Detection of Malicious Domain Names Based on Hidden Markov Model

1st Pin Lv
School of Cyber Security
University of Chinese Academy of Sciences
Beijing, China
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
lvpin@iie.ac.cn

2nd Lingling Bai
Faculty of Information Technology
Beijing University of Technology
Beijing, China
bjutljl@163.com

3rd Tingwen Liu
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
liutingwen@iie.ac.cn

4th Zhenhu Ning
Faculty of information technology
Beijing University of Technology
Beijing, China
nzh41034@163.com

5th Jinqiao Shi
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
shijinqiao@iie.ac.cn

6th Binxing Fang
Guangdong Institute of Electronic Information Engineering
University of Electronic Science and Technology
Dongguan,Guangdong,China
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China
fangbx@iie.ac.cn

Abstract*—The Domain Name System (DNS) is an important core infrastructure of the Internet, domain names and IP addresses is a distributed database that maps to each other, however, due to the defects of its own protocol, there have been many malicious attacks against domain names, such as spoofing attacks, botnets, and domain name registrations, as a result, the security of domain names has become one of the problems that must be solved for the safe and reliable operation of the Internet. Based on the hidden Markov model (HMM), this paper analyzes the difference between the malicious domain name and the normal domain name in the various characteristics of DNS communication, and uses Spark's fast extraction to distinguish their attributes, the Baum-Welch algorithm and Viterbi algorithm in the Markov model can quickly classify unknown domain names accurately to achieve effective detection of malicious domain names. Finally, the HMM was compared with the commonly used random forest model through experiments, and the accuracy and recall rate were compared. The results show that the application of HMM improves the performance of the classifier to obtain more accurate detection results.

Keywords—*malicious domain name, hidden Markov model, Spark*

I. Introduction

The domain name service system was originally established on the basis of mutual trust. It is a completely open service system[1]. Its complete trust in the domain name makes it an important part of malicious network behavior. The malicious domain name has brought serious harm to the society. It has spawned botnets and conducted DDOS (Distributed Denial of Service) attacks[2]. It has used broilers to conduct click fraud, harmful search, advertising and other industries. This has led to higher operating costs for e-commerce sites, impaired corporate reputations, and greater damage to e-commerce and bank users; Many botnets have also introduced malicious domain name algorithm generation techniques[3]. The malicious domain name generation algorithm generates a large number of domain names in order to strengthen its own organization and control capabilities and enhance its own survivability, and prolong the survival time of botnets. It can be seen that the detection of malicious domain names is facing severe challenges. The traditional method of malicious domain name detection is to detect malicious domain names through methods such as domain name blacklist, reverse technology, and data mining. However, as more and more new network technologies are applied, the generation and use of malicious domain names have become more and more flexible, traditional detection methods cannot effectively detect these malicious domain names. In addition with the continuous increase in the number of registrations, queries, and system deployments of the global domain name system, the complexity of the DDOS attack scale and attack technology for the domain name system is also significantly improved. In addition, the research on the future technological development of the domain name system and the technology of DNS privacy

978-1-5386-4210-8/18/$31.00 ©2018 IEEE
DOI 10.1109/DSC.2018.00105

protection will also become the focus and difficulty for the industry in the future for a long period of time.

After research and analysis, the realization of malicious domain name attacks will be achieved through the domain name resolution system. So when hackers use malicious domain names to perform sabotage activities, the corresponding data and its characteristic behavior patterns will be recorded in the DNS server log [4]. The records of these DNS logs contain both the parsed records of goodwill domain names and their corresponding data characteristics as well as the behavior data of malicious domain names. It can be seen from the above that we can analyze malicious records in the DNS server and their corresponding characteristics to obtain the specific behavior of malicious domain names, so as to achieve the detection of malicious domain names. This article addresses this issue by analyzing the parsing records and their corresponding characteristics of domain names in massive DNS logs. Using hidden Markov model and big data technology Spark[5], it combines domain names' own characteristics, timeliness, corresponding IP characteristics and related domain name sets to study the classification and prediction of malicious domain names.

II. Related work

The hidden Markov model was proposed as a statistical analysis model by a series of papers published in the 1960s and 1970s by the American mathematician Leonard E. Baum et al[6]. It has a wide range of applications in many fields such as speech recognition, biological information, and machine learning. The hidden Markov model is the extension and extension of the Markov model. That is, the state of the model at any moment is invisible. The observer cannot infer the transition probability and other relevant parameters by observing a state sequence. However, the hidden state can be predicted by observing the observed value of HMM at each time, and the observed value at this time is only related to the underlying hidden state.

