Nodes detection in CDN with enhanced naive bayes tree

Ruijie Hou, Changyong Guo*, Zhaoxin Zhang and Ning Li
Harbin Institute of Technology, School of Computer Science and Technology, 264209 Weihai, China

*Corresponding author. Email: guocy@hit.edu.cn

Abstract. In recent years, Content Distribution Network (CDN) plays a critical and central part of Internet infrastructure. There are more and more studies on CDN. Nodes detection in CDN, as a key technology among them, has become a hot research topic. Current methods mainly focus on collecting one or some CDN vendors nodes by manually constructing features set. However, because CDN nodes detection is not restricted by specific CDN vendors, these methods are not applicable. In this paper, we proposed a novel machine learning algorithm, i.e. enhanced Naive Bayes Tree to identify CDN IP addresses. This algorithm makes full use of the advantages of Decision Tree and Naive Bayes algorithm, and further improves the performance through enhanced part. We build this classifier based on analysis of the characteristics of DNS resolutions, HTTP logs and WHOIS lookup. Experimental results show that our approach outperforms in terms of accuracy and amount of packages. Moreover, we separately tested the enhanced part and it performs well. We believe that this method could be applied to CDN IP addresses detection.

1. Introduction
With the rapidly developing of the Internet, CDN (Content Delivery Network) play a more and more important role. Because of its benefits[1], CDNs have become an indispensable part of the Internet and served most of the popular websites. According to statistics, more than 74% of the Alexa top 1000 websites use the CDN service.[2]

From Figure 1, we can see the infrastructure of CDN is mainly consists of several surrogates which get content from origin servers, and request routing system.[3,4] As a geographically distributed network, the collection and identification of CDN nodes has always been a topic that cannot be avoided in CDN research. Several methods[7-19] were proposed to find the CDN network nodes in previous studies. These studies are mainly based on DNS resolution results, HTTP logs, and WHOIS query results to extract features by manual analysis. However, artificial feature extraction need a large workload. As a result, these methods can only be applied to collect one or some CDN vendors’ nodes, and cannot be used as a universal CDN nodes identification method. They are limited.
In this paper, we propose a new method, namely enhanced NB-Tree to detect CDN IP addresses. It consists of three parts: decision tree part, naive Bayes part and enhanced part. The decision tree can accurately identify CDN nodes based on CNAME features and WHOIS features. However, it is limited by the size of the features set. Naive Bayes part further recognition unidentified samples based on probability calculations. The enhanced part learns new CNAME features on the basis of the Naive Bayes part discrimination, and improves the recognition ability of the decision tree part. The enhanced NB-Tree not only makes full use of the advantages of decision trees and naive Bayes, but also realizes a virtuous circle, that is, the decision tree is enhanced based on the results of naive Bayes judgments. The experimental results show that the method we proposed has a good performance in accuracy and other aspects.

In summary, our major contributions are as follows:

1. Firstly, we design a new method, i.e. enhanced NB-Tree to identify CDN IP addresses.
2. Secondly, in addition to entropy gain, other factors affecting the construction speed of the decision tree must be considered. These factors can be expressed by a coefficient $\alpha$.
3. Thirdly, we add an enhanced part to the model to learn new CNAME features based on the results of the Naive Bayes part and enhance the recognition ability of the decision tree part.

The remainder of the paper is organized as follows. Section 2 gives related works about the recognition of the CDN nodes. Section 3 introduces data set and describes the framework and algorithm of the methodology. Section 4 presents the experimental results. Finally, we make our conclusion in section 5.

2. Related work

CDN detection is certainly not new. It has always been a basic problem throughout related research on CDN and has attract much attention. According to the basic for identifying CDN nodes, previous works can be divided into four categories, as shown below:

1) **DNS resolution.** According to a recent empirical analysis[5], the most popular request routing technique is based on DNS. So there are many methods based on DNS resolution to detect CDN nodes. Hailing[6] proposed a machine learning algorithm. It detects CDN-hosted site with three categories of feathers, i.e. IPs, domains and TTLs, which are extracted from the DNS records. In view of the fast-changing characteristics of both FF domain names and CDN domain names, Li[7] proposed a deep learning method mainly for fast-flux domain detection. In order to chart the network of Akamai and Limelightand, Cheng[3] designed a measurement platform, which collects CDN nodes by querying a large number of LDNSs all over the world for all of the customers’ CNAMEs they found. What’s more, they conduct measurements to characterize the performance of two large-scale commercial CDNs.
2) **HTTP Log.** CDN customers commonly configure how HTTP headers in requests should be modified when being forwarded by CDNs. So there are many previous studies have used it to identify CDN nodes. For instance, Cheng[3] finds a method with HTTP information to discover the CDN nodes during their expansion process. Guo[2] gathers CDN surrogates’ IP addresses with CDNs’ information exposure through HTTP headers or sub-domain names. And Adhikari[8, 9] infers the architecture of Netflix and Hulu mainly rely on HTTP logs.

