CMC vol.59, no.2, pp.591-606, 2019

Street-Level Landmarks Acquisition Based on SVM Classifiers

Ruixiang Li[1, 2], Yingying Liu[3], Yaqiong Qiao[1, 2], Te Ma[1, 2], Bo Wang[4] and Xiangyang Luo[1, 2, *]

Abstract: High-density street-level reliable landmarks are one of the important foundations for street-level geolocation. However, the existing methods cannot obtain enough street-level landmarks in a short period of time. In this paper, a street-level landmarks acquisition method based on SVM (Support Vector Machine) classifiers is proposed. Firstly, the port detection results of IPs with known services are vectorized, and the vectorization results are used as an input of the SVM training. Then, the kernel function and penalty factor are adjusted for SVM classifiers training, and the optimal SVM classifiers are obtained. After that, the classifier sequence is constructed, and the IPs with unknown service are classified using the sequence. Finally, according to the domain name corresponding to the IP, the relationship between the classified server IP and organization name is established. The experimental results in Guangzhou and Wuhan city in China show that the proposed method can be as a supplement to existing typical methods since the number of obtained street-level landmarks is increased substantially, and the median geolocation error using evaluated landmarks is reduced by about 2 km.

Keywords: Landmarks acquisition, SVM, street-level, IP geolocation.

1 Introduction

High-precision IP geolocation has a wide application prospect in protecting social network privacy [Wang, Zhang, Lu et al. (2018)], locating covert communication subjects [Ma, Luo, Li et al. (2018); Zhang, Qin, Zhang et al. (2018); Luo, Song, Li et al. (2016); Wang, Li, Luo et al. (2018)], and supporting indoor localization [Liu, Luo, Liu et al. (2018)]. The landmark-based IP geolocation method is a commonly used and accurate geolocation method. How to obtain abundant street-level landmarks in a short time is a problem that needs to be solved in street-level IP geolocation. Currently, the main methods for obtaining landmarks are databases query and Web-based mining [Guo, Liu, Shen et al. (2009); Zhu, Luo, Liu et al. (2015); Wang, Burgener, Flores et al. (2011); Jiang, Liu and Matthews (2016)].

The landmarks acquisition based on the IP location databases query uses the databases’

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, 450001, China.

[2] Zhengzhou Science and Technology Institute, Zhengzhou, 450001, China.

[3] Henan Institute of Animal Husbandry Economics, Zhengzhou, 450044, China.

[4] State University of New York at Buffalo, New York, 14260-1660, United States.

[*] Corresponding Author: Xiangyang Luo. Email: luoxy_ieu@sina.com.

CMC. doi:10.32604/cmc.2019.05208

API interface to acquire landmarks. Currently, most IP location databases (such as IP.cn [IP.cn (2018)]) provide free API interfaces, and some commercial companies (such as MaxMind [MaxMind (2018)]) provide fee-based API. Using databases' API interface can get a lot of landmarks in a short time. But, the accuracy of the landmarks provided by these databases is only at the city level, and the overall reliability of the database is not high enough [Shavitt and Zilberman (2011); Backstrom, Sun and Marlow (2010); Poese, Uhlig, Kaafar et al. (2011); Li, Zhang, Wang et al. (2017); Li, He, Xi et al. (2015)]. Therefore, it is hard to obtain a large number of reliable street-level landmarks using this method.

Structon, proposed by Guo et al. [Guo, Liu, Shen et al. (2009)], is a method of obtaining landmarks based on Web mining. The authors use regular expressions to extract location information from the webpage, and associate the location information with IP of Web to acquire street-level landmarks. Structon is a typical method based on Web mining, which brings possibility to street-level IP geolocation. After Structon, Zhu et al. [Zhu, Luo, Liu et al. (2015)] proposed a city-level landmark mining method based on Internet forum. Wang et al. [Wang, Burgener, Flores et al. (2011)] acquire organization's location and Web servers according online maps. Jiang et al. [Jiang, Liu and Matthews (2016)] associate the US universities' Web homepages with universities' locations to establish a university website landmark set, according to Wikipedia's list of US universities [Wikipedia (2018)]. Using those methods, a certain number of street-level landmarks can be obtained in a short time, and the efficiency of street-level landmarks acquisition is improved. But because of the limitation of incomplete data collection in online maps, Wikipedia and other collecting organizations, the number of street-level landmarks acquired by these methods is limited.

When acquiring street-level landmarks, the Web-based methods require downloading a web page, parsing the page, and extracting location from web page. When a large number of landmarks are acquired, a large number of page sources need to be downloaded, and a large number of web pages are processed at the same time. Due to the diversity of the web page structure, the time cost of the acquisition method is large. To improve the deficiency, a street-level landmarks acquisition method based on SVM classifiers is proposed. The method is based on the relationship between the network service type and the open ports. Firstly, the ports are used as a service feature to train the SVM classifier. After that, the trained SVM classifiers are used to identify the IP service type. Finally, according to the service domain name of IP, the geographic location of organization is obtained.

The rest of this paper is organized as follows. The related work is introduced in Section II. The principles and steps of street-level landmarks acquisition based on SVM classifiers are elaborated in details in Section III. The experimental results are given in Section IV. Finally, this paper is concluded in Section V.

2 Related work

Structon [Guo, Liu, Shen et al. (2009)] is a method for obtaining landmarks based on Web pages. The core idea is that Web pages are embedded with abundant geographic information (such as organization communication addresses), and associating the geographic information with the IP address of the Web server is landmark. The framework of Structon is shown in Fig. 1.

