. The higher the accuracy, the better the classification effect. The calculation formula is shown in Formula (10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

TP represents the number of positive samples predicted as positive samples.
FN represents the number of positive samples predicted as negative samples.
FP represents the number of negative samples predicted as positive samples.
TN represents the number of negative samples predicted as negative samples.

4. Experimental Result

The generation algorithm uses the 750,000 marked illegal domain names and compares them with the existing enumeration methods to obtain the result improvement and optimization degree results.

4.1. Illegal Domain Name Clustering Result

Using the Mini-Batch K-means clustering method, the sum of squared distances corresponding to different cluster numbers is calculated. The corresponding sum of squared

distances varies with the number of clusters, as shown in Figure 5. The abscissa represents the number of clusters, and the ordinate represents the sum of squared distances.

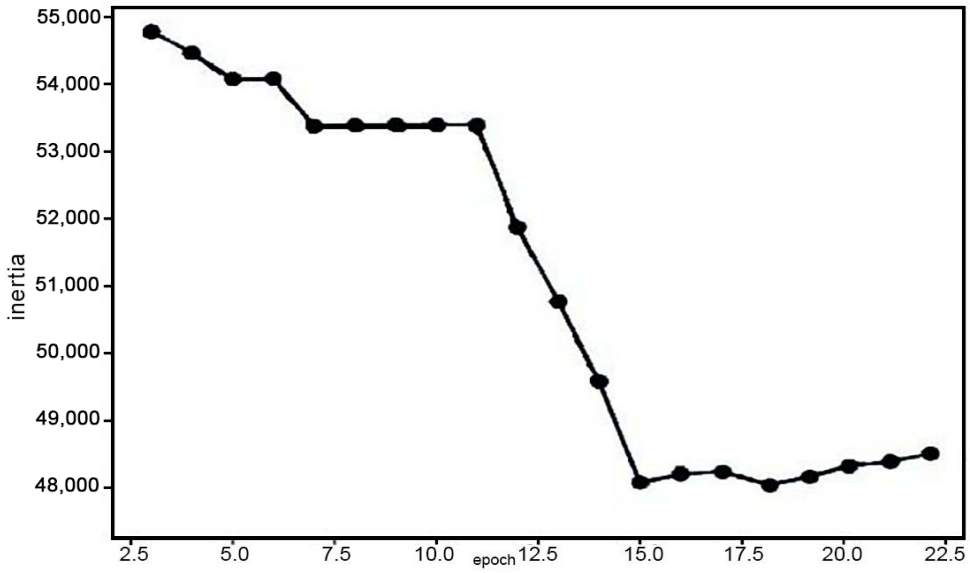

Figure 5. The sum of squared distances corresponding to different cluster numbers.

Through the above experiment process, we obtained a series of results of the classification effect of the number of clusters. We found that the clustering effect is the best when the number of clusters is 15.

The classification result introduces the sample set's silhouette coefficient [22] to measure the quality of the unsupervised clustering result. Its value range is $[-1, 1]$. When the value is negative, more clusters are assigned to errors, and the clustering effect is not ideal. On the contrary, when its value is positive, the larger the value, the closer the distance between samples of the same category. The farther the distance between samples of different categories, the better the clustering effect.

Among them, the average distance between sample a and other samples b in the same cluster is the minimum average distance between samples and samples in different clusters.

When the optimal value is 15, the silhouette coefficient of the sample set is 0.4, indicating that the clustering effect is ideal and the illegal domain names in the same cluster are relatively similar.

4.2. Illegal Domain Name Generating Result

One hundred thousand illegal domain names are randomly selected from the largest illegal domain name cluster obtained by illegal domain name clustering as experimental data. The 50,000 illegal domains were divided into 40,000 training data and 10,000 test data. The training data are used to train the generator and discriminator to work better. The test data only show the change in discriminator discrimination ability during training and does not participate in the network training.

After several experimental attempts, the input batch size is set to 256, the learning rate is set to 0.01, the generator is pre-trained 30 times, and the discriminator is pre-trained 20 times. When the generator and the discriminator can generate samples and distinguish between true and false samples, let the discriminator and generator confront training 50 times. In each confrontation process, a discriminator is trained after every six steps of generator training, and the ability of the discriminator to distinguish actual samples from generated samples is tested using test data Figure 6.