3) **WHOIS[10] lookup.** It is known that we can gain many information of a IP address through WHOIS lookup, for example, the owner of a IP address. So WHOIS lookup is also an important way to detect CDN nodes. Adhikari[11] and Haniling[7] use it to identify CDN nodes in their study.

4) **Published IP ranges.** Many CDN vendors provide publicly disclosed IP ranges or CDN-provided tools to identify CDN replica server IPs, Xue[12] uses these tools to collect CDN nodes in their research.

These methods are simple and effective in their study to collect only one or some CDN vendors’ nodes. However, because of the need to extract features manually, these methods are not applicable to determine whether an IP address is a CDN node, when we do not have a clear CDN vendor as the target. In this paper, we propose a new method, i.e. enhanced NB-Tree to identify CDN nodes, which serves as a complement to existing technologies.

3. Methodology

In this section, we propose a new algorithm, namely enhanced NB-Tree. General NB-Tree induces a hybrid of decision tree nodes contain univariate split as regular decision-tree, but the leaves contain Naive-Bayesian classifiers.[13] It utilizes the advantages of both them and performs well in many examples. Further, depending on the actual scene, the new method adds an enhanced part on the basis of NB-Tree. The enhanced part can learn new CNAME features depending on the result analysis of the Naive-Bayesian part to improve the ability of the decision-tree part. We call the new method as enhanced NB-Tree.

3.1. **Data Collection**

Previous studies[9] show that all CDNs employ locality-aware DNS resolution to achieve load balancing. Since a CDN typically returns IP addresses based on the location of the open resolvers, we resolve 161392 domain names by 2948 recursive DNS servers which belongs to three operators distributed in all cities of China to construct our dataset. Then we send HTTP requests to each node to ensure the node could be accessible and collect HTTP response header.

As our method is based on supervised learning algorithm, we need to construct the labeled dataset. On one hand, we find many CDN vendors provide their IP ranges[14, 15] or tools[16, 17, 18, 19] to judge CDN IPs. So we identify CDN IP addresses by this way. On the other hand, as we all know, if a domain name uses CDN acceleration, the results of DNS resolution differ from regions. Therefore, we treat the domain names with a single DNS resolution result as a normal domain names, and regard their IP addresses as ordinary IP addresses. Finally, we conservatively find out 15572 CDN IP addresses and 7680 ordinary IP addresses. The count of the labeled samples is listed in Table 1.

| Label    | Vendors | Numbers | Count |
|----------|---------|---------|-------|
| CDN IP   | Alibaba | 12321   | 15572 |
|          | Baidu   | 82      |       |
|          | Tencent | 1065    |       |
|          | Wangsu  | 2122    |       |
| Ordinary IP |        |         | 7680  |
| Summary  |         |         | 23252 |

Table 1. The description of the dataset.
3.2. The structure of NB-Tree
The structure of enhanced NB-Tree as shown in the figure 2. From it, we can see that enhanced NB-Tree consists of three parts. 1) Decision-Tree part. It will identify CDN IP addresses relying on the CNAME feature and the WHOIS lookup information. 2) Naive Bayes part. For the remaining samples that cannot be identified in step 1, it will estimate the probability that the sample belongs to each category, and classify the sample into a category with higher probability. 3) The enhanced part. It will learn new CNAME features depending on the result of the Naive Bayes part to enhance the ability of the decision part.

![Figure 2. The structure of enhanced NB-Tree.](image)

3.3. The decision tree part
Decision tree is commonly built by recursive partitioning.[20] The key step is to select a single attribute to split for the root of the tree. There are many implementation methods to construct decision tree. In this part, we use ID3[21], which use entropy gain as the criterion to split. The equation to calculate entropy gain as shown below:

$$Gain(A) = \text{Info}(D) - \text{Info}_A(D)$$

(1)

Where \(\text{Info}(D)\) refers to empirical entropy and it can be expressed by equation 2. And \(\text{Info}_A(D)\) refers to conditional entropy, which can be expressed as equation 3.

$$\text{Info}(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

(2)

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

(3)

Where \(m\) is the number of classes, \(p_i\) is the proportion of samples of the \(i\) th category in set \(D\). \(v\) is the number of classes, \(|D_j|\) is the number of samples belonging to \(i\) th category and \(|D|\) is the number of samples in set \(D\).