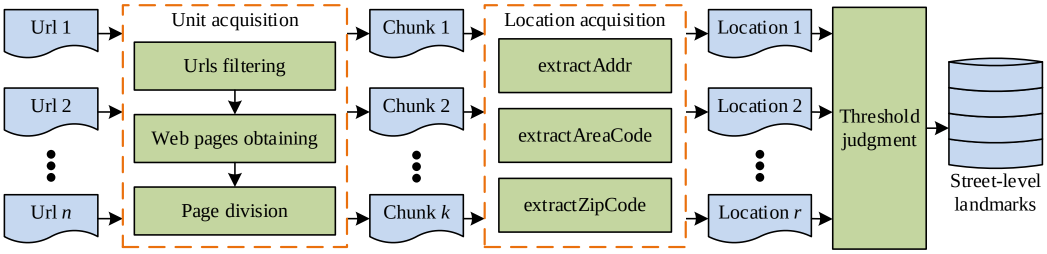


Figure 1: Framework of structon

The method is mainly divided into the following steps:

Step 1: Unit acquisition. lterPage function is called to filter the urls containing “blog”, “bbs”, and “forum”, and the Web pages are obtained using filtered urls. After that, the Web page is divided into a series of chunks according to the HTML tags using splitPage function.

Step 2: Location acquisition. ExtractAddr, extractAreaCode and extractZipCode function are used to try to extract the address, phone area code, and zip code from each chunk. In extractAddr function, the content is checked whether prefix begin with \Addr or its variants (such as \Address and \Contact Addr, etc.). And if so, regular expressions are used to extract location information. In extractAreaCode function, 10 regular expressions are designed to describe the phone number format. If the content is prefixed with \Tel, \Fax or their variant, the phone number is matched with a regular expression that matches the phone number format. Similar to extracting the area code, zipcodes are extracted from the content using extractZipCode function.

Step 3: Threshold judgment. After extracting all the locations from the page, lterLocation function is called. If the number of listAddr, listArea, or listZip is greater than the threshold (In Structon, the threshold is 10), the page is considered to be a yellow page, and the IP of the Web page is not associated with the address information. Otherwise, the location information near the bottom of the page is associated with the IP.

Structon is a method to obtain landmarks using Web pages, and a number of street-level landmarks can be obtained using this method. But due to network bandwidth limitations and the diversity of network structures, the method takes a lot of time to perform a large number of web pages downloading and location extraction. And because it is difficult to obtain massive institutional homepages, Structon is limited to obtain a large number of street-level landmarks.

3 Proposed method

To improve the deficiency of Structon, a street-level landmarks acquisition method based on SVM classifiers is proposed.

3.1 Symbol definition

The symbols definition is as following.

E: network entity. $E=\{IP,lng,lat,grade\}$, and IP, lng, lat and $grade$ represent the IP address, longitude, latitude, and service level of the network entity respectively. If multiple services are provided on one network entity, the service level value takes the highest service value. If the service on the network entity is unknown, the service level value of network entity is zero.

$Port(E_i)$: Open port set. A collection of all open ports of network entity E_i .

$SE(service)$: same service entity set. A set of all network entities which provide same service. $\forall E_i, E_j \in SE(service)$, there is $0 < grade_i = grade_j$, The set of hosts is recorded as $SE(Host)$.

$EPort(SE(service))$: Open port set with service entity. A set of all ports open by all network entities which provide same services. $EPort(SE(service)) = \bigcup_{i=1}^{n} Port(E_i)$, $|SE(service)| = n$.

$SPort(SE(service))$: Service port set. In the range of system port [Cotton, Eggert, Touch et al. (2011)] specified by IANA, the set is obtained according to the list of port numbers of common services [IANA (2018)].

$OPort(SE(service))$: Operation and maintenance port set. A set of operation and maintenance ports opened by not less than 10% of network entities in $SE(service)$.

$$OPort\left(SE\left(service\right)\right) = \left\{ port \middle| \left(\sum_{i=1}^{n} fun\left(port \in Port\left(E_i\right) \cap port \notin SPort\left(SE\left(service\right)\right)\right)\right) \geq 0.2n \right\} ,$$

$|SE(service)| = n$. Where $fun(x)$ is the judgment function, when x is true, then $fun(x) = 1$, and $fun(x) = 0$ vice versa.

$FeatureE\left(service\right)$: Feature network entity. A network entity that best represents the characteristics of a type of services. The network entity may be constructed. The open port $Port\left(FeatureE\left(service\right)\right) = SPort\left(SE\left(service\right)\right) \cup OPort\left(SE\left(service\right)\right)$.

$VP(E_i)$: Port vector. A vector determined by the port open condition of the network entity on the specified port set. The vector dimension is the number of elements in the specified port collection.

$VP\left(E_k\right) \rightarrow SVM_i$: Classification. E_k is classified by SVM_i .

$\left|VP\left(E_k\right) \rightarrow SVM_i\right|$: Classification result.

3.2 Principle of proposed method

The framework of the street-level landmarks acquisition method based on the SVM classifiers is shown in Fig. 2. The method mainly includes three parts: service level determination, IP classification and geographic location mapping. In the part of services level determination, the relationship between *OPort*() and *SPort*() of both services are used, and the level order is the basis of constructing classifier sequence. In the part of IP classification, the SVM classifiers are trained according to the vectorization results of port detection results of IPs with known services, and the IPs with unknown services type are classified by classifier sequence consisting of each services' optimal classifier. In the part of geographic location mapping, according to the characteristics of domain name corresponding to the IP, the relationship between the classified server IP and organization name is established.