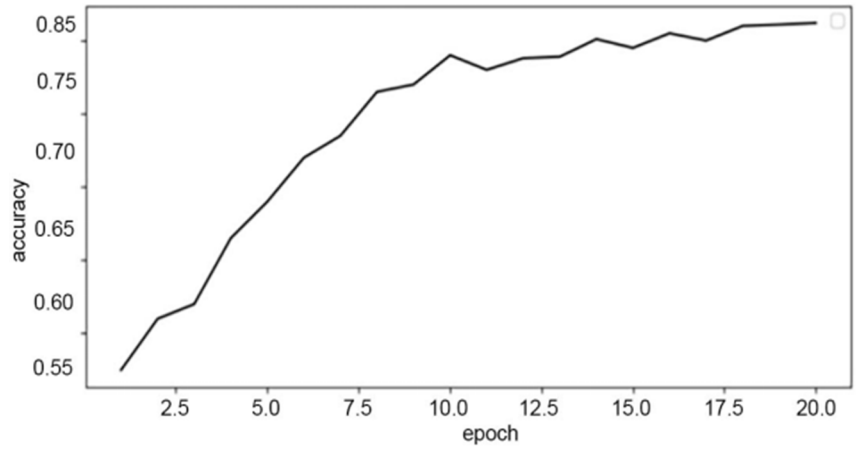

Figure 6. Classification accuracy of the discriminator.

As shown in Figure 6, after 20 rounds of pre-training, the accuracy rate of the discriminator for distinguishing true and false samples increased from 55% to 83%. The accuracy rate increased rapidly in the first 12 rounds and showed a slow growth trend after 12 rounds. It shows that after pre-training, the discriminator already has a particular ability to distinguish true and false samples. However, after 25 rounds of pre-training, the discrimination ability of the discriminator cannot be significantly improved, and it is necessary to enter the confrontation training stage to further improve the accuracy of the discriminator. Among them, accuracy refers to the proportion of correctly classified samples in all samples, which reflects the quality of the classification results from an overall level. The higher the accuracy, the better the classification effect.

The accuracy rate of the discriminator during the confrontation training process changes with the number of confrontations as shown in Figure 7.

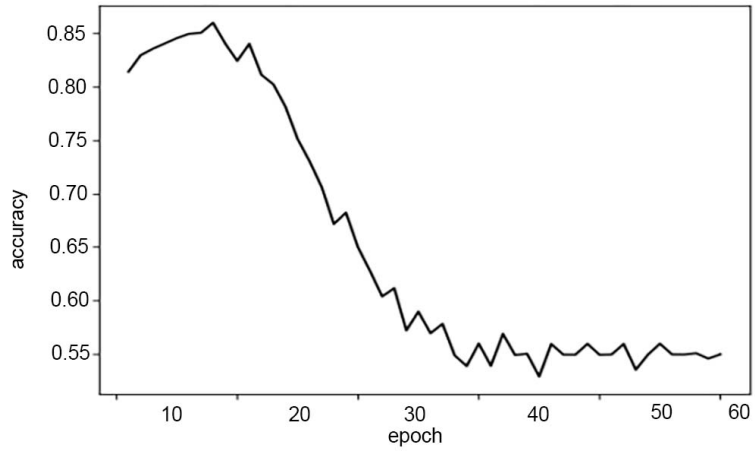

Figure 7. The accuracy of the discriminator during the confrontation training process changes with the number of confrontations.

As shown in Figure 7, the accuracy rate of the discriminator increased rapidly during the first 16 rounds of training until the 15th round rose to the highest accuracy of about 86%. After that, the performance of the discriminator dropped rapidly until the 30th round dropped to a minimum accuracy of about 50%. During the next 20 confrontational training rounds, the discriminator accuracy fluctuated around 50%. Since the discriminator has an advantage early in the confrontation, the ability to distinguish real and fake samples is still improving. Then in the mid-term of the confrontation, the generator began to have an advantage, and the ability of the discriminator to distinguish true and false samples began to decrease gradually. Hence, the accuracy rate began to decline gradually. Finally, in the

late stage of the confrontation, the generator can generate nearly actual samples. However, the discrimination cannot distinguish whether the samples are actual or generated and can only guess randomly. Therefore, the accuracy rate fluctuates around 0.55, and the sequence confrontation generation network has stabilized.

4.3. Generate Algorithm Results Comparison

The experiments of the illegal domain name generation algorithm and illegal domain name enumeration algorithm were carried out three times. The results are as follows:

In Experiment 1, SeqGAN trained from 40,000 illegal domain names continuously generated 100 batches of domain name sets, each of which contained 256 domain names, and the domain names in the same batch were not repeated. Training took almost 12 h and generation took 33 s.

