Mining Checkins from Location-sharing Services for Client-independent IP Geolocation

Hao Liu, Yaoxue Zhang, Yuezhi Zhou, Di Zhang, Xiaoming Fu† and K.K. Ramakrishnan‡
Key Laboratory of Pervasive Computing, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Sci. and Tech., Tsinghua University, Beijing 100084, China
†University of Goettingen, Goettingen, Germany
‡WinLab, Rutgers University, North Brunswick, NJ, USA

***Abstract*—Accurately determining the geographic location of an Internet host is important for location-aware applications such as location-based advertising and network diagnostics. Despite their fast response time, widely used database-driven geolocation approaches provide only inaccurate locations. Delay measurement based approaches improve the estimation accuracy but still suffer from a limited precision (about 10 km) and a long response time (tens of seconds) to localize a single PC, which cannot meet the demand of precise and real-time geolocation for location-aware applications. In this paper, we propose a new geolocation approach, Checkin-Geo, which exploits geolocation resources fundamentally different from existing database-driven (using DNS, Whois, etc.) or network delay measurement based approaches. In particular, we leverage the location data that users are willing to share in location-sharing services and logs of user logins from PCs for real-time and accurate geolocation. Experimental results show that compared to existing geolocation techniques, Checkin-Geo achieves 1) a median estimation error of 799 meters (an order of magnitude smaller than existing approaches), and 2) a negligible response time, which are promising for accurate location-aware applications.**

I. Introduction

Accurately determining the geographic location of an Internet host is the basis for effective location-aware applications [2], [5]. While a coarse-grained geolocation method, possibly with the accuracy at the city-level is sufficient for applications such as website localization, more accurate geolocation (e.g., 1 km [28], [29]) is very desirable for a number of other applications [2], [8]. For example, by using accurate geolocation techniques, online targeted advertising can recommend nearby business (e.g., restaurants, cinemas and fitness/music classes) within a few kilometers of the user's current location. In contrast, existing web advertising typically display advertisements targeted for large geographic range. This is especially the case with end-systems that have wired access, where assistance from tools (e.g., GPS) providing fine-grained location awareness doesn't exist. For such clients with wired access, accurate geolocation can also help a network diagnostics tool to identify exact failure locations, and help a service monitoring tool to report user experience (delay, bandwidth, bugs, etc.) with valuable location information instead of only the user ID and IP address.

Existing client-dependent geolocation techniques achieve a high accuracy based on GPS, or geographic information inferred from Wi-Fi, cell tower, etc. However, these techniques rely on client support (the server does not know where the user is, unless the user explicitly reports his information back to the server), which is not applicable for scenarios such as location-based access restrictions, context-aware security and online targeted advertising [2]. Accurate goelocation is still a challenge for the large number of devices (desktops, laptops) that use wired access [30] which do not have a capability of GPS, cellular, or Wi-Fi. Therefore, accurate client-independent geolocation techniques are required [2].

Client-independent geolocations typically map an IP address to the geographic location of the corresponding host, referred to as *IP geolocation* [1], [8]. Existing work towards IP geolocation can be categorized into database-driven geolocation and delay measurement based geolocation approaches. Database-driven approaches typically maintain a database with records of IP/location mappings whose geolocation information come from the Whois database [18], DNS [7], user contributions [20], etc. Database-driven geolocations [20]–[22] are widely used for its simplicity of deployment and fast response time. However, such IP/location mappings are very coarse-grained and usually at most achieve a city-level precision [1], [19]. Based on the idea that the network delay between two hosts has a positive correlation with the geographic distance between the two hosts, a number of delay measurement based approaches [2], [3], [5], [7], [8] have been proposed. These approaches estimate the location of the target IP address by measuring network delays from geographically distributed servers. Because of these delay measurement based approaches, the geolocation precision is improved to under 10 km. However, these delay measurement based approaches have not been widely deployed in reality due to the following reasons [1]: 1) high deployment cost: a large number of geographically distributed servers are needed for probing the target; and 2) long response time: the response time of localizing a single PC is typically tens of seconds due to large measurement delays, which cannot meet the real-time requirement of many location-aware applications.

