GeoBLR: Dynamic IP Geolocation Method Based on Bayesian Linear Regression

Fei Du[1,2], Xiuguo Bao[3], Yongzheng Zhang[1,2](✉), and Yu Wang[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences,
Beijing, China
{dufei,zhangyongzheng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China
[3] National Internet Emergency Center, CNCERT/CC,
Beijing, China

Abstract. The geographical location of dynamic IP addresses is important for network security applications. The delay-based or topology-based measurement method and the association-analysis-based method improve the median estimation accuracy, but are still affected by the limited precision (about 799 m) and the longer response time (tens of seconds), which cannot meet the location-aware applications of high-precise and real-time location requirements, especially the position of dynamic IP addresses. In this paper, we propose a novel approach for dynamic IP geolocation based on Bayesian Linear Regression, namely, *GeoBLR*, which exploits geolocation resources fundamentally different from existing ones. We exploit the location data that users would like to share in location sharing services for accurate and real-time geolocation of dynamic IP addresses. Experimental results show that compared to existing geolocation techniques, *GeoBLR* achieves (1) a median estimation error of 239 m and (2) a mean response time of 270 ms, which are valuable for accurate location-aware network security applications.

Keywords: Network security · Dynamic IP geolocation · Machine learning · Bayesian Linear Regression

1 Introduction

The ability to accurately identify the geographic of location of an internet IP address has significant implications for network security analysts (e.g. credit card fraud protection), security event forensics and law enforcement [13]. A striking amount of malicious activities have been reported from dynamic IP addresses space, such as spamming, botnets, etc. [18]. Consequently, The dynamic IP geolocation has become increasingly important in finding and preventing fast growing

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 310–328, 2019.
https://doi.org/10.1007/978-3-030-12981-1_22

network attacks, which can help law enforcement organizations and government agencies to identify the location information or network attack resources of criminals.

The dynamic IP geolocation is a challenging task because of insufficient labelled training data. The variability of the dynamic IP addresses and the excessive size of the dynamic IP addresses network make the task even harder. It is far more challenging to determine an dynamic IP address with fine-grained granularity without information from the Internet Service Provider (ISP).

In this paper, we study the geographic location of dynamic IP addresses, especially, focus on the dynamic behavior of IP addresses assignment [32]. There are two-fold reasons for geolocation inaccuracy. First, the adjustment period of dynamic IP addresses is *ephemeral*, assignment through DHCP protocol. As a result, the same place have observed different IP addresses, even if the observation interval is continuous within a span of 10 to 120 min. Second, the dynamic IP addresses for the same place are *itinerant*—similar IP addresses will be randomly assigned to the same place in consecutive period of time. In other words, dynamic IP addresses do not embed fine-grained information on the device with used one. Consequently, the positioning of dynamic IP addresses and the positioning of devices are intrinsically two different problem.

In existing work towards IP geolocation, Database-driven approaches typically build a database whose geolocation information come from the Whois database [1], DNS [27], user contributions [4], etc. These databases are compiled by combining data from different sources. Database-driven geolocations [1,2,4–7,27] are fast response time. Whereas, such IP/location mappings are very coarse-grained and usually achieve a city-level precision in most cases. Delay measurement based geolocation approaches such as GeoGet [21], Octant [31] and SLG [30], they have (1) high deployment cost, and (2) long response time, which cannot meet the real-time requirement of dynamic IP addresses. Statistical and data mining approaches [9,14,33] are implemented by applying kernel density estimation to delay measurement and using maximum likelihood estimation to distance from landmark. The main purpose of the machine learning approach (GeoCop [29]) is to improve the accuracy and robustness of existing geolocation methods. HG-SOM [17] is an advanced approach for an accurate and self-optimization model for IP geolocation, including identification of optimized Landmark positions. Moreover, the selection of correlated data and the estimated target location requires a sophisticated strategy to identify the correct position. These approaches also cannot meet the demand of fine-grained granularity.

Briefly, our approach, referred to as *GeoBLR*, is based on the fact that most of the IP addresses allocated to the home broadband access adopt dynamic IP address access technology. In this scenario, user end-hosts (such as mobile-phones, pads, laptops, etc.) access the network using wireless technology. The geographical location provided by the APP applications is used as a measurement landmark. Since the distance between the user's mobile phone and the IP access point (AP) is within the coverage of the wireless network, the geographical location within the range may be used as the geographic location of the access IP in the current time period.

The method includes the three phases: First, based on the observation that users are willing to share in location-sharing services by their Global Positioning System (GPS) units, we cluster the location fingerprint at various time periods respectively. Then estimate the candidate landmarks of each IP address. After this phase, we obtain the mappings from each individual IP address to its location candidates. Secondly, we use Bayesian Inference to calculate the maximum posterior probability of candidate multiple landmarks of dynamic IP address, select the geographical location with the highest probability, according to the semantics of the scenario. Next, We use Bayesian Linear Regression to optimization parameters, and correct the trusted landmark database. Finally, based on the mappings obtained from the first two phase, we design an classification model to estimate the mappings from dynamic IP addresses to geographical locations.

Our contribution in this paper is three-fold:

- We propose a novel approach to locate dynamic IP addresses that we call *GeoBLR*. The proposed approach has strong adaptive ability to data, can repeatedly use experimental data, and effectively prevent over-fitting, which can meet the demand of high-precision and real-time positioning.
- We use the largest convex polygon to cover the location area as the candidate set of landmarks, instead of adopting the center point of the *k-means* clustering algorithm, which is consistent with the real IP access scenario.
- We give a formal definition of dynamic IP address geolocation, adding the time attribute to the location description, which is more closer to the real network environment.

The paper is organized as follows: in Sect. 2 discusses related work. Section 3 gives definitions and relevant problem statements. Section 4 explains our algorithm in detail, including mathematical proof, feature extraction and analysis process. Section 5 describes the dataset and presents experimental results and comparative analysis. Finally, draws the conclusions in Sect. 6.

2 Related Work

To evaluate performance of the *GeoBLR* algorithm, we compare against current relevant geolocation approaches.