As shown in Figure 8, the proportion of illegal domain names in each batch of domain name collections is similar. Each batch of generated domain names contains 61 new illegal domain names, with an average concentration of 23.82%, and a maximum of 82 new illegal domain names can be generated in one batch, with a maximum concentration of 32.03%. The proportion of illegal domain names in a batch of domain names generated by the illegal domain name generation algorithm is relatively stable.

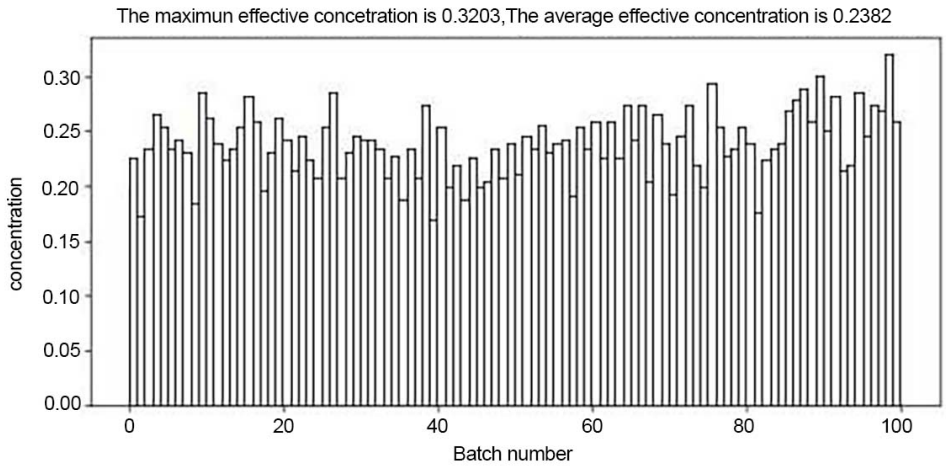

Figure 8. Histogram of batch number and concentration.

On average, each generated batch-sized domain name set contains 62.5% of invalid domain names (2.73% of non-standard domain names and 59.77% of unregistered domain names) and 37.5% of valid domain names. Therefore, this article counts the proportion of illegal domain names in the generated influential domain names, the effective concentration, as shown in Figure 9.

As shown in Figure 9, the average effective concentration is 63.54%, and the highest effective concentration reaches 84.42%. It can be seen that the effective concentration of the illegal domain name generation algorithm is relatively high. Since the domain name validity verification speed is much faster than the domain name illegality verification speed, this algorithm can greatly shorten the time to obtain illegal domain names.

In Experiment 2, SeqGAN trained from 40,000 illegal domain names continuously generated multiple batches of illegal domain names until 1,980,000 unique domain names were obtained, with 265 domain names in each batch, and the domain names in the same batch were not repeated. It took 8 h to train and 2 h to generate, for a total of 10 h.

With the increase in the number of generated unique domain names, the number of new illegal domain names contained in them is also increasing, as shown in Figure 10.

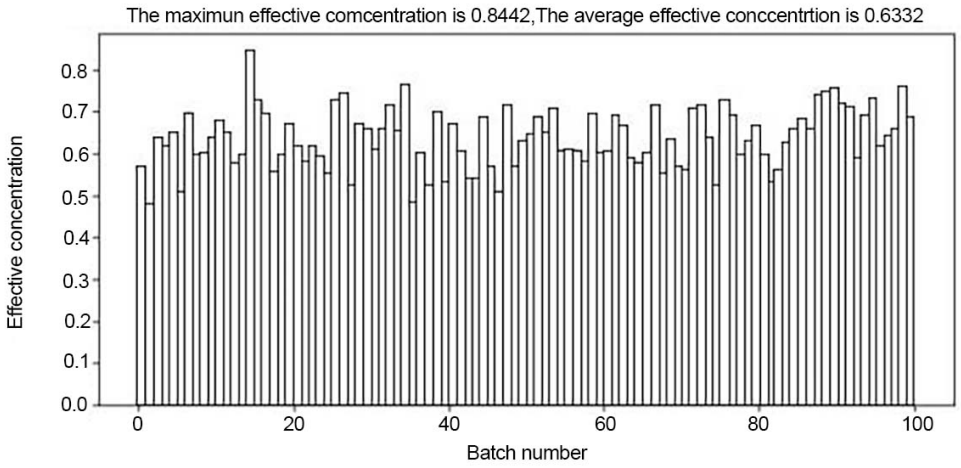

Figure 9. Histogram of batch number and effective concentration.