Recently, the increasing popularity of location-sharing services (e.g., Foursquare [25] and geolocated tweets in Twitter) offers us an additional opportunity to obtain the precise geographic locations of users who are willing to share their location publicly (typically using *smartphones*). Each location sharing is referred to as a *checkin*. In this paper, we demonstrate the efficacy of combining the check-in information (location information from smartphones) with the logs of the users logging onto the service with their computers (IP information from PCs [1]), to achieve a fast and more precise IP geolocation technique than existing approaches. Essentially, we leverage the accurate location data on smartphones obtained from GPS, cellular, Wi-Fi, etc., to help the geolocation for PCs which

[1]If not otherwise stated, PC in this paper means a computer (e.g., desktop or laptop) without capability of location sensing, in contrast to a smartphone or some other cellular devices.

978-1-4799-3360-0/14/$31.00 © 2014 IEEE

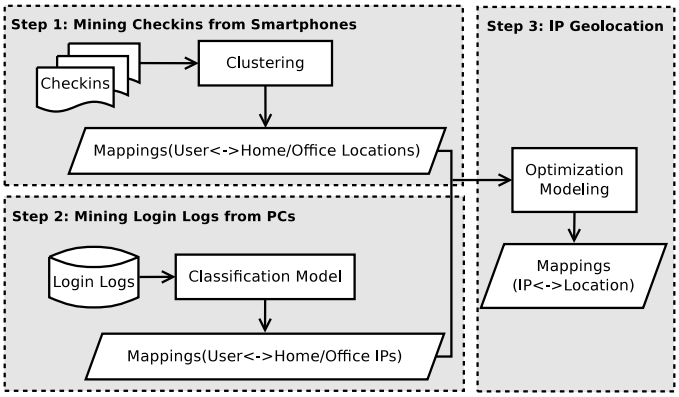


Fig. 1: Overview of Our Approach Checkin-Geo.

do not have a capability of location sensing. First, since users' mobility patterns have some regularity, it is possible to estimate the candidate home and office locations for a user based on this user's checkin history. Second, since a user typically uses his/her PC at home or office, his/her frequently used login IPs from that PC is likely to be at one of the estimated candidate home or office locations. We use an example to further illustrate our idea. We collect Bob's checkin history in Foursquare and estimate his home to be at H_1, H_2 or H_3, and his office to be at O_1,O_2 or O_3. We then collect Bob's login logs in Foursquare, Twitter, Facebook, etc. [2], and find IP_1 and IP_2 are the two IP addresses frequently used by Bob from a PC. In this case, IP_1 and IP_2 are likely to be from one of the locations H_1, H_2, H_3, O_1, O_2 or O_3. We finally use an optimization model to decide the most probable location for IP_1 and IP_2.

Briefly, our approach, referred to as Checkin-Geo, includes the following three steps (Figure 1). First, based on the observation that users are typically near their home at night and near their office during working hours, we cluster the checkins at night and work hours respectively, and estimate the home/office location candidates of each user. After this step, we get the **mappings from each user account to its home/office location candidates**. Secondly, we randomly sample a subset of users' login logs from PCs and manually tag whether a login record (and thus the login IP in this login record) is from the user's home or office. We use this tagged dataset to train a classification model, and then use this model to tag all the login logs automatically. To increase IP coverage, we employ /24 IP segmenting to transform IP records to IP segments. After this step, we get the **mappings from each user account to its home/office IP segments**. Finally, based on the two mappings obtained from the first two steps, we design an optimization model to estimate the **mappings from IPs to locations**. The mappings here can be derived because typically a user uses the same account accross the mobile app, PC client, and Web version of a service (e.g., a user Bob uses the account *Bob1980* to log in Facebook in the browser on his PC, and uses the same account to log in Facebook in his smartphone). Based on the mappings from IPs to locations, we can then return the most probable geographic location for any given IP address.