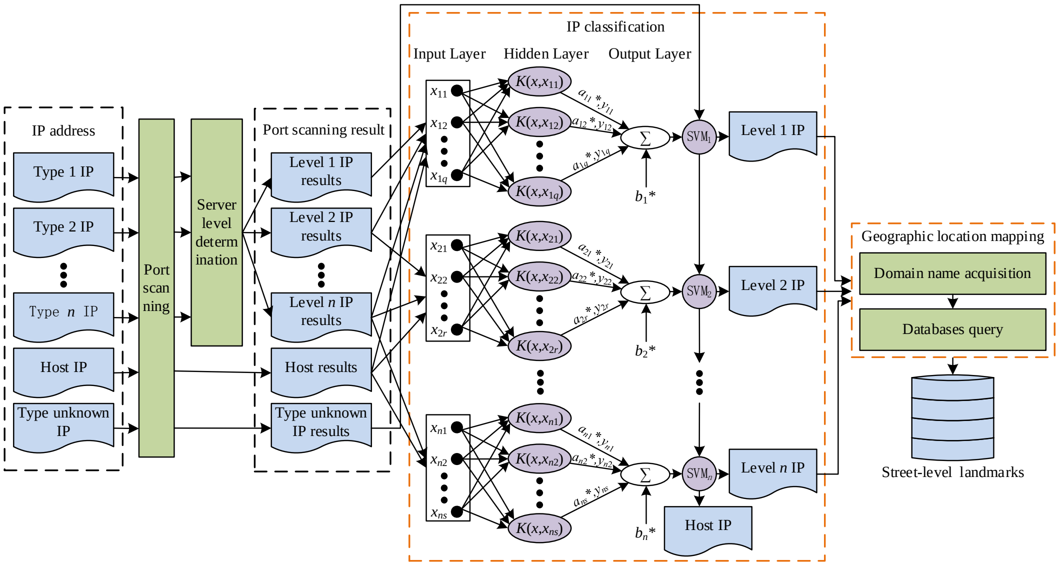


Figure 2: Framework of street-level landmarks acquisition based on SVM classifiers

The method is mainly divided into the following steps:

Step 1: Port scanning. Port scan tools are used to perform an open port scan of the IP address to obtain the port open status of each IP.

Step 2: Services level determining. For the IP address of the known service type, based on *OPort*() and *SPort*() of both services, partially ordered relation is established between two types of services. And the total ordered relation is constructed finally based on all partially ordered relations and conversion regulation. The service level positive integer value is determined by the total ordered relation, and the larger the value is, the lower the service level is.

Step 3: IP classification. The SVM classifier is trained for each service order by the total ordered relation, and all classifiers are constructed as partial binary trees. The IP of the unknown service type is classified using the constructed partial binary tree.

Step 4: Domain name classification. The domain name of the non-host IP is classified into multiple DNS servers using different methods according to the service type. If one IP maps multiple domain name, the multiple relationships between IP and domain name are established. After that, domain names are be classified in the basis of their structural characteristics.

Step 5: Landmarks acquisition. Relationship between the organization's name and the domain name is mapped by multiple databases query.

The services level determining and IP classification are described in detail as the following.

The service level is determined based on the service type and the actual requirement of operation and maintenance. For any two types of services (such as *ser1*, *ser2*), the feature network entities are $FeatureE\left(serv1\right)$ and $FeatureE\left(serv2\right)$ respectively. If Eq. (1) is satisfied,

$$\begin{cases} SPort\left(SE\left(serv1\right)\right) \cap OPort\left(SE\left(serv2\right)\right) = \varnothing \\ OPort\left(SE\left(serv1\right)\right) \cap SPort\left(SE\left(serv2\right)\right) \neq \varnothing \end{cases} \tag{1}$$

Then, the level of *ser1* is higher than *ser2*. That is

$\forall E_i \in SE\left(ser1\right), E_j \in SE\left(ser2\right)$, there is $0 < grade_i < grade_j$.

If Eq. (2) is satisfied

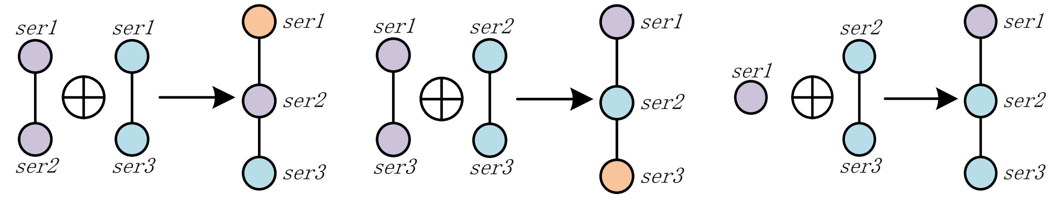
$$\begin{cases} SPort\left(SE\left(serv1\right)\right) \cap OPort\left(SE\left(serv2\right)\right) = \varnothing \\ OPort\left(SE\left(serv1\right)\right) \cap SPort\left(SE\left(serv2\right)\right) = \varnothing \\ OPort\left(SE\left(serv1\right)\right) \cap OPort\left(SE\left(serv2\right)\right) \neq \varnothing \end{cases} \tag{2}$$

Then, the level of *ser1* is of the same as *ser2*. That is

$\forall E_i \in SE\left(ser1\right), E_j \in SE\left(ser2\right)$, there is $0 < grade_i = grade_j$.

Otherwise, the level relationship between *ser1* and *ser2* cannot be determined.

After all services are compared according to the above strategies, the partial ordered relation between services is constructed. The strategy for converting the partial ordered relation to the total ordered relation is shown in Fig. 3.



(a) Same maximal element (b) Same minimal element (c) No association

Figure 3: Total ordered relation conversion strategy

There are three cases when converting two partial ordered relations into a total ordered relation.

Case 1: If the maximal elements of two partial ordered relations are the same, and the minimal elements are different, when the total ordered relation is converted, the maximal element of the total ordered relation is the maximal element of partial ordered relation, and the minimal element of total ordered relation is minimal element arbitrarily selected from two partial ordered relations.

Case 2: If the minimal elements of two partial ordered relations are the same, and the maximal elements are different, when the total ordered relation is converted, the minimal element of the total ordered relation is the minimal element of partial ordered relation, and the maximal element of total ordered relation is maximal element arbitrarily selected from two partial ordered relations.

Case 3: If there is no association between two partial ordered relations, a partial ordered relation is constructed whose maximal element is selected arbitrarily from one partial ordered relation and minimal element is selected from another partial ordered relation. Total ordered relation is converted according to Case 1 and Case 2 among three partial ordered relations.