Database-Driven Geolocation. These approaches try to establish a database with large number of IP/location mapping records, whose geolocation resources come from the Whois database [1], DNS [27], postal addresses from the website [26], user registration records [28]. Such as MaxMind [6], IP2Location [5], Neustar [7] and Digital Element [2]. We will compare geolocation accuracy with both the Maxmind database [6] or the IP2Location database [5]. Both of these databases are commercially available IP lookup packages. Unfortunately, these databases are hard to maintain and keep up-to-date, especially, since it cannot take into consideration dynamic IP assignment.

Data-Mining-Based Geolocation. These approaches try to discover the relationship between location and IP addresses from location-share applications, websites, query-logs and so on. One of the latest data mining-based geolocation technique—Checkin-Geo [24]—can achieve the median error distance to 799 m, which is very prominent. It first obtain that relationship data of “User ↔ Home/office Locations” from the application of some mobile phone application, then obtain the data of “User ↔ Home/office IPs” from the corresponding PC application program, and finally obtains the rule of user activity by machine learning method, and establishes the relationship of “IP ↔ Location” to achieve the target IP location. However, The data resources on which such technologies depend are usually only abundant in a few metropolitan areas and the public cannot access these resources due to privacy concerns. Besides, data mining-based technologies such as [16] are difficult to cover most IP addresses, so they are mapping IP address blocks to the one landmark in order to increase the coverage of IP addresses, which may result in larger positioning errors according to [15]. As compared with previous data-mining-based technologies, Structon [16] uses a new approach to obtaining IP geolocation from the website. In particular, it builds a Geo-IP lookup table and extracts location information using regular expression technique from each page from a large crawler crawling database. Since Structon does not combine delay-based measurement algorithm with the landmarks discovered, it can achieve city-level coarser geolocation granularity. Dan et al. [12] use query-logs-based technology to improve the accuracy of IP address location. It is a supplement and enhancement to the existing IP geolocation database. The main challenges by this technology are: (1) extracting explicit location information in the logs, (2) the query logs are a large scale and belong to CPU-intensive calculation, (3) for a given IP range, multiple candidate landmarks are extracted from the query logs, (4) the metropolises with large influence need to be modified to the surrounding small towns.

Statistical-Based Geolocation. Recent relevant approaches (e.g., [9,14,33]) that find the maximum likelihood probability of geographic location with respect to observed delay-distance measurements. While the construction of the probability distributions varies, e.g., nonparametric kernel density estimators in [14,33], parametric log-normal distributions in [9]) etc., All three methods assume conditional independence between measurements, in order to efficiently calculate the geographic location using the maximum likelihood probability. Spotter [33] leverage a probabilistic approach based on a statistical analysis of the relationship between network delay and geographic distance. [14] regards IP geolocation as classification problem based on machine learning, which makes it possible to incorporate other location information into the framework.

Wireless-Based Geolocation. These wireless-based approaches use GPS, WiFi, cellular and other wireless positioning systems as the source. The GPS-based geolocation method is widely used in mobile phones, computers and various embedded systems. The cellular and WiFi-based location algorithms include Google’s My Location [3] and Skyhook [8]. In particular, cellular-based geolocation provides users an estimated location through triangulation, while

Wi-Fi-based geolocation uses Wi-Fi access point as targeted location. These methods require the user's permission to share their locations.

Crowdsourcing-Based Geolocation. These Crowdsourcing-based approaches collect and process data based on crowd-sourcing principles. Ciavarrini et al. [10,11] proposed a method based on the crowd-sourcing principle to use the GPS positioning module in the mobile phone as a landmark. (1) Using the mobile phone as a landmark, its built-in GPS unit has a self-positioning function, (2) Considering the wireless connection in the delay-distance model, (3) Participating in the crowdsourced device through the Portolan platform, not only from the research institutions, it also comes to the real application environment. They also discussed the effects of four different delay-distance models on IP address location errors. Lee et al. [20] propose an IP address database construction method based on location-labelled Internet broadband performance measurement tool, and provide an IP geolocation database based on South Korea's 7-year Internet broadband performance data, which shows fine-grained granularity but only limited to South Korea.

All of the previously mentioned approaches rely on delay-based or topology-based measurements [23,34,35] and a lot of data analysis [22,25], and most methodologies do not take into account the dynamic IP addresses. In such scenarios, the measurement process is neither stable nor reliable. Therefore, this paper presents a novel approach, *GeoBLR*, which uses the location information shared by the user to solve the physical location of dynamic IP addresses and achieves the purpose of (1) a negligible response time and (2) a smaller than existing approaches of median estimation error.

3 Problem Statement

In this section, we present the definitions of relevant concepts and the formalized description of the problem.

Definition 1. *(Location Fingerprint). It is location information containing the spatial location description generated by the mobile device using GPS, cellular and Wi-Fi. A typical location fingerprint ω is a tuple (t, lat, lng, co, c, as), where t is time, lat is latitude, lng is longitude, co is the coordinate system where the latitude and longitude is located, c is the city where it is located, and as is the acquisition method of the location fingerprint (GPS, cellular, Wi-Fi) etc.*

Definition 2. *(Active dynamic IP addresses set $\mathcal{P}$). We define all the dynamic IP address of the broadband access user, does not include the private IP address dynamically allocated within the Network Address Translation (NAT) and the IP address used for the mobile phone.*

Definition 3. *(The landmark set $\mathcal{L}$). It is a geographical location set. A typical geographic location l is a tuple (lat, lng, co, des, r, c, p, n), where lat is latitude, lng is longitude, and co is the coordinate system in which the latitude and longitude is located, des is the semantic description of the location (may be the specific* *building name, or the location description relative to the POI point, or the unit name), r is the community (street level), and c is the city. p is the province or state in which it is located, and n is the country in which it is located.*

Definition 4. *(The coverage area of the largest convex polygon ζ). It is the polygon area covered by the set C of the observed location fingerprints ω_t in a corresponding coordinate system within a period of time t. Its area should cover the C.*

The area (ζ) is a convex polygon, not a circular area formed by the largest diameter in the set C. The convex polygon is more computationally efficient than the area covered by the circular area, and the acquisition of the covered building is more representative than the circular area.

Problem. The geolocation of dynamic IP addresses, given the IP address in the active dynamic IP addresses set $\mathcal{P}$, finds that there may be multiple different dynamic IP addresses in the corresponding geographical location ℓ_t ($\ell_t \in \zeta$) within the time t, and the same IP address may also correspond to different ℓ_t in different time periods t.