In summary, the major contributions of this paper include:

(1) We exploit information resources that are fundamentally different from existing approaches for IP geolocation. In particular, based on the observation that people's mobility follows a certain regular pattern, we leverage i) users' checkins shared (typically using smartphones) *publicly* in location-sharing services and ii) their login logs from PCs for IP geolocation. In principle, we are using devices with accurate location capabilities (e.g., smartphones) to help general PCs without a localization assistant tool (e.g., desktops).

(2) We leverage a huge amount of checkin histories and design a simple clustering method based on the users' mobility patterns to estimate users' home and office locations. Experimental results show that the candidates produced by our method cover home/office locations for 98% of the users. We model Checkin-Geo as an optimization problem and solve it through a complete-linkage hierarchical clustering method [24]. The most likely location for an IP segment is estimated to be at the intersection of multiple users' home/office location candidates.

(3) Experimental results show that compared with existing geolocation techniques, Checkin-Geo achieves 1) a median estimation error of 799 meters (an order of magnitude smaller than existing approaches), and 2) a negligible response time (similar to existing database-driven techniques). Checkin-Geo is fast so that it can complement/replace existing widely used database-driven geolocations, while at the same time provides a high precision (even much higher than delay measurement based approaches). Checkin-Geo has been deployed in a commercial online targeted advertising product [36] serving about 800 million users. Based on the ground-truth surveyed with about 5000 users across China, Checkin-Geo achieves a precision within 2.5 km (such a precision can be of significant help in practice for targeted advertising) for 68% of users, which suggests Checkin-Geo can provide fine-grained location-based targeted advertising for millions of users in this service.

II. Related Work

A. Geolocation Techniques

1) Client-dependent Geolocation Techniques:

The Global Positioning System (GPS) has been widely used and it provides precise location information to any device with a GPS receiver. However, GPS is energy expensive for mobile devices (e.g., smartphones) and cannot be used indoors due to weak GPS signals. Skyhook [32], Place Lab [26], and Google My Location [27] scan the location information of cell towers and Wi-Fi access points all around the world (typically through a car), and estimate the location of a client through information broadcast from nearby cell towers or Wi-Fi access points. However, the cost of deploying cars for surveying location information is high and there is a debate on whether scanning these information is legal in some countries for privacy concerns.

Moreover, all client-dependent geolocation techniques suffer from the following two limitations [2]. Firstly, these techniques require clients' support to report their locations to the server, which is not applicable for scenarios such as location-based targeted advertising, context-aware security, location-based access restrictions, and online service analysis. Secondly, there are many devices with only wired access [30] and they do not have a capability of GPS, cellular, or Wi-Fi.

2) Client-independent Geolocation Techniques:

Database-driven Geolocation. Database-driven geolocation techniques try to build a database with huge number

[2]Accounts links among different social services are usually publicly posted on users' profile pages. For example, many Foursquare users post their Twitter and Facebook page links in their profile pages in Foursquare.

Methods		Geolocation Resources	Median Estimation Error (km)	Response Time (sec)	Deployment
Delay Measurement Based Geolocation	GeoPing	Network Delay	382	Tens of Seconds to Several Minutes	Tens of Geographically Dispersed Servers
	CBG	Network Delay	228		
	TBG	Network Delay & Topology	67		
	Octant	Network Delay & Topology	35.4		
	Wang-Geo	Network Delay & Topology & Postal Addresses from Web	7.7		
Database-driven Geolocation	Existing Approaches	Whois, DNS, Postal Addresses from Web, User Contributions	City-level	Negligible	Negligible
		User Registration Records			
	Checkin-Geo	Checkins & Login Logs	0.8		