The maximal element's service level value in the total ordered relation is 1, and the minimal service level value is *n*. The service level value of host is recorded as *n*+1.

IP classification is divided into three sub-steps: vectorization, classifiers training, and classification.

Vectorization. Determine the vector dimension m according to the port scan results of the server IP in training.

$$m = \left| \bigcup_{i=1}^{n} Port\left(FeatureE\left(ser_i\right)\right) \right| \tag{3}$$

Where *n* is the number of service type. For each type of server, the port vector is established based on the open port in the set $\bigcup_{i=1}^{n} Port\left(FeatureE\left(ser_i\right)\right)$. For example, if $\bigcup_{i=1}^{n} Port\left(FeatureE\left(ser_i\right)\right) = \left\{25, 80, 110, 443, 8443\right\}$ and $Port\left(E_i\right) = \{80, 443, 8000, 8443\}$, then $VP(E_i) = (0,1,0,1,1)$.

Classifiers training. When training an SVM classifier, the training set consists of port vectors of two or more types of network entities. Such as when the *i*th $\left(1 \le i \le n\right)$ classifier is trained, the training set $C_i = \left\{VP\left(E_k\right) | \, grade_k \ge i\right\}$ and positive sample set $T_i = \left\{VP\left(E_k\right) | \, grade_k = i\right\}$. The *i*th layer node on the partial binary tree is the *i*th classifier (the root node is regarded as the first layer).

Classification. The training set is classified starting from the root node of the partial binary tree. And the test set is $S = \left\{VP\left(E_k\right) | \, grade_k = 0\right\}$.

When SVM_{i+1} exists,

If $\left|VP\left(E_k\right) \to SVM_i\right| = True$, then $grade_k = i$.

If $\left|VP\left(E_k\right) \to SVM_i\right| = False$, then $VP\left(E_k\right) \to SVM_{i+1}$.

Otherwise,

If $\left|VP\left(E_k\right) \to SVM_i\right| = True$, then $grade_k = i$.

If $\left|VP\left(E_k\right) \to SVM_i\right| = False$, then $grade_k = 0$.

3.3 Analysis about network traffic overhead

In this subsection, network traffic data costing will be analyzed about proposed method and Structon.

Using the crawler to obtain 1000 pure character Web pages (do not carry multimedia data such as video, picture, audio, etc.), the Web page size distribution is shown in Fig. 4.

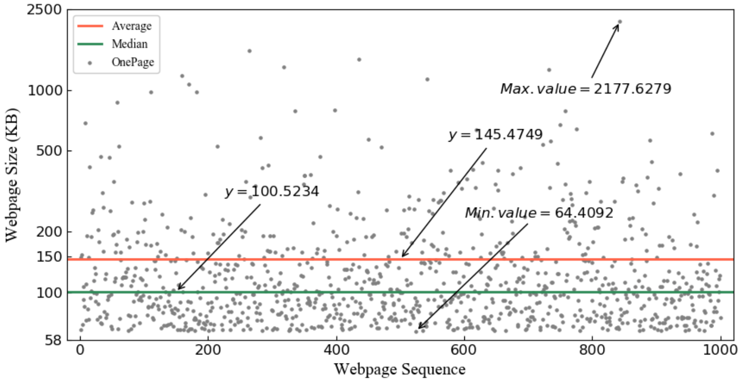


Figure 4: Web page size distribution

In the dataset consist of 1000 Web page size, the minimum value is about 64 KB, and the maximum value is about 2177 KB, and the average value is about 145 KB, and the median is about 101 KB. This means that using the Structon method to obtain a Web page consumes at least 64 KB of network traffic.

According to the structure of data frames and data packets in the TCP/IP protocol cluster, when a TCP port is scanned, the traffic data on the Ethernet is 128 Bytes. When a UDP port is detected by the ICMP error message, the traffic required is 138 Bytes actually.

When the number of detection ports is less than 475, the detection traffic data for one IP is less than 64KB, and the time overhead of Structon is larger than the proposed method in the case of same bandwidth.

4 Experiments

In order to verify the effectiveness of the proposed method, the SVM model parameters optimization experiment and the street-level landmarks acquisition experiment are carried out.

4.1 SVM model parameters optimization experiment

According to IANA [IANA (2018)], the service ports corresponding to DNS, Email, and Web services are shown in Tab. 1.

Table 1: Relationship between service and open ports

Services	Ports
DNS	53,853
Email	25,109,110,143,465,993,995
Web	80,443,591,8008,8080,8443

Nmap is used to detect the open ports ranging from 0 to 49151 for 380 DNS servers IP (280 for training sets and 100 for test sets), 1100 Email servers IP (1000 for training sets and 100 for test sets), 1000 Web server IP (900 for training sets and 100 for test sets), and 1200 host IP (1200 for training sets) respectively. For each type of IP, the *OPort*() is obtained by statistics. According to the service level determination strategy, the partial ordered relations among DNS, Email and Web services are Web<DNS and Web<Email, and finally the total ordered relation Web<Email<DNS is constructed. According to union of *FeatureE*(DNS), *FeatureE*(Email), *FeatureE*(Web) and *FeatureE*(Host) ($SPort(\text{Host}) = \varnothing$ in this paper) shown in appendix, the vector dimension *m*=317 of the training and classification is obtained. Meanwhile, the port scan results are vectorized.

When the kernel functions are linear, rbf and sigmoid respectively, and the penalty factor *C* is 2.0, 1.0, 0.5, and 0.2 respectively, the SVM classifiers of DNS, Email and Web are trained (the classifiers named SVM_{11}, SVM_{12}, and SVM_{13} respectively).