However, we should not only consider entropy gain but also other factors in the process of constructing decision tree. For example, the obtaining difficulty between CNAME and WHOIS lookup is also an important factor, which is expressed by coefficient \(a\). Finally, the actual equation in our experiment is as follows:

$$GainN(A) = a(\text{Info}(D) - \text{Info}_A(D))$$

(4)

3.4. The naive bayes part
As we all know, the Naive Bayes classifier represents one of the most popular methods for handling classification problems in the AI field. And experiments on real world data have repeatedly shown it to be competitive with much more sophisticated induction algorithms.[22, 23] Assume that all attributes are fully independent, Naive Bayes uses equation 5 to estimate the probability of a test instance \(x\) belonging to class \(c\). 

$$P(c|x) = \frac{P(x|c) P(c)}{P(x)}$$

(5)
\[ P(c|x)_{NB} = P(c) \prod_{j=1}^{m} P(a_j|c) \]  

(5)

Where \( m \) is the number of attributes, \( a_j \) is the \( j \)th attribute value of \( x \), the prior probability \( P(c) \) and the conditional probability \( P(a_j|c) \) are defined using equation 6 and equation 7 respectively. In order to avoid the zero probability problem caused by our dataset, we have performed Laplace smoothing on the equations.

\[ P(c) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1}{n + n_c} \]  

(6)

\[ P(a_j|c) = \frac{\sum_{i=1}^{n} \delta(a_i, a_j, c_i, c) + 1}{\sum_{i=1}^{n} \delta(c_i, c) + n_j} \]  

(7)

As shown in the figure 3, we can see a visualization of the Naive-Bayes classifier for our data. We chose eight features to study. The red area (indicated by the letter r) represents the percentage of samples without features in each type of set. And the green area (indicated by the letter g) represents the percentage of samples with features in each type of set. We can clearly see that compared with f-i features, a-e features are more excellent determiners. Compared with Non-CDN, the probability of f-i features in the CDN dataset is still very large, so they are also used in our research.

![Visualization of the Naive-Bayes classifier](image)

Figure 3. Visualization of the Naive-Bayes classifier for our dataset.

3.5. The enhanced part.

In our experiments, we hope that the Decision Tree part can give more accurate conclusions to reduce the number of request packets and time consumption. To achieve it, we designed an enhancement module that can derive a new CNAME based on the results of the Naive Bayes part.

As we all know, CDN vendors will provide services for many enterprises. Therefore, its CNAME feature will appear in the DNS resolution path of many domains belonging to different enterprises. If such a resolution path is greater than the threshold, we consider it to meet the conditions and add it to the CNAME whitelist.

3.6. The description of the NB-Tree algorithm

The enhanced NB-Tree algorithm that we use to identify CDN IP is shown as table 2. This algorithm is similar to the classic recursive partitioning scheme. The difference is that if the IP address cannot be partially judged by the decision tree, the created leaf node is a Naive Bayes part instead of a node.
predicting a single category. And there is an enhanced part that can improve the detection ability of the decision tree part.

**Algorithm:** Enhanced NB-Tree(D)

**Input:** a test instance x  
**Output:** the class label of x

1) Get CNAME feature and WHOIS feature of the instance x. Construct the feature vector and input it into the decision tree part.  
   if the decision tree part classifies the test sample x as a CDN: return CDN  else go to step 2).  
   end if

2) Get HTTP header of the instance x and encode the HTTP features  
   while \( i \) < number of classes  
      \( P(c_i|x) = P(c) \)  
   while \( j \) < number of features  
      \( P(c_i|x) = P(c_i|x) \times P(a_j|c_i) \)  
   if \( P_{cdn} > P_{not-cdn} \) then return CDN and go to step 3) else return Ordinary IP  
   end if

3) Get the unit information of the domain and CNAME which occurs in the domain name resolution process  
   if the unit of domain is not the unit of CNAME feature, go to step 4)  
   end if

4) if the number of customers of CNAME feature reaches the threshold, add CNAME feature to CNAME White List  
   end if

4. Experiment and evaluation

4.1. Performance measure metrics

To evaluate the experimental results, we adopt 5-fold cross validation method as we only have a relative small dataset. And the final results are calculated by the average results of multiple cross validation. In addition, we utilize three metrics, i.e. ACC, TPR and TNR to measure the performance, namely as follows [24]:

\[
ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad (8)
\]
\[
TPR = \frac{TP}{TP + FN} \quad (9)
\]
\[
TNR = \frac{TN}{TN + FP} \quad (10)
\]

Where \( TP \) refers to the proportion of positive examples correctly classified, \( TN \) refers to the proportion of negative examples correctly classified, \( FP \) refers to the proportion of positive examples misclassified, \( FN \) refers to the proportion of negative examples misclassified.