4 Bayesian Linear Regression-Based Location

Compared with active measurement, passive data acquisition can effectively reduce the cost of data acquisition, it also introduces some new problems, such as (1) the data "jitter" caused by the hybrid WLAN access and the heterogeneity of the terminal mobile phone. The offset of location fingerprint caused by different operations system, application and driver; (2) data collection depends on the active participation of users, there is regional imbalance, some areas are data intensive, and some areas are sparse. These factors objectively cause measurement errors in IP geolocation. In addition, the configuration strategies of different ISPs, the heterogeneity and diversity of broadband access networks also cause the "last half mile" problem.

In problems where we have limited data or have some prior knowledge that we want to use in our model, the Bayesian Linear Regression approach can both incorporate prior information and show our uncertainty. Bayesian Linear Regression reflects the Bayesian framework: we form an initial estimate and improve our estimate. And as we gather more evidence, our model becomes less wrong. Bayesian reasoning is a natural extension of our intuition.

Since the maximum likelihood estimation always makes the model too complicated to produce a phenomenon of over-fitting, Bayesian Linear Regression can not only solve the over-fitting problem in the maximum likelihood estimation, but also it could make better use of the data sample fully. The training model can effectively and accurately determine the complexity of the model.

Base on the above analysis, the processing flow of our algorithm is shown in Fig. 1. This process can be divided into three phases, which are preprocessing phase, calibration phase and geolocation phase. The complete *GeoBLR* geolocation methodology is presented in Algorithm 1.

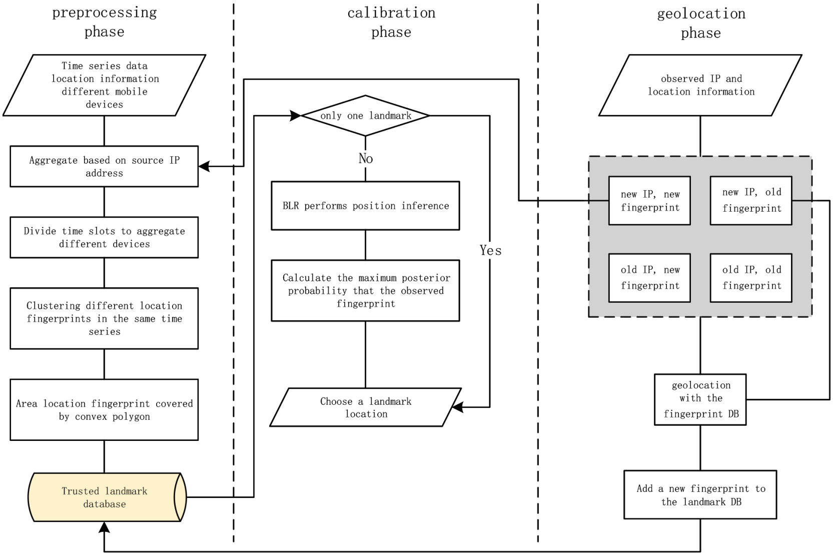

preprocessing phase
calibration phase
geolocation phase
Time series data location information different mobile devices
Aggregate based on source IP address
Divide time slots to aggregate different devices
Clustering different location fingerprints in the same time series
Area location fingerprint covered by convex polygon
Trusted landmark database
only one landmark
No
Yes
BLR performs position inference
Calculate the maximum posterior probability that the observed fingerprint
Choose a landmark location
observed IP and location information
new IP, new fingerprint
new IP, old fingerprint
old IP, new fingerprint
old IP, old fingerprint
geolocation with the fingerprint DB
Add a new fingerprint to the landmark DB

Fig. 1. Flow chart of the proposed *GeoBLR* geolocation process.

```
Algorithm 1. GeoBLR Geolocation Algorithm
Input:
    A set of $N$ record information generated by $m$ mobile devices,
    $\mathcal{T} = \left\{T_1^{D_1}, T_2^{D_2}, ..., T_m^{D_m}\right\}$, where $T_i^{D_i} = \{s_1, s_2, ..., s_{D_i}\}$
Output:
    The mapping between dynamic IP addresses and landmarks, $\mathcal{P} \longleftrightarrow \mathcal{L}$
cluster location fingerprints $C$ from set $\mathcal{T}$, then deriving the convex polygon region $\zeta$ from $C$;
construct a landmark dataset $\mathcal{L}$ from the convex polygon area $\zeta$,
    $\mathcal{L} = \{\ell_1, \ell_2, ..., \ell_k, ...\ell_n\}$, $(n \geq 3)$;
for each IP $\in \mathcal{P}$ do
    train parameters $[\beta_1, \beta_0]$ using Bayesian linear regression on the data set $\mathcal{T}$ to maximize the probability of Eq. (8);
    obtain the maximum value of posterior probability using Eq. (2),
    associate IP$\leftrightarrow \ell$;
end for
given an IP and observed location fingerprint $\omega$;
if IP $\in \mathcal{P}$ and $\omega \in \mathcal{L}$ then
    return $\ell \longleftarrow \omega$;
end if
if (IP $\in \mathcal{P}$ and $\omega \notin \mathcal{L}$) or (IP $\notin \mathcal{P}$ and $\omega \in \mathcal{L}$) then
    update the $\mathcal{L}$ and $\mathcal{P}$;
    return $\ell \longleftarrow \omega$;
end if
if IP $\notin \mathcal{P}$ and $\omega \notin \mathcal{L}$ then
    enter other procedure to process;
end if
```

4.1 Preprocessing Phase

In the preprocessing phase, data is aggregated by source IP addresses. In the same IP address, the data is divided into multiple time series according to time t. In the same time period, multiple identical mobile device information will appear, and the device information needs to be similarly measurement. We use the bottom-up hierarchical clustering model (DBSCAN algorithm) to compare the device IDs of N records, and the clusters that successfully clustered will enter the next stage. Eventually, N records will be divided into m different devices.