When SVM_{11} is being trained, the training set is

$$\begin{aligned} C_1 = &\left\{VP\left(E_a\right)\middle| E_a \in SE\left(DNS\right)\right\} \cup \left\{VP\left(E_b\right)\middle| E_b \in SE\left(Email\right)\right\} \\ &\cup \left\{VP\left(E_c\right)\middle| E_c \in SE\left(Web\right)\right\} \cup \left\{VP\left(E_d\right)\middle| E_d \in SE\left(Hosts\right)\right\} \end{aligned}$$

Where $1 \le a,b,c,d \le 200$, and the positive sample set is

$$T_1 = \left\{VP\left(E_k\right)\middle| E_k \in SE\left(DNS\right)\right\}, 1 \le k \le 200$$

When training SVM_{12}, the training set is

$$C_2 = \left\{VP\left(E_a\right)\middle| E_a \in SE\left(Email\right)\right\} \cup \left\{VP\left(E_b\right)\middle| E_b \in SE\left(Web\right)\right\} \cup \left\{VP\left(E_c\right)\middle| E_c \in SE\left(Hosts\right)\right\}$$

Where $1 \le a,b,c \le 200$, and the positive sample set is

$$T_2 = \left\{VP\left(E_k\right)\middle| E_k \in SE\left(Email\right)\right\}, 1 \le k \le 200$$

When training SVM_{13}, the training set is

$$C_3 = \left\{VP\left(E_a\right)\middle| E_a \in SE\left(Web\right)\right\} \cup \left\{VP\left(E_b\right)\middle| E_b \in SE\left(Hosts\right)\right\}$$

Where $1 \le a,b \le 200$, and the positive sample set is

$$T_3 = \left\{VP\left(E_k\right)\middle| E_k \in SE\left(Web\right)\right\}, 1 \le k \le 200$$

Use the trained model to classify another 100 known types of servers. The values of True Positive (*TP*), False Positive (*FP*), True Negative (*TN*), and False Negative (*FN*) are obtained by statistics. According to

$$Precision = TP / (TP + FP) \tag{4}$$

$$Recall = TP / (TP + FN) \tag{5}$$

The precision and recall rate of the model under different kernel functions and penalty factors are calculated. The results are shown in Fig. 5 and Fig. 6 respectively.

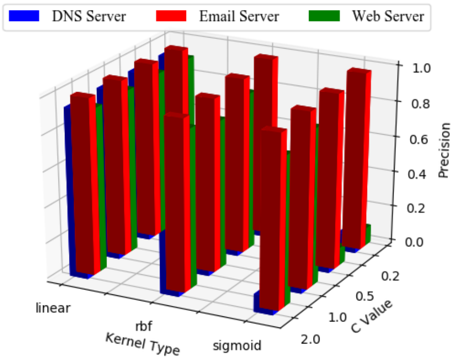


Figure 5: Precision of model under various parameters

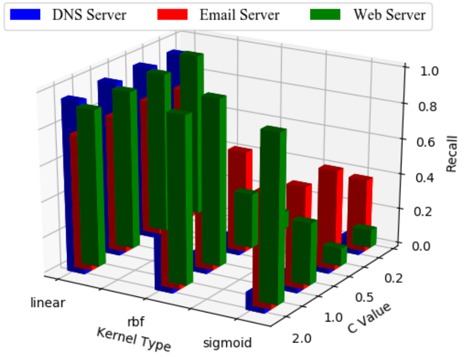


Figure 6: Recall of model under various parameters

The results of precision and recall show that the kernel function linear is better for DNS, Email, and Web server classifiers than other kernel functions. At the same time, when the kernel function is linear, the penalty factor C has less influence on the accuracy and recall results. According to

$$F_1 = 2*\frac{Precision*Recall}{Precision+Recall} \tag{6}$$

When kernel function is linear and penalty factor C=0.2, the maximum average value of F_1 is taken.

4.2 Street-level landmarks acquisition experiment

The results of 380 DNS server IPs, 1100 Email server IPs, 1000 Web server IPs, and 1200 host IPs are selected respectively. According to appendix, the port scanning results are vectorized, and the vector dimension *m*=317 of the training and classification is obtained. The DNS server classifier named SVM_{21}, the Email server classifier named SVM_{22}, and the Web server classifier named SVM_{23} are trained according to the total ordered relation. A partial binary tree is constructed, and the root node is SVM_{21}, and the leaf node is SVM_{23}.

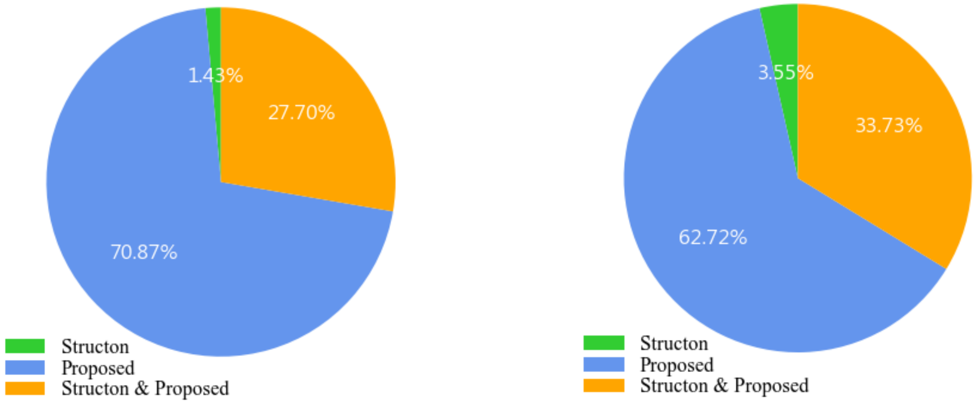
Based on the voting strategy, the IP addresses of Guangzhou and Wuhan are selected from the three location databases of Baidu [Baidu (2018)], IPIP [IPIP (2018)] and IP.cn. The IP numbers are 7028366 and 47728221 and the online IP numbers are 3341747 and 2000357 respectively. According to the port list in appendix, Nmap is used to perform port detection of online IP. Each IP is vectorized based on the port scanning result. All online IPs are classified using the constructed partial binary tree to distinct the IP of the DNS, Email, and Web server. Nslookup command is used to acquire the domain name corresponding to the IP based on different parameters. The IP numbers of the domain names in Guangzhou and Wuhan are 29161 and 25591 respectively. All domain names are sequentially used to obtain institutional information by using database query. Certainly, evaluating the landmarks obtained using street-level landmarks evaluation method [Li, Sun, Hu et al. (2018)] is necessary.