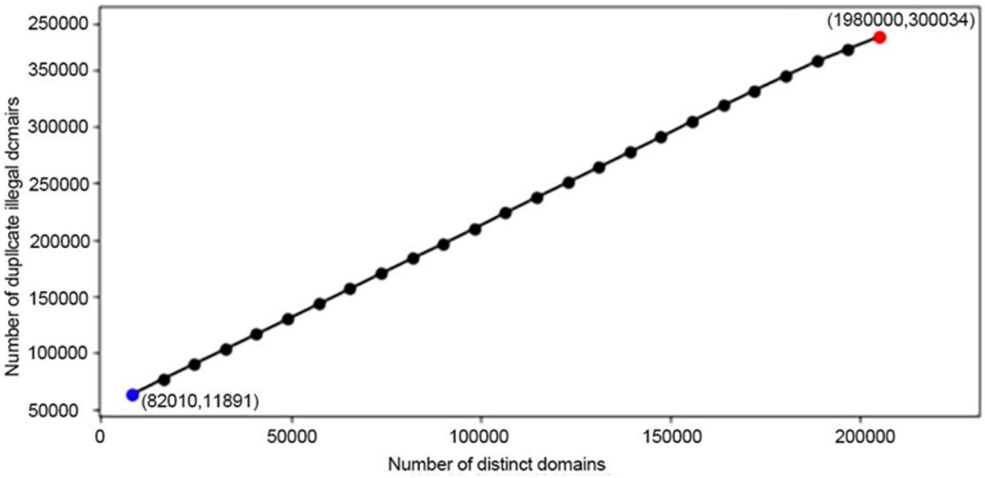

Figure 10. Changes in the number of new illegal domain names included when different numbers of domain names are generated.

As shown in Figure 10, when the number of generated domain names reaches 1,980,000, the new illegal domain names contained in it are about 300,034. The concentration is 15.15%, and the expansion degree is 7.5. It can be seen that when the number of generated domain names continues to increase, the illegal domain name generation algorithm can obtain more than 350,000 illegal domain names, which proves the effectiveness of the illegal domain name generation algorithm.

As shown in Figure 11, observing the number of batches required to generate 27,000 new domain names for the first time, the batches required to generate the same number of new domain names are getting larger and larger. It shows that the illegal domain name generation algorithm cannot generate new domain names infinitely. There is an upper bound on the number of unique domain names generated. The number of illegal domain names the illegal domain name generation algorithm can expand through 40,000 illegal domain name information is limited.

Based on the above two experimental results, it is found that the illegal domain name generation algorithm can expand more than 300,000 illegal domain names through 40,000 illegal domain name information, and the expansion degree is more significant than 7.5. When the algorithm generates 1,980,000 domain names, the number of illegal domain names obtained is about 300,034, with a concentration of 15.15%. In addition, for each batch of domain names generated, the number of illegal domain names is relatively stable

and covers various illegal types such as gambling, pornography, and fraud. The average concentration of the algorithm is 23.82%, and the average effective concentration is 63.54%.

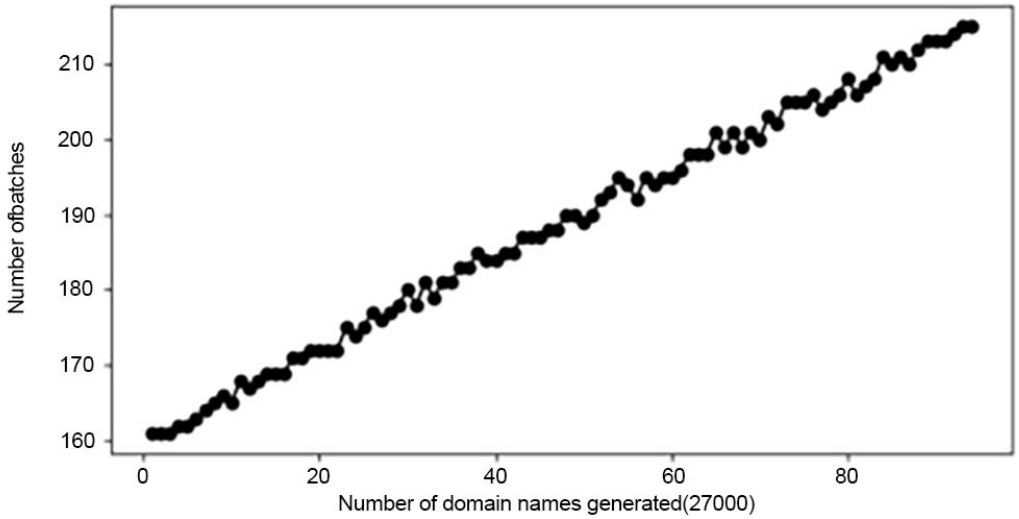

Figure 11. Changes in the number of batches required to generate the same number of new domain names.