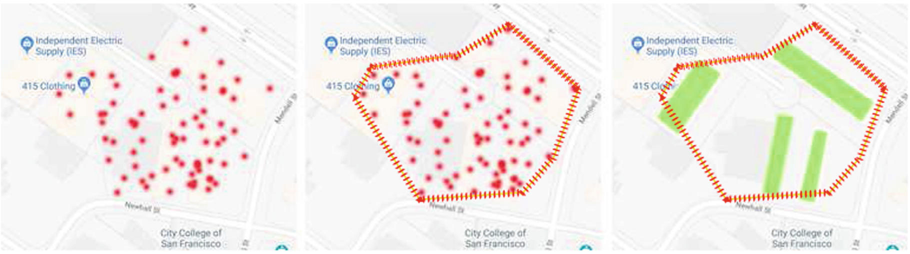

Independent Electric Supply (IES)
415 Clothing
Newhall St
City College of San Francisco
Mendell St
Independent Electric Supply (IES)
415 Clothing
Newhall St
City College of San Francisco
Mendell St
Independent Electric Supply (IES)
415 Cl
Newhall St
City College of San Francisco
Mendell St

Fig. 2. Mapping between location fingerprints and physical location landmarks.

For multiple records of the same device, the set $T_i = \{s_1, s_2, ..., s_D\}$ is the D records generated by the mobile device i within the time t. We perform density-based clustering on the location fingerprints of these D records, and select a representative position record s_j of the cluster center as the position fingerprint of all the records of the D records. In this step, we mainly remove the abnormal positioning fingerprint, which improves the validity of the data.

Time series data generated for m different mobile devices in time t, i.e., $\left\{T_1^{D_1}, T_2^{D_2}, ..., T_i^{D_i}, ..., T_m^{D_m}\right\}$, extract its location fingerprint set C and draw its largest convex polygon area in the map ζ, the area of the convex polygon should cover all locations of the fingerprint point set ($\zeta \supseteq C$). As shown in Fig. 2.

A collection of location landmarks contained in a convex polygon region $\mathcal{L}$, $\mathcal{L} = \{\ell_1, \ell_2, ..., \ell_k, ...\ell_n\}$, n is the number of landmarks ($n \geq 3$). In this way, the location fingerprint in the collected data is mapped to the physical location, and a trusted database of location fingerprints will be constructed.

4.2 Calibration Phase

In the correction phase, the transformed position fingerprint and source IP are mainly used as input of the positioning algorithm to process the location inference. Considering high-precision and real-time, here we use Bayesian linear regression algorithm, which is very suitable for passively acquired data for its simplicity and high accuracy.

$$P\left(\ell_r|\omega\right)=\frac{P\left(\omega|\ell_r\right)P\left(\ell_r\right)}{\sum_{i=1}^{n}P\left(\omega|\ell_i\right)P\left(\ell_i\right)} \tag{1}$$

$P\left(\ell_r|\omega\right)$ indicates the posterior probability of the occurrence of observing fingerprint, then $P\left(\ell_r\right)$ indicates the prior possibility of a landmark. In the calibration phase, the probability of landmark i and landmark j is not the same $(i \neq j)$ according to the specific context semantics. We will assign a plurality of landmarks in the convex polygon region according to the semantics: $P\left(\ell_r\right)=\rho_r\frac{m}{n}$, which $\sum_{r\in n}\rho_r=1$, ρ is the weight of the semantics. m is the number of occurrences of the random position ℓ_r in n observations. the value of $\sum_{i=1}^{n}P\left(\omega|\ell_i\right)P\left(\ell_i\right))$ is usually 1, Therefore the estimated location ℓ_k is the one obtaining the maximum value of the posterior probability.

$$\ell_k=\arg\max_{r}P\left(\omega|\ell_r\right) \tag{2}$$

Supposing each location fingerprint $\omega=(o_1,o_2,...,o_m)$ and it has M values, then the $P\left(\omega|\ell_r\right)=\prod_{i\in M}P_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)$, bring it into Eq. (2).

$$\begin{aligned}\ell_k&=\arg\max_{r}P\left(\omega|\ell_r\right)\\&=\arg\max_{r}\prod_{i\in M}P_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)\propto\ln\left(\arg\max_{r}\prod_{i\in M}P_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)\right)\\&=\arg\max_{r}ln\left(\prod_{i\in M}P_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)\right)\\&=\arg\max_{r}\sum_{i\in M}lnP_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)\end{aligned} \tag{3}$$

Assuming that the random variable at position ℓ is Y, the position fingerprint ω that can be observed is X, and the observed position fingerprint is a relatively small-scale discrete point. We assume that Y is obtained from a normal distribution and construct a Bayesian Linear Regression model is as follows:

$$y_i=\beta_1x_i+\beta_0+\epsilon_i \tag{4}$$

y is the response variable, β's are the weights (known as the model parameters), *x's* are the values of the predictor variables, and ϵ is an error term representing random sampling noise or the effect of variables not included in the model, $\epsilon\sim N\left(\mu=0,\sigma^2\right)$.

The likelihood estimate of $lnP_{o_i|\mathcal{L}}\left(o_i|\ell_r\right)$ is:

$$\begin{aligned} P\left(Y|X,\beta_1,\beta_0\right) &= \prod_{i=1}^{N} P\left(\beta_1 x_i + \beta_0 + \epsilon_i | x_i, \beta_1, \beta_0\right) \\ &= \prod_{i=1}^{N} P\left(\epsilon_i | x_i, \beta_1, \beta_0\right) = \prod_{i=1}^{N} P\left(\epsilon_i\right) \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{\left(\epsilon_i - 0\right)^2}{2\sigma^2}} \\ &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{\left(y_i - \left(\beta_1 x_i + \beta_0\right)\right)^2}{2\sigma^2}} \\ &= \prod_{i=1}^{N} f_o\left(x_i | \beta_1 x_i + \beta_0, \sigma^2\right) \end{aligned} \tag{5}$$

Where $f_o()$ is a function as:

$$f_o\left(x_i|\beta_1 x_i + \beta_0, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{\left(y_i - \left(\beta_1 x_i + \beta_0\right)\right)^2}{2\sigma^2}} \tag{6}$$

Here, the likelihood equation can be simplified to:

$$P\left(Y|X,\beta_1,\beta_0\right) = \prod_{i=1}^{N} f_o\left(x_i|\beta_1 x_i + \beta_0, \sigma^2\right) \tag{7}$$

The parameters can be estimated from the sampled data, and the maximum likelihood estimate is:

$$\begin{aligned} \hat{L}\left(\beta_1,\beta_0\right) &= \underset{\beta_1,\beta_0}{\arg\max} \prod_{i=1}^{N} f_o\left(x_i|\beta_1 x_i + \beta_0, \sigma^2\right) \\ &\propto \underset{\beta_1,\beta_0}{\arg\max} \sum_{i=1}^{N} lnf_o\left(x_i|\beta_1 x_i + \beta_0, \sigma^2\right) \end{aligned} \tag{8}$$

In a small dataset we might like to express our estimate as a **distribution** of possible values. This is where Bayesian Linear Regression comes in.