Typically, a HMM represented by a five- tuple [7], where :
(1) S denotes a set of hidden states in the model, $S = \{S_1, S_2, \ldots, S_N\}$,N *d*enotes the number of states, and the state sequence at the Tth time point is $Q = \{\mathrm{Q}_1, \mathrm{Q}_2, \ldots, \mathrm{Q}_N\}$.
(2) V denotes the set of observations that each state may output, $V = \{V_1, V_2, \ldots, V_M\}$,and M denotes the number of different observations. The sequence of observations at the Tth time point is $Q = \{Q_1, Q_2, \ldots, Q_T\}$.
(3) π represents the probability distribution of the initial state, $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$; $\pi_i = P(Q_1 = S_i)(1 \le \mathrm{i} \le N)$ represents the probability of taking a state in the initial state.
(4) A represents the transfer matrix between states, $A = [a_{ij}]_{N \times N}$; $a_{ij} = P(Q_{t+1} = S_j \mid Q_t = S_i)(1 \le i, j \le \mathrm{N})$ represents the probability of transition from state at time t to state at time t+1;
(5) B represents the probability distribution matrix of observations, $\mathrm{B}=[\mathrm{b}_{jk}]_{M \times N}$; $\mathrm{b}_{jk} = P(O_t = V_k \mid Q_t = S_j)(1 \le j \le N, 1 \le k \le M)$ indicates the probability of observing the value S_j when the state V_k appears.

There are three basic issues surrounding the hidden Markov model :

1. Given a model, how to calculate the probability of a particular output sequence;

2. Given a model and a specific output sequence, how to find the state sequence most likely to produce this output;

3. Given a sufficient amount of observation data, how to estimate the parameters of the hidden Markov model.

III. Application of HMM in detection of malicious d omain name issues

A malicious domain name is an abstract and concrete concept. The domain name attributes extracted from the massive DNS logs may have different states at different times. They may all be well-behaved domain name signatures, may also have malicious domain name signatures, and may have different levels of malicious damage. At different times, the transition between the malicious strength status of these domain names is in line with the Markov character. These states are not observable, but the relevant domain name attributes extracted from the massive DNS logs, such as the domain name's own characteristics, temporality, corresponding IP characteristics, and related domain name sets that are observable and statistical[8,9]. And there is indeed a specific link between these observed variables and the malicious strength of the domain name. These observation variables are determined by the malicious strength of the domain name, and the malicious strength of this domain name is determined by the state of the domain. The construction of such a model requires the application of a dual mapping process [10].

A. Selection of Observation Variables

Observed variables selected for identifying domain names should be able to distinguish between good and bad domain names. That is, in the domain name detection process, the values of these observation variables will be very different in the absence of malicious domain names and malicious domain names. In summary, by analyzing the relevant characteristics of domain names in DNS log information, we have extracted a total of 12 characteristics of five categories. Table 1 shows the description of the 12 features:

Table 1 Extracted Feature Value Table

Attribute set	***Attribute name***	
Domain character characteristics	1	Number accounting
	2	Domain character entropy
	3	Domain length
Domain name corresponds to IP characteristics	4	Domain name corresponds to the number of IP
	5	Domain name corresponds to IP difference
	6	The number of domain names corresponding to the country
Domain cache cycle characteristics	7	Average TTL
	8	TTL changes
	9	TTL standard deviation
NS recording features	10	Number of domain name server

		changes
	11	The average length of change of the domain name server
Domain name lifetime characteristics	12	Domain life cycle

B. Malicious domain name recognition system based on HMM model

The domain name detection time series contains 12 characteristic values of the domain name at each time point. This observable multidimensional time series is mainly used for parameter training and estimation of malicious domain name detection HMM models. For a specific time point, we can obtain some attribute observations for a specific domain name at that time point. Take this multidimensional attribute observation as input, and use the HMM model established by the system to identify and determine the current click observation value. Calculate the probability of its appearance, and then determine whether there is a malicious domain name at this time.