In Experiment 3, using the illegal domain name algorithm based on the similarity in names to enumerate all domain names that may be similar to the 40,000 illegal domain names in the experiment, the enumeration took nearly 96 h.

The illegal domain name enumeration algorithm enumerates 3,003,040 domain names, including 1,501,529 valid domain names and 1,501,511 invalid domain names. The valid domain names include 300,023 illegal names, with a concentration of about 10%, an adequate concentration of about 20%, and an expansion degree of 7.5.

According to the results of Experiment 1 and Experiment 3, the performance comparison table of the illegal domain name generation algorithm and IDSN algorithm in various aspects is obtained. As shown in Table 2, the concentration and efficiency in the table are calculated when the two algorithms generate 300,000 domain names. The illegal domain name generation algorithm is superior to the IDSN algorithm in terms of efficiency, concentration, expansion, and coverage of illegal domain names.

Table 2. Performance comparison table of the algorithm and illegal domain name enumeration algorithm in various aspects.

Performance	Illegal Domain Name Generation Algorithm	Illegal Domain Name Enumeration Algorithm	Comparison Results
Efficiency (hour)	10	96	−86
Concentration (%)	15.15	10	+4.16
Extensibility (individual)	>7.5	7.5	Larger
Coverage (ten thousand)	>30	30	More extensive

To sum up, the illegal domain name generation algorithm based on the similarity of names proposed in this paper has a certain validity, stability, and superiority.

In Experiment 1, through SeqGAN concatenation generation 100 domain names, each next 256 domain names, guaranteed each next non-repetition.

In Experiment 2, SeqGAN concatenation generated 1.98 million unique region names.

Based on the above results, or the parameters of the algorithm such as extension degree and concentration.

In Experiment 3, pass through base names similar to non-jurisdictional names single-level arithmetic, single-level homologous quantitative domain names, attainment degree sum expansion degree, etc.

Due to this, the actual result can only be compared to the actual test 1, the combined result is the 3rd comparison.

5. Conclusions

This paper designs and implements an illegal domain name generation algorithm based on domain name structure. According to the similarity between illegal domain name structures, new illegal domain names are actively found from the unknown domain space. Therefore, it can effectively solve the problem that the existing illegal domain name generation methods are only effective for specific types of illegal domain names and cannot be extended to other types. Experimental results show that every batch of 265 domain names generated by the algorithm contains 61 new illegal domain names, the average concentration is 23.82%, and the generation effect is relatively stable. After clustering, the feature of 40,000 illegal domain names in the largest cluster is selected, and the algorithm obtains 300,034 new illegal domain names by generating 1,980,000 domain names. Currently, the concentration is 15.15%, and the expansion is 7.5, indicating that new illegal domain names can be effectively generated. Compared with the illegal domain name enumeration algorithm based on similar names, the illegal domain name generation algorithm performs better in terms of efficiency, concentration, expansion, and coverage. It can be concluded that the illegal domain name generation algorithm has more stability, effectiveness, and superiority. This research mainly aims to generate illegal websites based on the characteristics of domain names. In the future, we will add other attributes and characteristics of illegal websites to enhance the breadth of features. Integrate various website characteristics and attributes to improve the efficiency of illegal website generation.

Author Contributions: Conceptualization, Z.Z.; methodology, Y.C.; software, Y.L.; validation, Y.L. and T.C.; formal analysis, Y.L.; investigation, Y.C., T.C. and C.L.; resources, Z.Z.; data curation, Y.C. and C.L.; writing—original draft, Y.L.; supervision, Z.Z.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported 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].