4.3 Geolocation Phase

In the geolocation phase, it is very important to be able to form a robust and credible landmark database in the first two phases. The characteristics of dynamic IP addresses are mainly two aspects: (1) the change period is random and irregular, the time when an IP resides in a certain landmark is not fixed; (2) there is a phenomenon of "itinerant", that is, an IP will be repeated at the same landmark position with a certain probability.

In the first two phases, the geolocation of dynamic IP addresses can be corrected for a certain time period t by collecting the time series characteristics of the data in combination with the prior knowledge of the dynamic IP address.

During IP address geolocation, the IP address records that appear within a period of time t are as follows:

(1) It is a new IP address, the location fingerprint is present, it may be the newly assigned IP of the device in the landmark. Then the geographic location of this IP address is the location of the landmark.
(2) It is a newly appearing IP address. If the location fingerprint is also newly observed, it may be a building that is not in the landmark database. In the pre-processing stage and the correction stage, the landmark corresponding to the location fingerprint is added to the trusted landmark database. Continue to observe the subsequent periods of $t+1$ and $t+2$, and if the IP address is not a dynamic IP address, enter the positioning processing of other related categories.
(3) It is a previously located IP address, and the location fingerprint has also appeared, the corresponding landmark is the geographical location of the current IP address.
(4) It is a previously located IP address, and the location fingerprint comes out newly, the dynamic IP address may be obtained by the device in the new landmark. We examine the location fingerprint associated with the IP address in the current time window t, perform the first two phases. Processing and adding the landmark to the landmark database, and the landmark is the geographic location of the current IP address.

5 Experiments

5.1 Data Sets

In order to evaluate our proposed approach, we employ two datasets with information from the Internet Service Provider (ISP), namely, GeoCN2018 and GeoCC2018. These two datasets were collected based on the principle of crowdsourcing. In order to protect privacy and the legitimacy of research, the sensitive information of user has been processed.

As of 2018, mobile apps (which hides the real name of the app) have over 750 million users. Each user is actually associated with several unique devices, generally, which could be a smart-phone or tablet computer. In the study of this paper, we collected one-month HTTP usage data from May 11, 2018 to June 11, 2018. The volume of our data set is approximately 1.4 TB.

Our one-month dataset covers more than 0.23 million (230,374) IP addresses, The user location collected by Android apps includes two types: coarse-grained location and fine-grained location. The coarse-grained granularity gives the city information of the user, and the fine-grained granularity gives the latitude and longitude coordinates.

We divided 230,374 IP addresses into two data sets, GeoCN2018 in China, GeoCC2018 outside China, The GeoCN2018 dataset contains approximately 760G. The GeoCC2018 database is smaller, with 440G data.

In our one-month data, including *time*, *url*, *latitude* and *longitude*, *ac*, *Host*, *User-Agent*, *city*, *device ID*, etc. As shown in Table 1.

Table 1. Comparison datasets between GeoCN2018 and GeoCC2018.

Comparison	GeoCN2018	GeoCC2018
Time range	3 consecutive weeks	3 consecutive weeks
IP addresses numbers	6,500	4,000
Dynamic IP addresses numbers	3,000	2,000
Android devices numbers	18,400	10,030
City numbers	1,500	700
latitude & *longitude* numbers	1,302,300	890,345
Coordinate system type	WGS84/GCJ02/DB09	WGS84
Mobile apps numbers	10	6
AC type	2G/3G/LTE/Wi-Fi	3G/LTE/Wi-Fi
Labelled landmark numbers	806	713

Note: The latitude and longitude coordinates are generated by the app and are not the original GPS coordinates, after the user opens the location sharing, if the application accesses the network through Wi-Fi or a cellular (2G/3G/LTE), the background positioning module periodically feeds back the user's location information to the application server. Therefore, it is necessary to calculate the corresponding real physical position in combination with a specific coordinate system and app, namely, GPS "drift" phenomenon.

We have taken a series of steps to protect the privacy of the users involved in the dataset. First, all raw data collected for this study were kept in an ISP data server. Second, our data collection and analysis pipelines were completely managed by two ISP staff. Finally, the ISP staff have made the user identifiers anonymous. The dataset only includes the statistics for the users covered during our study.

5.2 Performance Criteria

- *Error Distance*: We use the error distance—the distance from the measured location to the actual location—to quantitatively evaluate the accuracy of the geolocation.

$$d_{error} = |d_{measurement} - d_{truth}|$$

In consideration of the actual situation, it is difficult to collect dynamic IP addresses with ground-truth locations. Compared with the existing IP geolocation techniques, a dataset with hundreds of IP addresses as sample is fully sufficient to evaluate their technical differences.

- *Response Time*: For dynamic IP geolocation, we use response time as an indicator of the performance of the algorithm. For instance, the response time of m dynamic IP addresses is $t_1, t_1, ..., t_m$, then the *mean* response time is defined as:

$$t_{RT} \quad = \quad \frac{1}{m}\sum_{i=1}^{m} t_i$$

  Considering the variability of dynamic IP addresses, response time is an important factor affecting the accuracy of the geolocation algorithm. It usually takes hundreds of milliseconds to tens of seconds to locate a single IP address.

5.3 Implementation Details

In this section, we describe the implementation details. The *GeoBLR* algorithm have implemented in experimental tests. Figure 1 illustrates the algorithm procedure, which consists of four parts:

(1) Preprocessing engine. Implements our candidate landmarks selection strategy described in Definition 4. using the coverage area of the largest convex polygon to get a set of landmarks. Compared to the circular coverage area of the largest diameter, it has lower computational complexity.