The concrete steps of the HMM-based malicious domain name detection model at each stage are as follows:

Step1 Initialize the HMM parameters in the new model:

(1) State S: The hidden Markov model corresponding to a specific domain name access behavior does not know the hidden state number in advance. We use heuristics to determine the hidden state number of the HMM model by observing the effect of the model's identification when the number of implicit states N=2, 3, 4, and 5 . Once the number of hidden states has been determined, it should be within a certain period of time. It stays the same. The heuristic state set is re-used at regular intervals to eliminate the influence of new domain name feature behavior patterns that may occur in domain name access behavior.

(2) Observed value V: The resolution record of the domain name in the massive DNS log of the DNS server and its corresponding characteristics. For example, the time series formed by observable variables such as the lifetime of the domain name in the unit time, the TTL mean value, the TTL standard deviation, and the number of IP addresses. M represents the number of domain name features in the collection.

(3) The probability distribution of the initial state $\prod = \{\pi_i\}, \pi_i = P(Q_1 = S_i)$ represents the probability of initial occurrence of all domain names S_i in the system. M 1represents the number of initial occurrences of domain name S_i in different states. N1 represents the sum of the number of occurrences of domain name S_j in different initial states, $\pi_i = M1/N1$.

(4) State transition probability matrix $A = [\alpha_{ij}]_{N\times N}, \alpha_{ij} = P(Q_{t+1} = S_j \mid Q_t = S_i)(1 \le i, j \le N)$ indicates the probability that all the different state domain names in the system are at S_i certain time t and transition to state S_j at time t+1, where t only indicates the sequence of domain name states. M2 represents the number of transfers from S_i a to S_j . N2 represents the sum of the number of transitions from S_i to S_k . $\alpha_{ij} = M2/N2$.

(5) Probability distribution matrix of observations $B = [b_{jk}]_{M\times N}, b_{jk} = P(O_t = V_k \mid Q_t = S_j)$ represents when the domain name S_j appears in different states, the corresponding characteristic of the domain name resolution record is the probability of V_k . M3 indicates the number of times that the parsing record feature is V_k when the domain name state is S_j . N3 represents the total number of times the parsing record feature is V_k . $b_{jk} = M3/N3$, $1 \le j \le \mathrm{N}, 1 \le \mathrm{k} \le M$.

Stp2 Uses the Baum-Welch algorithm, according to the corresponding characteristics (observed sequence) $\mathrm{O}=\{\mathrm{O}_1\mathrm{O}_2\cdots\mathrm{O}_T\}$ of the domain name resolution records in the DNS server at time T in the training set, and the implicit state sequence is $\mathrm{Q}=\{\mathrm{Q}_1\mathrm{Q}_2\cdots\mathrm{Q}_\mathrm{T}\}$. The training of the parameters in the HMM model makes $\mathrm{P}(\mathrm{O}|\lambda)$ the largest.

(1) Determine the log-likelihood function of complete data consisting of observation data O and state data S [11].

$$\log P(O,S \mid \lambda)$$

(2) Step E: Find Q Function

$$Q(\lambda,\lambda^*) = \sum_S P(S \mid O,\lambda)\log P(O,S \mid \lambda^*)$$

(3) Step M: Solving new model parameters by maximizing Q functions

$$\overline{\lambda} = \arg\max_{\lambda^*} Q(\lambda,\lambda^*)$$

$$\mathrm{Q}(\lambda,\lambda^*) = \sum_S P(S \mid O,\lambda)\log P(O,S \mid \lambda^*)$$

$$= \sum_S \frac{P(O,S \mid \lambda)}{P(O \mid \lambda)}\log P(O,S \mid \lambda^*)$$

$$= \frac{1}{P(O \mid \lambda)}\sum_S P(O,S \mid \lambda)\log P(O,S \mid \lambda^*)$$

For a given parameter, $P(O \mid \lambda)$ is a constant, so:

$$\max_{\lambda^*} Q(\lambda,\lambda^*) = \max_{\lambda^*}\sum_S P(O,S \mid \lambda)\log P(O,S,\mid \lambda^*)$$

$$\lambda^*=\{\hat{\pi},\hat{A},\hat{B}\},\hat{\pi}=\{\hat{\pi}_\mathrm{i}\},\hat{A}=\{\hat{\alpha}_{ij}\},\hat{B}=\{\hat{b}_j(k)\}$$

$$P\left(O,S \mid \lambda^*\right) = \hat{\pi}_{s_1}\hat{b}_{s_1}\left(o_1\right)\hat{\alpha}_{s_1s_2}\hat{b}_{s_2}\left(o_2\right)\cdots\hat{\alpha}_{s_{T-1}s_T}\hat{b}_{s_T}\left(o_T\right)$$