TABLE I: A Comparison of Checkin-Geo and Related Works in IP Geolocation.

of IP/location mapping records, whose geolocation resources come from the Whois database [18], DNS [7], postal addresses from the Web [4], user contributions [20] and users' registration records [7]. Database-driven geolocations are widely used in commercial systems (e.g., [20]–[22]) for their fast response time and easy deployment. However, the geolocation error is large since the geolocation resources are quite coarse-grained. Typically, database-driven geolocations can only provide a city-level geolocation [1], [19], which cannot meet the demand of precise geolocation for many location-aware applications.

Delay Measurement Based Geolocation. Delay measurement based geolocation approaches estimate the geographic location of a target IP based on the network delays from known landmarks to the target. The rationale beneath these approaches is the positive correlation between network delay and geographic distance. Thus measured network delays can be converted to distances or distance constraints. In these approaches, a number of geographically dispersed servers are deployed to measure the network delays or routes from these servers to the target IP. Depending on the geolocation resources that these approaches leverage, they can be categorized into: (i) estimating the location simply based on network delay measurements, e.g., GeoPing [7] and CBG [5]; (ii) combining network delay and topology measurements, e.g., TBG [8] and Octant [3]; and (iii) combining network delays, network topology and postal addresses from websites, e.g., Wang-Geo [2].

With contributions from these works, the median estimation error of delay measurement based geolocation has been reduced from the original hundreds of kilometers to under 10 km [2], [3], [5], [7], [8]. However, delay measurement based geolocation techniques have never been widely deployed in reality for the following two reasons [1]. First, they need a number of geographically dispersed servers for probing and measuring the target IP. Thus the deployment is difficult. Second, these techniques usually suffer from large measurement overhead, with a response time ranging from tens of seconds to several minutes to localize a single IP [1], [5]. For example, Wang-Geo [2] needs a response time of 25.9 seconds on average (see Section VIII-C).

Checkin-Geo proposed in this paper leverages a huge amount of checkins to mine the mappings of IP/location. Compared with existing client-independent geolocation techniques, Checkin-Geo demonstrates the following advantages: (i) good geolocation precision with an order of magnitude smaller than the state-of-the-art; (ii) fast response time; and (iii) easy deployment. Table I shows the detailed comparison among Checkin-Geo and related works in IP Geolocation.

B. Studying and Mining Location Data

With the increasing popularity of location-aware devices, a considerable amount of research efforts have been attracted to study location data in recent years. Leveraging the movement trajectories sampled at high frequency from volunteers, researchers have tried to predict users' future activities [9], infer people's transportation modes [10], and identify semantic regions associated with users' activities [15]. Based on the observation that phone call records contain both the time and the associated cell tower ID of each call event, Isaacman *et al.* [12] propose an algorithm to identify important locations for users, and Cho *et al.* [11] develop a model to predict the locations and dynamics of future human movements. In particular, Isaacman *et al.* [12] design a logistic regression to identify important locations including home/work for a user. Different from [12] which focuses on the exact single home or work location, we are interested in all the potential home/office location candidates. Experimental results show that the candidates produced by our method based on *checkins* cover home/office locations for 98% of users with an accuracy of 2 km, while [12] estimates the exact home/work with 95th percentile error of 3.86 miles (home) and 21.23 miles (work).

III. Preliminaries

Checkin-Geo combines checkins from location-sharing services and login logs from PCs for IP Geolocation. In this section, we describe these two data structures and the corresponding datasets used in our approach.

A. Checkins

In location-sharing services, a location sharing record from a user is referred to as a *checkin*. Typically, a checkin contains 1) a user account, 2) a timestamp, 3) a location, and 4) a message (optional). Table II shows a checkin example, which shows Bob has shared his location in GPS coordinate (31.213948, 121.239434) at 13:20:20, 2012-02-12. Moreover, we know that Bob was enjoying pizza in a restaurant called Mr. Pizza. From one of the largest location-sharing services in China, we collect a total of 1,398,827 *publicly* shared checkins from 16,134 users in Shanghai between Oct. 19, 2010 and Jan. 4, 2012. Although we are using checkins *publicly* shared by users, we take a further step to protect privacy by hashing each user account to an unique identifier.