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to most of the data studied in this paper are illegal websites and illegal data, which are only used for research and study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ITU. Measuring Digital Development: Facts and Figures 2021. Available online: https://www.itu.int/itu-d/reports/statistics/facts-figures-2021/ (accessed on 1 December 2021).
2. Invernizzi, L.; Comparetti, P.M.; Benvenuti, S.; Kruegel, C.; Cova, M.; Vigna, G. Evilseed: A guided approach to finding malicious web pages. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 428–442.
3. Wang, Q.; He, L.; Han, W. Realization of Harmful Website Discovery and Discrimination System Based on Crawlers. *Inf. Netw. Secur.* **2012**, *12*, 140–142.
4. Sato, K.; Ishibashi, K.; Toyono, T.; Miyake, N. Extending black domain name list by using co-occurrence relation between dns queries. *IEICE Trans. Commun.* **2012**, *95*, 794–802. [CrossRef]
5. Guerid, H.; Mittig, K.; Serhrouchni, A. Privacy-Preserving Domain-Flux Botnet Detection in a Largescale Network. In Proceedings of the 2013 Fifth International Conference on Communication Systems and Networks, Bangalore, India, 7–10 January 2013; pp. 1–9.
6. Khalil, I.; Yu, T.; Guan, B. Discovering Malicious Domains through Passive DNS Data Graph Analysis. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an, China, 30 May–3 June 2016; pp. 663–674.
7. Yuan, C.; Qian, L.; Zhang, H.; Zhang, T. Generation of malicious domain name training data based on generation of anti-network. *Comput. Appl. Res.* **2019**, *36*, 1540–1543, 1568.

8. Cheng, Y.; Jiang, H.; Zhang, Z.; Du, Y.; Cai, T. Birds of a Feather Flock Together: Generating Pornographic and Gambling Domain Names Based on Character Composition Similarity. *Wirel. Commun. Mob. Comput.* **2022**, *2022*. [CrossRef]
9. Cheng, Y.; Chai, T.; Zhang, Z.; Lu, K.; Du, Y. Detecting Malicious Domain Names with Abnormal WHOIS Records Using Feature-Based Rules. *Comput. J.* **2022**, *65*, 2262–2275. [CrossRef]
10. Marchal, S.; François, J.; Engel, T. Proactive Discovery of Phishing Related Domain Names. In Proceedings of the 15th International Symposium on Research in Attacks, Intrusions, and Defenses, Amsterdam, The Netherlands, 12–14 September 2012; pp. 190–209.
11. Bi, X. Research on APT Discovery Technology Based on Malicious Domain Name Detection. 2022. Available online: https://kns.cnki.net/kcms2/article/abstract?v=3uoqIhG8C475KOm_zrgu4lQARvep2SAke-wuWrktdE-tSIT2YIbQ2H_SqDW--2FOj4oZMiXvJfZXHUWHC6djVGBJTZTUJdNh&uniplatform=NZKPT (accessed on 1 December 2021).
12. Zhang, W.; Gong, J.; Liu, Q.; Liu, S.-D.; Hu, X.-Y. Lightweight domain name detection algorithm based on morpheme features. *J. Softw.* **2016**, *27*, 2348–2364.
13. Peng, K.; Leung, V.C.M.; Huang, Q. Clustering approach based on mini batch Kmeans for intrusion detection system over big data. *IEEE Access* **2018**, *6*, 11897–11906. [CrossRef]
14. Béjar Alonso, J. K-Means vs Mini Batch K-Means: A Comparison. 2013. Available online: https://upcommons.upc.edu/handle/2117/23414 (accessed on 1 December 2021).
15. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2852–2858.
16. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-Aware Neural Language Models. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2741–2749.
17. Bojanowski, P.; Joulin, A.; Mikolov, T. Alternative Structures for Character-Level RNNs. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015; pp. 1–9.
18. Wang, K.-F.; Gou, C.; Duan, Y.-J.; Lin, Y.-L.; Zheng, X.-H.; Wang, F.-Y. Generative Adversarial Networks: The State of the Art and Beyond. *Acta Autom. Sin.* **2017**, *43*, 321–332.
19. Liang, J.; Wei, J.; Jiang, Z. Generative Adversarial Networks GAN Overview. *J. Front. Comput. Sci. Technol.* **2020**, *14*, 1–17.
20. Yu, Y.; Chen, D. Botnet Detection Based on Convolutional Neural Network. *Comput. Appl. Softw.* **2022**, *39*, 336–341, 349.
21. Zhang, Z.; Cheng, Y.; Wu, X. Illegal Domain Name Mining Method Based on Domain Name Structure Similarity: China. CN108712403A, 26 October 2018.
22. Zhu, L.; Ma, B.; Zhao, X. Clustering validity analysis based on silhouette coefficient. *J. Comput. Appl.* **2010**, *30*, 139–141.