Table 2: Number of IP\landmarks retained at each stage

Step	Guangzhou	Wuhan
Databases Voting	7028366	4772821
IP online test	3341747	2000357
Classification	126899	105113
Domain name query	29161	25591
Institution acquisition	3212	2406
Landmarks evaluation	758	652

As shown in Tab. 2, more than half of IPs in the database is in offline state. After classification, the number of IPs has been greatly reduced due to the exclusion of host IP and router IP.

Then, according to Structon, the names of institutions in Guangzhou and Wuhan are obtained from the regional yellow pages, and 119,466 and 104,853 Web pages are found according to the name of the organization, and 53,749 and 49,928 pre-landmarks are obtained. The number of reliable street-level landmarks after evaluation using the evaluation method is 224 and 252 respectively. The number of duplicated landmarks compared with the landmarks obtained after the evaluation of this method is 213 and 228 respectively. The repetition ratio, comparing the proposed method and Structon, are 95.09% and 90.48% in Guangzhou and Wuhan city respectively. By Structon and proposed method, 769 and 676 landmarks are obtained in Guangzhou and Wuhan city, and the statistical results are shown in Fig. 7

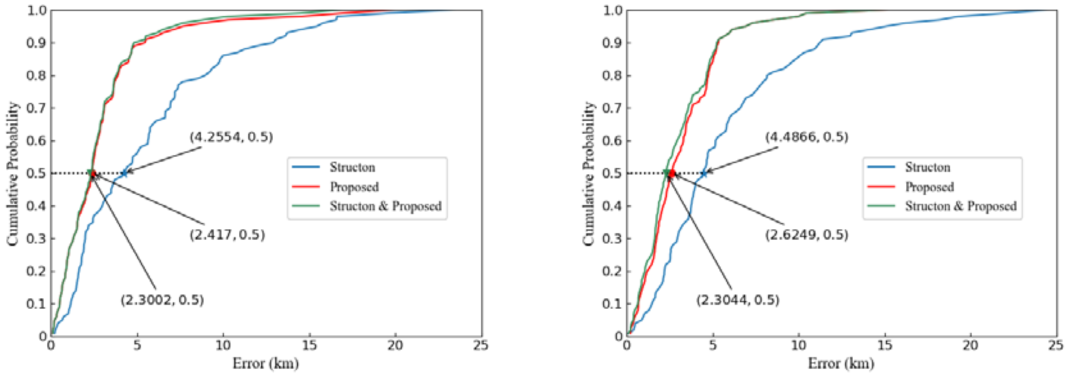


(a) Guangzhou (b) Wuhan

Figure 7: Landmarks statistical results

In Fig. 7, the blue portion indicates the landmarks that can be obtained using the method in this paper, but cannot be obtained using the Structon method. The green portion indicates the landmarks that can be obtained using the Structon method, but cannot be obtained using the method in this paper. And the orange portion indicates the landmarks can be obtained using either of them. In Guangzhou and Wuhan, 769 and 676 landmarks were obtained respectively. Using the method in this paper, more than 95% of the landmarks (sum of blue and orange portion) were obtained, and the number of landmarks obtained by Structon was not more than 40% (sum of green and orange portion). Therefore, this method can be used as a complement to Structon when obtaining abundant street-level landmarks.

200 IPs of known locations in Guangzhou and Wuhan (100 in each city) are geolocated by landmarks obtained by Structon, the method in this paper, and both methods respectively. The relationship between geolocation error and cumulative probability is shown as Fig. 8.



(a) Guangzhou

(b) Wuhan

Figure 8: Relationship between geolocation error and cumulative probability

Fig. 8 shows that after augmenting landmarks using the method in this paper, the median error of geolocation is reduced by about 2 km.

During landmarks acquisition and evaluation, the time overhead in Structon focuses on Web page acquisition, location information extraction and router paths measurement mainly, while the time overhead of the proposed method focuses on ports scanning, classifiers training, databases query and router paths measurement mainly. In the condition of ubuntu16.04 64-bit operating system platform, 4 CPU cores, 8G memory, 1T mechanical hard disk, 10M bandwidth and four processes parallel, the running time of the proposed method is about 36 days and 19 hours, and Structon is about 45 days and 4 hours. The time overhead of the method in this paper is reduced by 18.54% compared with Structon.

The experimental results show that the proposed method, as a supplement to the Structon method, can increase the number of street-level landmarks obtained, and the time overhead is less. And the geolocation accuracy is improved by using the added landmarks.

5 Conclusions

To improve the deficiency of Structon, a street-level landmarks acquisition method based on SVM classifiers is proposed. The method applies the SVM classifiers to the server IP classification, and realizes the association between the server IP and the organization name. The experimental results show that the proposed method can obtain more reliable street-level landmarks in a shorter time. And using the added landmarks to geolocation, the median error of geolocation is reduced. Finding better classifiers to classify IP and better ways to achieve a link between domain names and institutional names is the focus of future work

Acknowledgements: The work presented in this paper is supported by the National Key R&D Program of China [Nos. 2016YFB0801303, 2016QY01W0105], the National Natural Science Foundation of China [Nos. U1636219, U1804263, 61602508, 61772549, U1736214, 61572052] and Plan for Scientific Innovation Talent of Henan Province [No. 2018JR0018].

References

Backstrom, L.; Sun, E.; Marlow, C. (2010): Find me if you can: improving geographical prediction with social and spatial proximity. *19th ACM International Conference on World Wide Web*, pp. 61-70.

Baidu (2018): Web service API. http://lbsyun.baidu.com/index.php?title=webapi/ip-api.