B. Login Logs

A login record is the information stored when a user logins to a website. As shown in Table III, each login record typically

Account	Bob
Time	2012-02-12 13:20:20
Location	(31.213948, 121.239434)
Message	The pizza is great in Mr. Pizza!

TABLE II: Checkin Example.

Account	Bob
IP	100.200.3.4
Login Time	2012-01-05 11:20:20
User-agent	Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0)

TABLE III: Login Record Example.

includes: 1) user account, 2) user's IP address, 3) login time, and 4) user-agent. The user-agent string comes from HTTP request header and it can be used to identify device type, operating system and browser information. For example, the user-agent string in Table III indicates that Bob uses Microsoft Internet Explorer 8.0 on 64-bit Windows 7. We get login logs of the 16,134 users from the same location-sharing service as the one used for checkins. Similarly, user accounts are also hashed to unique identifiers to protect privacy.

IV. Problem Statement

Definition 1. (*Checkin Record & Checkin History*). A *checkin record* c_u is a triple (g,t,m) which means user u shares his/her location g (typically using a mobile phone) at time t and broadcasts a message m. A *checkin history* C_u is the set of checkin records of user u, i.e., $C_u = \{c_u\}$.

Definition 2. (*Login Record & Login History*). A *login record* l_u is a triple (p,t,a) which means user u logins to a website (typically using a PC) at time t with IP address p and user-agent a. A login history L_u is the set of login records of user u, i.e., $L_u = \{l_u\}$.

Definition 3. (*IP Set*). We define all the login IPs for a given user set U as $P_U = \{l_u.p | u \in U\}$, and a larger IP range whose geographic locations we could like to estimate as $P'_U, P_U \subseteq P'_U$.

PROBLEM. Given a user set U, and the corresponding checkin histories $\{C_u | u \in U\}$ and login histories $\{L_u | u \in U\}$, estimate the geographic location of each IP in P'_U.

V. Mining Checkins from Location-sharing Services

In this section, based on the observation that a user is likely to be near home at night and near to office during work hours, we cluster checkin histories at night and work hours respectively to estimate the home/office locations for each user (Section V-A). Using users with known home/office locations as testing data, we demonstrate that our clustering method can estimate users' home and office locations accurately (Section V-B). In this case, we can estimate the home/office locations through the clustering method for all the users in our dataset (Section V-C).

It is true that users' checkin patterns can be very complicated and they may check in at many different places [14]. However, in Checkin-Geo, we use the possible home and office locations (where users are likely to use their PCs) to study the mappings from these locations to the IP addresses used on users' PCs. Thus we focus on estimating user's home/office locations in this paper (although the estimation for other places may also be possible). Note that we estimate users' home/office locations by clustering checkins shared at night and work hours respectively, which does not depend on users' explicit checkins at their home/office places.

A. Clustering Checkins based on Timestamps

For each user, we use a hierarchical clustering method [17] to cluster *checkins at night* (shared from 8:00 p.m. to 7:59 a.m. every day) and *checkins during work hours* (from 8:00 a.m. to 6:59 p.m. on weekdays), respectively. In clustering, the distance between two checkins are defined as the geographic distance of the two corresponding checkin locations. After clustering, for each user, we get the estimated home locations, i.e., *home candidates*, and the estimated office locations, i.e., *office candidates*.