$$Q(\lambda,\lambda^*) = \sum_s P(O,S \mid \lambda)\log P(O,S \mid \lambda^*)$$

$$= \sum_s P(O,S \mid \lambda)\log(\hat{\pi}_{s_1}\hat{b}_{s_1}(o_1)\hat{\alpha}_{s_1s_2}\hat{b}_{s_2}(o_T)(o_2)\cdots\hat{\alpha}_{s_{T-1}s_T}\hat{b}_{s_T}(o_T))$$

$$= \sum_s P(O,S \mid \lambda)\log\hat{\pi}_{s_1} + \sum_S P(O,S \mid \lambda)\left(\sum_{t=1}^{T-1}\log\hat{\alpha}_{s_{T-1}s_T}\right) +$$

$$\sum_S P(O,S \mid \lambda)\left(\sum_{t=1}^{T}\log\hat{b}_{s_t}(o_t)\right)$$

$$\max \sum_{s} P(O,S \mid \lambda) \log \hat{\pi}_{s_1} \qquad with \sum_{i=1}^{N} \hat{\pi}_i = 1$$

$$\max \sum_{s} P(O,S \mid \lambda)\left(\sum_{t=1}^{T-1} \log \hat{\alpha}_{s_t s_{t+1}} \right) \qquad with \sum_{j=1}^{N} \hat{a}_{ij} = 1$$

$$\max \sum_{s} P(O,S \mid \lambda)\left(\sum_{t=1}^{T} \log \hat{b}_{s_t}(o_t) \right) \qquad with \sum_{k=1}^{N} \hat{b}_j(k) = 1$$

$$\pi_i^{n+1} = \frac{P(\mathrm{O},\mathrm{s}_1 = i \mid \lambda^n)}{P(O \mid \lambda^n)} \qquad i = 1,2,\cdots,N$$

$$\alpha_{ij}^{n+1} = \frac{\sum_{t=1}^{T-1} P(O, s_t = i, s_{t+1} = j \mid \lambda^n)}{\sum_{t=1}^{T-1} P(O, s_t = i \mid \lambda^n)}$$

$$b_j(k)^{n+1} = \frac{\sum_{t=1}^{T} P(O, s_t = j \mid \lambda^n)\delta_{t,k}}{\sum_{t=1}^{T} P(O, s_t = j \mid \lambda^n)}$$

IV. Experimental Testing and Analysis

A. Test Preparation

Test preparation includes building a Spark system, preparing DNS logs, and collecting good and evil domain names. Because Spark does not have its own external storage system [12-14], to use Hadoop's HDFS file system, it is necessary to first build the Hadoop system and then build the Spark system. In addition, in order to distinguish the effect of the HMM model built in this paper on malicious domain name recognition, we select two evaluation indicators: accuracy and recall rate . In short, the accuracy rate refers to the overall accuracy of the system, and the recall rate refers to the correct rate of malicious domain names.

This article collected a few millions of malicious domain names from different sources such as malware.com and Malware Domain List. Well-meaning domain name comes from the top Alex domain name, because the front site includes all walks of life, and most of them are trustworthy.

- Test Environment

First, set up a Spark cluster and a Hadoop cluster. The entire Spark cluster consists of four servers. Under the Spark cluster, one is the master node and is responsible for the resource management of the cluster. The other three are used as slave nodes to perform the calculation tasks of the detection model.

B. Test and Analysis of Classification Effect under hidden Markov model

- Classifier effect test under the original parameters

This system is based on the idea of using EM algorithm, which achieves a local maximum by iterative calculation and finally obtains model estimation parameters. In theory, the model parameters obtained by training the HMM through the Baum-Welch algorithm are local optimal solutions, but not necessarily global optimal solutions. Therefore, given the training data, different local optimal solutions may be obtained using the Baum-Welch algorithm under different initial models. If a good initial model can be selected, it may make the local optimum into a global optimum or an approaching global optimal model. After research and analysis the selection of the initial values of the initial probability distribution and the state transition probability matrix has little effect on the parameter training and the implicit state estimation of the model[15], so it can be selected uniformly or randomly according to the model, and the observation probability and the selection of the matrix initial value is more important. It can be divided by the K-means clustering method [16], and then the corresponding probability distribution is counted as the initial value. Test the model with the Spark cluster. Ten tests were respectively performed to establish the model, and the accuracy and recall rate of the model were recorded. The test results are shown in Table 2 and Figure 1 below.