(2) Calibration engine. The *GeoBLR* algorithm is deployed on the calibration engine. The aim of Bayesian Linear Regression is not to find the single "**best**" value of the model parameters, but rather to determine the posterior distribution for the model parameters. Here we can observe the two primary benefits of Bayesian Linear Regression: (1) If we have domain knowledge, or a guess for what the model parameters should be, we can include them in our model, unlike in the frequentist approach which assumes everything there is to know about the parameters comes from the data. If we don't have any estimates ahead of time, we can use non-informative priors for the parameters such as a normal distribution. (2) The result of performing Bayesian Linear Regression is a distribution of possible model parameters based on the data and the prior. This allows us to quantify our uncertainty about the model: if we have fewer data points, the posterior distribution will be more spread out.

(3) Landmark database. It stores the landmarks we use, including their IP addresses, location and status information. This database is constantly changing, which means that we need to track the status of the landmarks, and maintain the landmark database dynamically, clean up the landmarks as reported many errors, as well as adding new ℓ as landmarks (described in Definition 3).

(4) Geolocation engine. If a dynamic IP address is done with a fine-grained location, Calibration Engine will check the corresponding landmark and update it in Landmark database.

We also implement two state-of-the-art algorithm, namely, *uCheckin* and *GeoQL*. The *uCheckin* is based on the partial implementation of the Checkin-Geo [24] algorithm, which is mainly a complete-linkage hierarchical clustering method [19]. Due to the lack of the login logs from PCs, we can only implement the "checkin" from location-sharing. The *GeoQL* is based on the algorithmic idea in the paper [12], it is an optimization algorithm based on heuristic rules from the location information of the query logs. Corresponding to the coarse-grained location information in our dataset, we can improve them using *GeoQL*.

We compare *GeoBLR* with IP2Location (download the latest database from the webside), IP2Location is based on database-driven techniques and is also very popular IP geolocation database. Primarily, we compare *GeoBLR* with *uCheckin*. In addition, we also compare *GeoBLR* with *GeoQL*.

We split GeoCN2018 dataset into part-overlapping subsets of 1070, 895 and 743 dynamic IP addresses, for preprocessing, calibration and geolocation, respectively. Compared to the GeoCN2018 dataset, the GeoCN2018 dataset are reserved for preprocessing, calibration and geolocation, while each one sets consists of 600 dynamic IP addresses.

On the GeoCN2018 dataset and GeoCC2018 dataset, we evaluated the error distance distribution and response time of *GeoQL*, *uCheckin* and *GeoBLR*, respectively.

5.4 Results

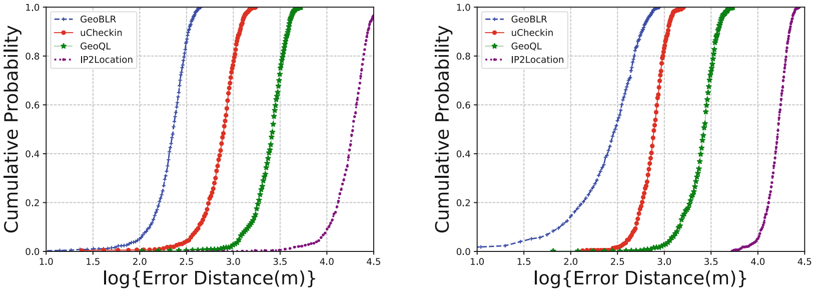
A. Error Distance

Table 2 gives the *mean*, *median*, *max*, *std.* and *mode* error distance of targets in all the four algorithms. From the experimental results in Table 2, it can be concluded that *GeoBLR* and *uCheckin* have higher precision than *GeoQL* and IP2Location, *GeoBLR* has smaller standard deviation (*std.*) than *uCheckin*, and on the denser GeoCN2018 dataset, the stability of *GeoBLR* experimental results is more than that on the sparse GeoCN2018 dataset.

It is distinctly that IP2Location has the worst estimation precision. The *uCheckin* has better precision and the *GeoBLR* has achieved the best precision. We use the metric *median* error to further compare *uCheckin* and *GeoBLR*. Since the *mean* error can be influenced by abnormal or large errors from few IP addresses, the *median* error is widely used in geolocation systems. The *median* errors of IP2Location, *GeoQL*, *uCheckin* and *GeoBLR* are 13,783 m, 2,700 m, 819 m, and 239 m, respectively. This indicates that *GeoBLR* achieves a median estimation error with an order of magnitude smaller than *uCheckin*, *GeoQL* and IP2Location in most cases. Figure 3a demonstrates the cumulative probability of the error distance from each individual testing IP in the GeoCN2018 dataset. Figure 3b shows the cumulative probability of the error distance from each individual IP address in the GeoCC2018 dataset. It can be drawn from the

Table 2. The results of 4 algorithms on datasets GeoCN2018 and GeoCC2018.

GeoCN2018	*GeoBLR*	*uCheckin*	*GeoQL*	*IP2Location*
mean error distance	233.47	807.75	2,689.31	19,611.48
median error distance	232	808	2,690	19,688
max error distance	455	1,750	5,336	37,250
std error distance	80.47	282.71	884.55	6,566.98
mode error distance	230	871	1,789	15,105
GeoCC2018	*GeoBLR*	*uCheckin*	*GeoQL*	*IP2Location*
mean error distance	311.94	801.87	2,701.61	16,828.46
median error distance	297	802	2,726.5	16,788
max error distance	849	1592	5,113	28,716
std error distance	179.92	237.68	893.87	3,843.54
mode error distance	222	859	1,839	18,825

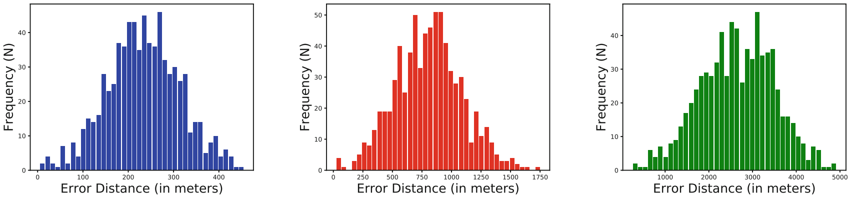


(a) 743 IPs (b) 600 IPs

Fig. 3. The logarithm of the Error Distances for (a) The 743 IPs in GeoCN2018 Dataset and (b) The 600 IPs in GeoCC2018 Dataset.

comparison of the curves in Fig. 3 that the error range of IP2Location is large, for the dynamic IP address, its application is not significant because the granularity is too coarse. Compared with GeoCC2018 dataset, The GeoCN2018 dataset has a higher density and a more concentrated landmark set. The GeoCC2018 has a lower density but a wider distribution. Our approach is sensitive to the distribution and density of landmarks compared to other approaches (e.g. *uCheckin* and *GeoQL*).