Definition 4. (*Home Candidate & Office Candidate*). A home cluster $cluster_{u,h}$ is one of the clusters resulted from the hierarchical clustering on the *checkins at night* of a user u. A *home candidate* is $r_{u,h} = (g,n)$ which means g is the center of all the checkin locations in $cluster_{u,h}$, and n is the number of checkins in $cluster_{u,h}$. The *home candidates* for a user u are $R_{u,h} = \{r_{u,h}\}$. Similarly, an *office candidate* is $r_{u,o} = (g,n)$ and the *office candidates* for a user u are $R_{u,o} = \{r_{u,o}\}$. We refer to $r_{u,h}$ whose checkin number n ranks the i th (descending order) in $R_{u,h}$ as $r_{u,h,i}$. Similarly, $r_{u,o}$ whose checkin number n ranks the i th (descending order) in $R_{u,o}$ is referred to as $r_{u,o,i}$.

We run the hierarchical clustering with single-linkage [17] and distance threshold of 150 m. In this case, we stop clustering if the distance between each two clusters is farther than 150 m; otherwise, the nearest two clusters are aggregated into a new cluster. We have experimented with a range of distance thresholds and find that a distance threshold between 100 m and 500 m works well in practice. Here, we simply use the threshold of 150 m for demonstration. After clustering, we get the home candidates $R_{u,h}$ and the office candidates $R_{u,o}$ for each user u.

B. Efficacy Evaluation

Next, we use users with known home or office locations to evaluate the effectiveness of our clustering method. Based on the observation that a small subset of users have claimed their exact home or office locations in the messages of their checkins, we can infer the ground-truth home/office locations of these users. For example, a user u has shared a checkin at the same location $c_u.g$ for 10 times, and each time u broadcasts a message $c_u.m$ like '*I am at home*'. In this case, we can infer $c_u.g$ is the home location of u. We explore checkin histories of all the 16,134 users by filtering their checkin messages with keywords such as '*Home*', '*At Home*', and '*My Home*'. As a result, we get a total of 753 users which satisfy the following two requirements: 1) the user has shared his/her home location for more than 10 times; and 2) all the home locations the user has shared are within a distance of 100 meters. In this way, we get 753 users with known home locations. Similarly, we get 417 users with known office locations.

We use the 753 users with known home locations and 417 users with known office locations to evaluate the effectiveness of our home/office estimation method through clustering. Experimental results show that for 98.3% of users, at least one of the first three home candidates $r_{u,h,j}, j = 1,2,3$ are within 2 km of the user's actual home location. For 99.3% of users, at

least one of the first three office candidates $r_{u,o,j}, j = 1, 2, 3$ are within 2 km of the user's actual office location.

C. Home/Office Estimation for All Users

Therefore, the results from our testing dataset demonstrate that for 98% of users, our clustering method can estimate their home/office locations (using three candidates) with an accuracy of 2 km. In this case, for each user u of the 16,134 users in our whole dataset, we can infer the corresponding home location candidates $R_{u,h}$ and office location candidates $R_{u,o}$, by clustering *checkins at night* and *checkins during work hours*, respectively. We thus get the mappings from each user to the corresponding home/office location candidates, i.e., $u \leftrightarrows R_{u,h}$ and $u \leftrightarrows R_{u,o}$.

We have investigated the users for which we produce large estimation errors. We find that most of them have non-standard schedules such as night shifts, and thus home is usually estimated to be office and office is estimated to be home. We refer to this problem as *inverse estimation*, and we will show that Checkin-Geo can tolerate *inverse estimation* in Section VII-A.

VI. Mining Login Logs from PCs

In this section, we first analyze users' login logs and use a classification model to tag whether an IP address is from the corresponding user's home or office (Section VI-A). Thus we get the mappings from each user to the corresponding home and office IPs. We then employ a /24 segmenting method to increase the IP coverage, and get the mappings from each user to the corresponding home and office IP segments (Section VI-B). Although here we primarily identify home and office IPs for IP geolocation, the technique in this section itself can also be used to customize search results and to target advertisement based on a user's context (home, office, travel, etc) associated with an IP [13].