4.2. Experimental results

4.2.1. Comparison between CNAME feature and WHOIS feature. In our experiments, we compare the recognition capabilities of CNAME feature and WHOIS feature from the perspectives of entropy gain and time delay. 1) Entropy gain. The entropy gains of CNAME and WHOIS are shown in table 2. From the table, we can see that the recognition performance of CNAME features is better. 2) Time delay. We conducted 870 domain name resolution tests and 977 WHOIS query tests to gain the time delay of them. Their time delay distributions are shown in figure 4. The average DNS resolution delay is 0.0238s, and the average WHOIS query delay is 2.9714s by calculating. Therefore, we define the coefficient \( a \) of CNAME as 1 and the coefficient \( a \) of WHOIS as 9.
### Table 3. Entropy Gain of CNAME and WHOIS.

|          | Info(D) | Info(A) | Gain(A) |
|----------|---------|---------|---------|
| CNAME    | 0.9013  | 0.5065  | 0.3948  |
| WHOIS    | 0.9013  | 0.8896  | 0.0117  |

![DNS resolution delay distribution](image1.png)

(a) DNS resolution delay distribution

![WHOIS query delay distribution](image2.png)

(b) WHOIS query delay distribution

**Figure 4.** The time cost of gaining DNS Resolution and WHOIS query.

In summary, the recognition ability of CNAME feature is stronger than WHOIS feature, and its time delay is lower. According to equation 4, we use it as the first classification feature of the decision tree.

#### 4.2.2. Results of Two Classifiers

We conclude an experiment to test the performance of Naive Bayes classifier and enhanced NB-Tree classifier. The experimental results are shown in figure 5. The blue dotted line is the experimental result of enhanced NB-Tree, and the red solid line is the performance of the naive Bayes classifier. We can clearly see that when the TPR is similar, the ACC and TNR results of enhanced NB-Tree perform better than the Naive Bayes classifier. This means that enhanced NB-Tree has a better performance to identify CDN nodes. The main reason is that the decision tree part correctly classified the samples that may be misclassified by the naive Bayes algorithm in advance. Therefore, the FP of enhanced NB-Tree is lower than that of the Naive Bayes algorithm, which will improve the ACC and TNR of enhanced NB-Tree.

![ACC](image3.png)

(a) ACC

![TPR](image4.png)

(b) TPR

![TNR](image5.png)

(c) TNR

**Figure 5.** Comparison of experimental results of enhanced NB-Tree and Naive Bayes.

#### 4.2.3. Performance of enhanced part

We also conduct an experiment to test the effect of the enhanced part. 1484 instances with “alikunlun.com” CNAME feature and 1500 ordinary instance are tested in our experiment. The experimental results as shown in the table 3.

**Table 3.** Comparison of the performance of the NB-Tree without enhanced part and NB-Tree with enhanced part.

|                | Accuracy percent | WHOIS queries times | HTTP request |
|----------------|------------------|---------------------|--------------|
| NB-Tree        | 86.629%          | 4467                | 1513          |
| Enhanced NB-Tree| 99.821%         | 1513                | 1513          |

We can see that on one hand, the accuracy of NB-Tree without enhanced part is around 86.6%, besides it needs more information to make a judgment, which means more time delay and traffic. On the other hand, the accuracy of enhanced NB-Tree is around 99.8%, and it need only 1513 times
WHOIS lookup and HTTP request. As a result, we can conclude that the enhanced part has a very good performance.

We believe that the reason is that the enhanced part can learn new CNAME features, and add new characters to the whitelist, which results in improving the recognition ability of the decision tree part. The new features added by the enhanced part are summarized on the basis of the previous conclusions and are reliable. We use new features to identify subsequent test samples, which can effectively reduce the cost of acquiring additional information and reduce the misjudgment of the Naive Bayes part.

5. Conclusion

CDN detection is the most basic and critical issue in CDN-related research. In view of the limitations of traditional methods of manually constructing key dictionaries, this paper proposes a new method, that is, enhanced NB-Tree to identify CDN IP addresses automatically. It consists of three parts. We build the decision tree part based on the CNAME feather and the WHOIS lookup results. Further, we use the Naive Bayes part to judge the unrecognized IP address. In addition, we improve the ability of the decision tree part by enhanced part, which depends on the result of the Naive Bayes part. It can be seen from the experimental results that our method works well. It can identify CDN IP addresses and the accuracy can reach more than 99%.