Table 2 Classification Results under Original Parameters

Detection times	***Domain data volume***	***Accuracy***	***Recall***
1	17200	80.32%	82.78%
2	26906	83.51%	80.72%
3	37809	86.79%	83.89%
4	48976	81.86%	79.86%
5	57680	90.32%	81.57%
6	65789	82.57%	80.78%
7	78970	89.85%	84.67%
8	89070	84.62%	87.64%
9	90789	91.52%	89.32%
10	109080	87.76%	85.87%

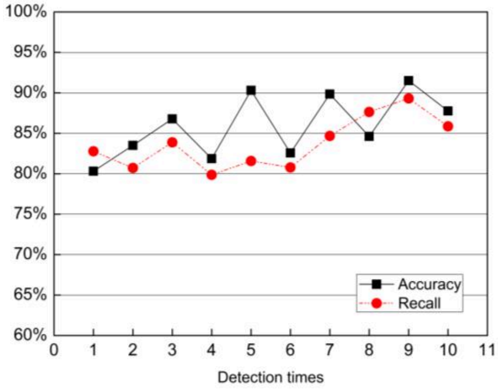

Fig. 1. Accuracy and recall of classification model under original parameters

As can be seen from Figure 1, the accuracy and recall rate of the malicious domain name classification in the test set under the original parameters of the HMM is very stable and the performance is basically met, but the accuracy and recall rate of the model classification results under the original parameters is not ideal, so it is necessary to adjust the initial value of the observation probability through the clustering method to increase the ratio of accuracy and recall.

- Classification Model Effect Test under Tuning Parameters

The initial value of the observation probability matrix is divided by the K-means clustering method.

After the initial value is selected, the classification model is trained through the training set, and the test model is used to test the effect of the classification model. Analysis of Figure 3 shows that compared with the parameter optimization, the accuracy rate and recall rate of the classification have been significantly improved. Therefore, it can be preliminarily judged that the HMM can obtain more reliable malicious domain name prediction results after the initial value of the observed probability matrix is adjusted.

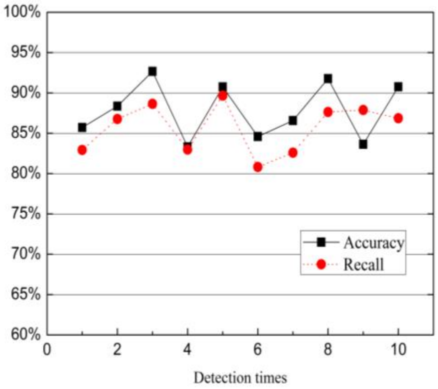

Fig. 2. Accuracy and recall of classification models under tuning parameters

C. Comparison and Analysis of Classification Effect under the Random Forest Model

After the completion of the development of the classification model and the optimization of the initial parameters of the observed values, this model showed an excellent classification effect. However, in addition to the hidden Markov model in the data mining field, there are other excellent classification algorithms, so it is necessary to test the performance of the property set extracted by the system on other classification models.

First use the samples sampled from the text data set to form a balanced training set. It is used to train the random forest model and use the test set to test the effect of the classification model. The specific result is shown in the figure 3 below.