Cotton, M.; Eggert, L.; Touch, J.; Westerlund, M.; Cheshire, S. (2011): BCP 165, RFC 6335. Internet assigned numbers authority (IANA) procedures for the management of the service name and transport protocol port number registry. https://www.rfc-editor.org/rfc/rfc6335.txt.

Dan, O.; Parikh, V.; Davison, B. D. (2016): Improving IP geolocation using query logs. *9th ACM International Conference on Web Search and Data Mining*, pp. 347-356.

Guo, C.; Liu, Y.; Shen, W.; Wang, H. J.; Zhang, Y. (2009): Mining the web and the internet for accurate IP address geolocations. *IEEE International Conference on Computer Communications*, pp. 2841-2845.

Hu, Z.; Heidemann, J.; Pradkin, Y. (2012): Towards geolocation of millions of IP addresses. *Internet Measurement Conference*, pp. 123-130.

IANA (2018): Service name and transport protocol port number registry. https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

IP.cn (2018): IP inquire. https://ip.cn/.

IPIP (2018): IP inquire. https://www.ipip.net/ip.html.

Jiang, H.; Liu, Y.; Matthews, J. N. (2016): IP geolocation estimation using neural networks with stable landmarks. *IEEE Conference on Computer Communications Workshops*, pp. 170-175.

Li, H.; He, Y.; Xi, R.; Wang, Z. (2015): A complete evaluation of the Chinese IP geolocation databases. *8th IEEE International Conference on Intelligent Computation Technology and Automation*, pp. 13-17.

Luo, X.; Song, X.; Li, X.; Zhang, W.; Lu, J. et al. (2016): Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes. *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13557-13583.

Li, H.; Zhang, P.; Wang, Z.; Du, F.; Kuang, Y. et al. (2017): Changing IP geolocation from arbitrary database query towards multi-databases fusion. *22th IEEE Symposium on*

Computers and Communications, pp. 1150-1157.

Liu, W.; Luo, X.; Liu, Y.; Liu, J.; Liu, M. et al. (2018): Localization algorithm of indoor Wi-Fi access points based on signal strength relative relationship and region division. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 71-93.

Li, R.; Sun, Y.; Hu, J.; Ma, T.; Luo, X. (2018): Street-level landmark evaluation based on nearest routers. *Journal on Security & Communication Networks*, vol. 2018, no. 2, pp. 1-12.

Ma, Y.; Luo, X.; Li, X.; Bao, Z.; Zhang, Y. (2018): Selection of rich model steganalysis features based on decision rough set α-positive region reduction. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1.

MaxMind (2018): IP inquire. https://www.maxmind.com/zh/home.

Poese, I.; Uhlig, S.; Kaafar, M. A.; Donnet, B.; Gueye, B. (2011): IP geolocation databases: unreliable? *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53-56.

Siwpersad, S. S.; Gueye, C. A. B.; Uhlig, S. (2008): Assessing the geographic resolution of exhaustive tabulation for geolocating internet hosts. *International Conference on Passive and Active Network Measurement*, pp. 11-20.

Shavitt, Y.; Zilberman, N. (2011): A geolocation databases study. *Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 2044-2056.

Wang, Y.; Burgener, D.; Flores, M.; Kuzmanovic, A.; Huang, C. (2011): Towards street-level client-independent IP geolocation. *8th USENIX Conference on Networked Systems Design and Implementation*, pp. 365-379.

Wang, J.; Li, T.; Luo, X.; Shi, Y. Q.; Jha, S. K. (2018): Identifying computer generated images based on quaternion central moments in color quaternion wavelet domain. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1.

Wang, Q.; Zhang, Y.; Lu, X.; Wang, Z.; Qin, Z. et al. (2018): Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 591-606.

Wikipedia (2018): Lists of American institutions of higher education. https://en.wikipedia.org/wiki/Lists of American institutions of higher education.

Zhu, G.; Luo, X.; Liu, F.; Chen, J. (2015): An algorithm of city-level landmark mining based on internet forum. *18th IEEE International Conference on Network-Based Information Systems*, pp. 294-301.

Zhang, Y.; Qin, C.; Zhang, W.; Liu, F.; Luo, X. (2018): On the fault-tolerant performance for a class of robust image steganography. *Signal Processing*, vol. 146, pp. 99-111.

Appendix: Feature ports of different services

FeatureE(DNS)={20, 22, 53, 80, 88, 111, 139, 389, 443, 445, 465, 514, 555, 587, 636, 995, 1001, 1433, 2000, 2144, 2200, 2222, 2260, 2383, 2601, 3268, 3390, 4045, 4444, 4567, 5666, 6112, 8008, 8081, 8087, 8291, 8443, 8888, 9010, 9090, 9102, 9500, 9999, 32774, 32776, 32778}

FeatureE(Email)={13, 17, 20, 21, 22, 53, 80, 82, 84, 89, 99, 110, 139, 143, 199, 211, 443, 445, 464, 465, 500, 514, 563, 587, 593, 636, 800, 843, 873, 902, 990, 993, 995, 999, 1022, 1027, 1030, 1031, 1032, 1047, 1048, 1051, 1053, 1060, 1063, 1080, 1089, 1098, 1110, 1218, 1433, 1723, 1755, 1863, 1935, 2001, 2049, 2106, 2323, 2383, 2500, 2800, 3000, 3260, 3261, 3268, 3306, 3690, 4343, 4443, 4899, 5000, 5060, 5080, 5222, 5432, 5678, 5900, 5903, 5989, 6000, 6002, 6004, 6100, 6129, 6666, 6789, 7000, 7777, 7778, 7999, 8001, 8007, 8008, 8009, 8022, 8080, 8081, 8083, 8085, 8087, 8089, 8099, 8194, 8291, 8443, 8649, 9003, 9010, 9081, 9091, 9900, 9998, 9999, 10001, 10009, 10025, 20000}