Acknowledgment

This work is funded by the National Key Research and Development Program of China under grant 2018YFB0804703.

References

[1] M. Kyryk, N. Pleskanka, M. Pitsyk. QoS mechanism in content delivery network[C]// International Conference on Modern Problems of Radio Engineering Telecommunications & Computer Science(TCSET). IEEE. (2016).

[2] R. Guo, J. Chen, B. Liu, et al. Abusing CDNs for Fun and Profit: Security Issues in CDNs' Origin Validation[C]// 2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS). IEEE, (2018).

[3] C. Huang, A. Wang, J. Li, K. Ross. Measuring and evaluating large-scale CDNs[C]// IMC '08: Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, ACM. (2008).

[4] M. Mahiuddin, K. Ashraffuzzaman, M. A. Rahman. Secure dynamic flow policy for content delivery networks[C]// 15th International Conference on Computer and Information Technology (ICCIT), Chittagong (2012).

[5] S. Hao, Y.B. Zhang, H.N. Wang, et al. End-Users Get Manuevered: Empirical Analysis of Redirection Hijacking in Content Delivery Networks[C]// 27th USENIX Security Symposium (USENIX Security'18), Baltimore (2018).

[6] H.L. Li, L.T. He, H. Zhang, et al. CDN-hosted Domain Detection with Supervised Machine Learning through DNS Records. // ICISS 2020: 2020 The 3rd International Conference on Information Science and System. (2020).

[7] X.X. Chen, G.C. Li, et al. A Deep Learning Based Fast-Flux and CDN Domain Names Recognition Method[C]// 2nd International Conference on Information Science and System (ICISS), Tokyo (2019).

[8] V.K. Adhikari, Y. Guo, F. Hao, et al. Unreeling netflix: Understanding and improving multi-CDN movie delivery[C]. // IEEE INFOCOM Conference. IEEE, Orlando (2012).

[9] V.K. Adhikari, Y. Guo, F. Hao, et al. A tale of three CDNs: An active measurement study of Hulu and its CDNs[C]// IEEE Conference on Computer Communications (INFOCOM). IEEE, Orlando (2012).

[10] L. Daigle, “WHOIS Protocol Specification,” 2004

[11] V.K. Adhikari, Y. Guo, F. Hao, et al. Measurement Study of Netflix, Hulu, and a Tale of Three CDNs[J]. IEEE/ACM Transactions on Networking 23, 1984(2015)
[12] J.A. Xue, D. Choffnes, J. Wang. CDNs Meet CN An Empirical Study of CDN Deployments in China[J]. IEEE Access 5, 5292(2017)
[13] L.M. Wang, X.L. Li, C.H. Cao, et al. Combining decision tree and Naive Bayes for classification[J]. Knowledge-Based Systems 19, 511(2006)
[14] 360 IP range, accessed on January 13, 2021. [Online]. Available: https://wangzhan.qianxin.com/notice/detail/10057
[15] Baidu IP range, accessed on January 13, 2021. [Online]. Available: https://su.baidu.com/help/index.html#/10_changjianwenti/0_HIDE_FAQ/20_baiduyunjiasuji edianIPdizhiduan.md
[16] Alibaba IP Verification tool, accessed on January 13, 2021. [Online]. Available:https://help.aliyun.com/document_detail/146385.html?spm=a2c4g.11186623.2.10 9.79283161hRiY4W
[17] Baidu IP Verification tool, accessed on January 13, 2021. [Online]. Available:https://console.bce.baidu.com/cdn/?_=1611645480058#/cdn/tools/ip
[18] Tencent IP Verification tool, accessed on January 13, 2021. [Online]. Available:https://cloud.tencent.com/document/api/228/37868
[19] Wangsu IP Verification tool, accessed on January 13, 2021. [Online]. Available:https://cdn.console.wangsu.com/v2/index/#/service/ipQuery?CODE=SI_IP_QUE RY
[20] R. Kohavi, Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid[C]/ 2th International Conference on Knowledge Discovery and Data Mining. (1996).
[21] Quinlan J R . Induction of Decision Trees[J]. Machine Learning 1, 81(1986)
[22] W. Iba, K. Thompson, An analysis of bayesian classifiers[C]/10th Conference on Artifificial Intelligence. (1992).
[23] A. Mccallum. A Comparison of Event Models for Naive Bayes Text Classification[C]/ Proc. AAAI-98 Workshop on Learning for Text Categorization. (1998).
[24] B.E. Putro, T. Saepurohman. A Classification Approach to Predicting Beef Knuckle Quality using the Decision Tree and Naves Bayes Method: Case Study: Tiga Bersaudara Factory[C]/ 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA). IEEE, (2020).