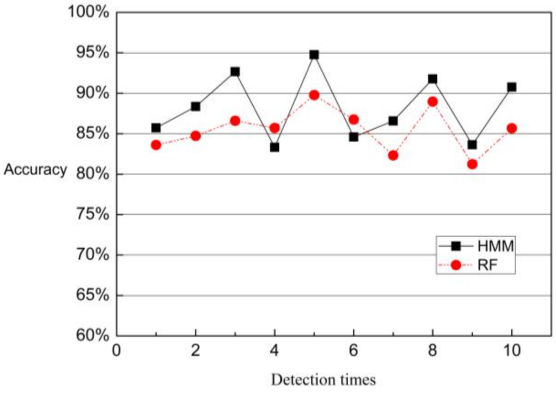

Fig. 3. Comparison of HMM and RF Classification Accuracy

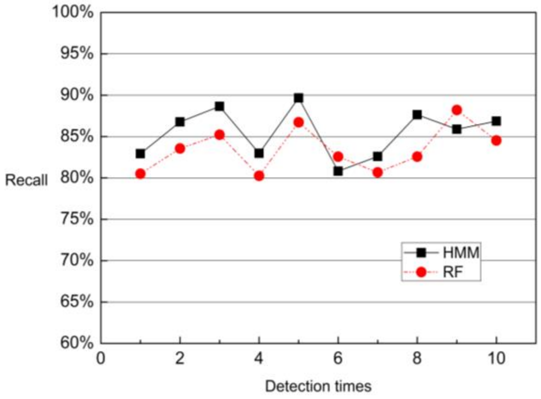

Fig. 4. Comparison of HMM and RF Classification Recall Rate

From the comparative analysis of Figure 3 and Figure 4, it can be seen that due to the unreasonable choice of parameters in the domain name resolution records and corresponding feature behaviors obtained from the massive DNS logs regardless of the behavior of the network or its own characteristics, it leads to the random forests classification model has an overfitting phenomenon in the classification problem of large noise, which leads to the weakening of generalization ability. However, the Hidden Markov Model's detection of malicious domain names first uses a given amount of observation data, estimates the parameters of the Hidden Markov Model, and then uses the estimated parameters and the specific output sequence to find the most likely state sequence of this output, this method avoids the deficiencies of the above defects, to a large extent, improve the performance of the classifier.

V. CONCLUSION

This paper proposes a scheme for domain name classification using hidden Markov model in Spark environment. The core idea of the solution is to use the tools in the big data ecosystem to improve the effectiveness and efficiency of malicious domain name detection through the performance advantages embodied in parallel processing and the precise classification advantages of data mining domain classification algorithms. The experimental comparison of classification detection by HMM, greatly improves the accuracy and recall rate of the model. However, the Baum-Welch algorithm in the HMM parameter estimation algorithm has limitations and is limited to the parameters defined by the hidden Markov model. Based on the constraints, no other constraint information can be added. After analyzing and researching, these additional constraint information will change the parameter structure of the hidden Markov model, which brings great difficulty for parameter estimation. However, in practical problems, there is often a priori constraint information that represents the background of the actual demand of the system. In addition, due to the randomness and uncertainty of the training data, errors are inevitable in the process of model establishment. If we can reasonably use the constraint information when building a model, we can improve the accuracy of the model to a certain extent, so as to better predict and analyze domain names.

REFERENCES

[1] Chung W J, Kim H S, Jung H Y, et al. Domain name service system and method thereof: US, US7756065[P]. 2010.

[2] Christenson D A, Gloe C T. Managing an alias host and domain names on a DNS server[J]. 2015.

[3] Arends R. DNS security introduction and requirements[J]. Rfc, 2005.

[4] Ichise H, Jin Y, Iida K. Detection Method of DNS-based Botnet Communication Using Obtained NS Record History[C]// Computer Software and Applications Conference. IEEE, 2015:676-677.

[5] Nabi Z. Introduction to Spark[M]// Pro Spark Streaming. Apress, 2016.

[6] Baum L E, Petrie T. Statistical Inference for Probabilistic Functions of Finite State Markov Chains[J]. Annals of Mathematical Statistics, 1966, 37(6):1554-1563.

[7] Baum L E, Eagon J A. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology[J]. Bull.amer.math.stat, 1967, 37(3):360-363.

[8] Wang Z T. Hidden Markov Model(HMM) and Its Application[J]. Journal of Hunan University of Science & Engineering, 2009.

[9] Cowen L L E, Besbeas P, Morgan B J T, et al. Hidden Markov models for extended batch data[J]. Biometrics, 2017, 73(4).

[10] Zhu M, Guo C S. Hidden Markov Model and Its latest Application and Progress[J]. Computer Systems & Applications, 2010.

[11] Zisapel R, Peles A, Fuks S. Load balancing[J]. 2001, 34(1):42.

[12] Holz T, Gorecki C, Rieck K, et al. Measuring and Detecting Fast-Flux Service Networks[C]// Network and Distributed System Security Symposium, NDSS 2008, San Diego, California, Usa, February -, February. DBLP, 2008:487 - 492.

[13] Nazario J, Holz T. As the net churns: Fast-flux botnet observations[C]// International Conference on Malicious and Unwanted Software. IEEE, 2008:24-31.

[14] Passerini E, Paleari R, Martignoni L, et al. FluXOR : Detecting and Monitoring Fast-Flux Service Networks[M]// Detection of Intrusions and Malware, and Vulnerability Assessment. Springer Berlin Heidelberg, 2008:186-206.

[15] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[16] Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System[C]// MASS Storage Systems and Technologies. IEEE, 2010:1-10.