FeatureE(Web)={21, 23, 26, 80, 82, 84, 88, 90, 100, 111, 119, 135, 139, 211, 222, 280, 389, 443, 445, 465, 541, 555, 625, 631, 636, 800, 808, 843, 873, 888, 901, 990, 993, 1000, 1002, 1010, 1025, 1027, 1029, 1031, 1033, 1034, 1035, 1040, 1043, 1081, 1089, 1098, 1099, 1102, 1104, 1201, 1720, 1723, 2000, 2002, 2004, 2006, 2030, 2049, 2100, 2103, 2105, 2111, 2222, 2301, 2383, 2601, 2638, 3030, 3268, 3306, 3690, 4445, 5000, 5030, 5080, 5190, 5222, 5432, 5500, 5550, 5560, 5631, 5666, 5900, 5989, 5999, 6001, 6566, 7000, 7002, 7070, 7200, 7443, 7777, 8000, 8002, 8007, 8009, 8011, 8022, 8080, 8082, 8084, 8086, 8088, 8089, 8090, 8100, 8180, 8200, 8291, 8300, 8600, 9002, 9009, 9011, 9071, 9080, 9091, 9100, 9110, 9200, 9207, 9502, 9998, 10000}

FeatureE(Host)={6, 21, 22, 23, 80, 82, 89, 100, 110, 119, 179, 222, 264, 443, 513, 555, 587, 646, 800, 873, 880, 900, 992, 1025, 1027, 1029, 1031, 1080, 1108, 1111, 1112, 1122, 1218, 1234, 1236, 1248, 1352, 1433, 1443, 1521, 1723, 1812, 1863, 1947, 1999, 2001, 2003, 2009, 2013, 2020, 2022, 2043, 2100, 2107, 2111, 2170, 2179, 2200, 2222, 2323, 2366, 2382, 2500, 2601, 2602, 2800, 3000, 3005, 3007, 3030, 3221, 3301, 3306, 3322, 3333, 3389, 3800, 3986, 4001, 4003, 4004, 4005, 4045, 4443, 4662, 4900, 5000, 5002, 5003, 5004, 5009, 5050, 5051, 5061, 5100, 5678, 5900, 5902, 5903, 5904, 5922, 5960, 6000, 6002, 6004, 6006, 6129, 6389, 6666, 6668, 6689, 6969, 7001, 7007, 7070, 7778, 8000, 8002, 8008, 8010, 8011, 8021, 8080, 8082, 8084, 8089, 8099, 8181, 8200, 8222, 8383, 8500, 8800, 9001, 9003, 9010, 9080, 9090, 9100, 9102, 9500, 9502, 9900, 9917, 9929, 9998, 10000, 10002, 10003, 10009, 10010, 10012, 10082, 11110, 12000, 12265, 12345, 15003, 16016, 19801, 20005, 20031, 20221, 27000, 30000, 44443, 48080}

FeatureE(DNS) ∪ *FeatureE*(Email) ∪ *FeatureE*(Web) ∪ *FeatureE*(Host)={6, 13, 17, 20, 21, 22, 23, 26, 53, 80, 82, 84, 88, 89, 90, 99, 100, 110, 111, 119, 135, 139, 143, 179, 199, 211, 222, 264, 280, 389, 443, 445, 464, 465, 500, 513, 514, 541, 555, 563, 587, 593, 625, 631, 636, 646, 800, 808, 843, 873, 880, 888, 900, 901, 902, 990, 992, 993, 995, 999, 1000, 1001, 1002, 1010, 1022, 1025, 1027, 1029, 1030, 1031, 1032, 1033, 1034, 1035, 1040, 1043, 1047, 1048, 1051, 1053, 1060, 1063, 1080, 1081, 1089, 1098, 1099, 1102, 1104, 1108, 1110, 1111, 1112, 1122, 1201, 1218, 1234, 1236, 1248, 1352, 1433, 1443, 1521, 1720, 1723, 1755, 1812, 1863, 1935, 1947, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2009, 2013, 2020, 2022, 2030, 2043, 2049, 2100, 2103, 2105, 2106, 2107, 2111, 2144, 2170, 2179, 2200, 2222, 2260, 2301, 2323, 2366, 2382, 2383, 2500, 2601, 2602, 2638, 2800, 3000, 3005, 3007, 3030, 3221, 3260, 3261, 3268, 3301, 3306, 3322, 3333, 3389, 3390, 3690, 3800, 3986, 4001, 4003, 4004, 4005, 4045, 4343, 4443, 4444, 4445, 4567, 4662, 4899, 4900, 5000, 5002, 5003, 5004, 5009, 5030, 5050, 5051, 5060, 5061, 5080, 5100, 5190, 5222, 5432, 5500, 5550, 5560, 5631, 5666, 5678, 5900, 5902, 5903, 5904, 5922, 5960, 5989, 5999, 6000, 6001, 6002, 6004, 6006, 6100, 6112, 6129, 6389, 6566, 6666, 6668, 6689, 6789, 6969, 7000, 7001, 7002, 7007, 7070, 7200, 7443, 7777, 7778, 7999, 8000, 8001, 8002, 8007, 8008, 8009, 8010, 8011, 8021, 8022, 8080, 8081, 8082, 8083, 8084, 8085, 8086, 8087, 8088, 8089, 8090, 8099, 8100, 8180, 8181, 8194, 8200, 8222, 8291, 8300, 8383, 8443, 8500, 8600, 8649, 8800, 8888, 9001, 9002, 9003, 9009, 9010, 9011, 9071, 9080, 9081, 9090, 9091, 9100, 9102, 9110, 9200, 9207, 9500, 9502, 9900, 9917, 9929, 9998, 9999, 10000, 10001, 10002, 10003, 10009, 10010, 10012, 10025, 10082, 11110, 12000, 12265, 12345, 15003, 16016, 19801, 20000, 20005, 20031, 20221, 27000, 30000, 32774, 32776, 32778, 44443, 48080}