On the GeoCN2018 dataset, the error distance histograms of the three comparison algorithms *GeoBLR*, *uCheckin*, and *GeoQL* are shown in Fig. 4. The error distance distributions of the three algorithms are approximately normal distribution. Its parameter values are similar to calculated values. The error distance of the algorithm *GeoBLR* is concentrated in the interval of 50–350 m, the algorithm *uCheckin* is concentrated between 250–1250 m, and the algorithm *GeoQL* is concentrated in the range of 900–4500 m. Experimental data shows that *GeoBLR* implements fine-grained positioning compared to *uCheckin* and *GeoQL*.

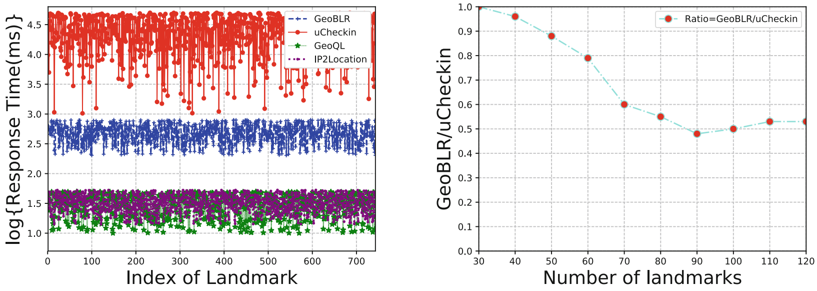


(a) *GeoBLR* (b) *uCheckin* (c) *GeoQL*

Fig. 4. The histogram of in GeoCN2018 of (a) *GeoBLR*, (b) *uCheckin*, (c) *GeoQL*.

B. Response Time

Obviously, *GeoQL* and IP2Location belong to the database-driven techniques whose IP/Location mappings are precomputed and does not need any delay. Consequently, the response time of *GeoQL* and IP2Location is negligible. On the contrary, our experiments show that *GeoBLR* has a *mean* response time of 309 ms. The major computational overhead for *GeoBLR* comes from the calibration phase, where the computational complexity is large. Our experiments show that the dynamic IP address changes from 1 min to 2 days, and in most cases is 120 min, depending on the specific strategy of the ISP. Therefore, although the response time of database-driven geolocation is negligible, but it is hard to maintain and keep up-to-date frequently, which is not applicable to dynamic IP geolocation. In the Fig. 5a, we observe the response time of 743 different dynamic IP addresses on four different algorithms. Considering that *GeoBLR* algorithm does not calculate all location fingerprints, therefore, the computational complexity is lower than *uCheckin*.

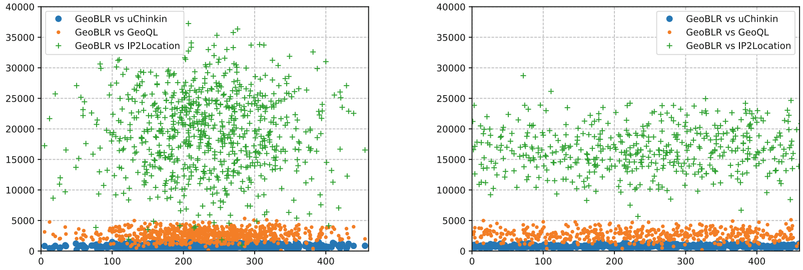


(a) Response Time(ms) (b) Ratio

Fig. 5. (a) The logarithm of response time of *GeoBLR* in GeoCN2018 and (b) The ratio between the median error by *GeoBLR* and the one by *uCheckin*.

C. Comparison

We use *ratio* between the *median* error achieved by the *GeoBLR* method and the *median* error achieved by the *uCheckin* method to quantify the impact factors of landmark density. When the number of landmarks is small, the performance of the two algorithms tend to be similar. On the contrary, as the number of landmarks involved in the IP geolocation increases, the performance of the *GeoBLR* method, with respect to the *uCheckin* method, increases as well. When the landmark density is too large, the performance improvement of the *GeoBLR* method tends to be slow. As shown in Fig. 5b. In fact, Our algorithm relies more on semantic of landmarks than on quantitative of ones.



(a)

(b)

Fig. 6. (a) The scatter in GeoCN2018 dataset and (b) The scatter in GeoCC2018 dataset.

On the two different datasets GeoCN2018 and GeoCC2018, we also compared the scatter plot of error distance between our proposed algorithm and the other three algorithms (*uChechin*, *GeoQL*, IP2Location), as shown in Fig. 6. The algorithms with large error distance distribution such as *GeoQL* and IP2Location are affected by the distribution of landmarks, and the position of dynamic IP addresses are more fluctuating, while the algorithm of smaller error distance is more correlated (*GeoBLR* and *uCheckin*). The reason for our analysis is that most of dynamic IP addresses only have one candidate landmark, the final result of the algorithm *GeoBLR* and *uCheckin* tend to be consistent.

6 Conclusion

In this paper, we propose a dynamic IP geolocation method that is based on Bayesian Linear Regression, introducing a time attribute to describe the dynamic IP address in the location information, and using the largest convex polygon coverage area to select the candidate landmarks. Our experimental results demonstrate that our method achieves state-of-the-art results (1) error distance 50–300 m, and (2) 100–350 ms response time, which can introduce regularization into the estimation process and prevent the risk of over-fitting of data. It also can be easily extended to leverage other types of information. We believe that results achieved in this scenario are more representative of real-world operating conditions.

Acknowledgment. This work was supported by the National Key R&D Program 2016, 2016YFB080 1300/2016YFB0801304.

References

1. Apnic - query the apnic whois database. http://wq.apnic.net/apnic-bin/whois.pl
2. Digital element. http://info.digitalelement.com
3. Google maps with my location. http://www.google.com/mobile/gmm/index.html
4. Hostip.info. http://www.hostip.info/
5. Ip2location.geolocate ip address location using ip2location. https://www.ip2location.com/
6. Maxmind.detect online fraud and locate online visitors. http://www.hostip.info/
7. Neustar. https://www.home.neustar/
8. Skyhook.location technology and intelligence. https://www.skyhookwireless.com/
9. Arif, M.J., Karunasekera, S., Kulkarni, S., Gunatilaka, A., Ristic, B.: Internet host geolocation using maximum likelihood estimation technique. In: 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 422–429. IEEE (2010)
10. Ciavarrini, G., Disperati, F., Lenzini, L., Luconi, V., Vecchio, A.: Geolocation of internet hosts using smartphones and crowdsourcing. In: WMNC, pp. 176–183 (2015)
11. Ciavarrini, G., Luconi, V., Vecchio, A.: Smartphone-based geolocation of internet hosts. Comput. Netw. **116**, 22–32 (2017)
12. Dan, O., Parikh, V., Davison, B.D.: Improving IP geolocation using query logs. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 347–356. ACM (2016)
13. Ding, S., Luo, X., Yin, M., Liu, Y., Liu, F.: An IP geolocation method based on rich-connected sub-networks. In: 2015 17th International Conference on Advanced Communication Technology (ICACT), pp. 176–181. IEEE (2015)
14. Eriksson, B., Barford, P., Sommers, J., Nowak, R.: A learning-based approach for IP geolocation. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 171–180. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12334-4_18
15. Gueye, B., Uhlig, S., Fdida, S.: Investigating the imprecision of IP block-based geolocation. In: Uhlig, S., Papagiannaki, K., Bonaventure, O. (eds.) PAM 2007. LNCS, vol. 4427, pp. 237–240. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71617-4_26
16. Guo, C., Liu, Y., Shen, W., Wang, H.J., Yu, Q., Zhang, Y.: Mining the web and the internet for accurate IP address geolocations. In: IEEE INFOCOM 2009, pp. 2841–2845. IEEE (2009)
17. Hillmann, P., Stiemert, L., Dreo, G., Rose, O.: On the path to high precise IP geolocation: a self-optimizing model. Int. J. Intell. Comput. Res. (IJICR) **7**, 682–693 (2016)
18. Jin, Y., Sharafuddin, E., Zhang, Z.L.: Identifying dynamic IP address blocks serendipitously through background scanning traffic. In: Proceedings of the 2007 ACM CoNEXT Conference, p. 4. ACM (2007)
19. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika **32**(3), 241–254 (1967)
20. Lee, Y., Park, H., Lee, Y.: IP geolocation with a crowd-sourcing broadband performance tool. ACM SIGCOMM Comput. Commun. Rev. **46**(1), 12–20 (2016)

21. Li, D., et al.: IP-geolocation mapping for moderately-connected internet regions. IEEE Trans. Parallel Distrib. Syst. **24**, 381–391 (2012)
22. Li, H., Zhang, P., Wang, Z., Du, F., Kuang, Y., An, Y.: Changing IP geolocation from arbitrary database query towards multi-databases fusion. In: 2017 IEEE Symposium on Computers and Communications (ISCC), pp. 1150–1157. IEEE (2017)
23. Li, M., Luo, X., Shi, W., Chai, L.: City-level IP geolocation based on network topology community detection. In: 2017 International Conference on Information Networking (ICOIN), pp. 578–583. IEEE (2017)
24. Liu, H., Zhang, Y., Zhou, Y., Zhang, D., Fu, X., Ramakrishnan, K.: Mining checkins from location-sharing services for client-independent IP geolocation. In: IEEE INFOCOM, 2014 Proceedings, pp. 619–627. IEEE (2014)
25. Mun, H., Lee, Y.: Building IP geolocation database from online used market articles. In: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 37–41. IEEE (2017)
26. Ng, T.E., Zhang, H.: Predicting internet network distance with coordinates-based approaches. In: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, vol. 1, pp. 170–179. IEEE (2002)
27. Padmanabhan, V.N., Subramanian, L.: An investigation of geographic mapping techniques for internet hosts. In: ACM SIGCOMM Computer Communication Review, vol. 31, pp. 173–185. ACM (2001)
28. Siwpersad, S.S., Gueye, B., Uhlig, S.: Assessing the geographic resolution of exhaustive tabulation for geolocating internet hosts. In: Claypool, M., Uhlig, S. (eds.) PAM 2008. LNCS, vol. 4979, pp. 11–20. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79232-1_2
29. Wang, T., Xu, K., Song, J., Song, M.: An optimization method for the geolocation databases of internet hosts based on machine learning. Math. Probl. Eng. **2015**, 17 (2015)
30. Wang, Y., Burgener, D., Flores, M., Kuzmanovic, A., Huang, C.: Towards street-level client-independent ip geolocation. In: NSDI, vol. 11, p. 27 (2011)
31. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: a comprehensive framework for the geolocalization of internet hosts. In: NSDI, vol. 7, p. 23 (2007)
32. Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., Wobber, T.: How dynamic are IP addresses? In: ACM SIGCOMM Computer Communication Review, vol. 37, pp. 301–312. ACM (2007)
33. Youn, I., Mark, B.L., Richards, D.: Statistical geolocation of internet hosts. In: Proceedings of 18th International Conference on Computer Communications and Networks, ICCCN 2009, pp. 1–6. IEEE (2009)
34. Zhao, F., Luo, X., Gan, Y., Zu, S., Cheng, Q., Liu, F.: IP geolocation based on identification routers and local delay distribution similarity. Concurrency Comput.: Practice Exp. e4722 (2018)
35. Zhao, F., Luo, X., Gan, Y., Zu, S., Liu, F.: IP geolocation base on local delay distribution similarity. In: Wen, S., Wu, W., Castiglione, A. (eds.) CSS 2017. LNCS, vol. 10581, pp. 383–395. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69471-9_28