A. Determining the Scenario Source of an IP

In this subsection, we describe how to determine whether a login IP is from the corresponding user's home or office. Given the login history L_u of a user u, it is easy for a human to determine whether each IP $l_u.p$ in L_u is from home or office. For example, L_u contains two different IPs, i.e., p_1 and p_2. Login records with p_1 are usually from work hours, while login records with p_2 are usually created at night or on weekends. In this case, we can infer that p_1 is the user's office IP address and p_2 is the user's home IP address. The following factors are important when we determine the source (home/office) of an IP.

1) *PC Login Proportion* (the percentage of login records from PCs). If all the login records of an IP for a user are from mobile devices such as smartphones and tablets, this IP is likely to be a temporal IP allocated from mobile ISPs and we can simply omit this IP. Note that the device type can be inferred from the user-agent field l_a as described in Section III-B. The example user-agent in Table III shows the OS is Windows 7 and thus we know it is from a PC instead of a mobile device.

2) *Login Times on Important Holidays* (the times that an IP has been used on important holidays). Only a very small portion of people will work on important holidays, e.g., the Christmas in western countries and the Spring Festival in China. The login logs used in this paper are from users in Shanghai, China, and thus an IP is probably from home if it has been used on the Spring Festival (a 7-day holiday).

3) *Login Days*. If an IP has only been used in few days (e.g., two days), we do not have enough login records to determine its source, and we had better give up this IP.

4) *Login Times during Work Hours*. If an IP has been usually used during work hours, it is likely to be from office. We use the same definition for *Work Hours* as Section V-A, i.e., 8:00 a.m. to 6:59 p.m. on weekdays.

5) *Login Times during Rest Hours*. We define the time not in *Work Hours* as *Rest Hours*, i.e., 7:00 p.m. to 7:59 a.m. on weekdays and the whole weekends. If an IP has been used frequently at *Rest Hours*, it is likely to be from home.

We have a total of 92,153 login records from 16,134 users with 31,634 different IPs. To manually determine the scenario source of each of these IPs is difficult. Moreover, if we explore the login logs of users throughout China or the whole world instead of only users from Shanghai, the task is impossible. Therefore, we employ an automatic method with the following three steps. Firstly, we manually tag the scenario sources of a small portion of these IP login records. Secondly, we use this small tagged dataset to train a classification model. Finally, we use the classification model to automatically determine the scenario source of each IP in the whole login logs.

We randomly sample 1,899 login records of 266 users and manually tag the scenario sources of these IP records. We define three sources: *home*, *office*, and *others*. We tag an IP as *others* if all the records of this IP are from mobile devices or there are not enough login days to determine the scenario source. We use the 5 factors described above as classification features, and employ the classic classifier Support Vector Machine (SVM) [23] with Radial Basis Function (RBF) kernel to classify these IPs into three categories: *home*, *office* and *others*. We divide the 1,899 login records into 5 subsets of nearly equal size. Each time, we use 4 subsets for training and 1 subset for testing. Through such a five-fold validation, we achieve a prediction accuracy of 94.9%. We then use the trained SVM model to classify all IPs in the login logs.

B. Increasing IP Coverage

After the SVM based classification, we get the mappings from each user u to the corresponding home/office IPs, i.e., $u \leftrightarrows P_{u,h}$ and $u \leftrightarrows P_{u,o}$ in which $P_{u,h}$ and $P_{u,o}$ denote the home and office IPs of user u, respectively. However, we only have a total of 31,634 IPs in the login logs. The 31,634 IPs are $P_U = \{l_{u.p} | u \in U\}$ as defined in Definition 3, which is quite small compared with the millions of IPs (i.e., P'_U) in Shanghai that we could like to localize. Therefore, we need to enlarge the IP coverage from P_U to P'_U.

We use IP segmenting to enlarge P_U to P'_U based on the observation that IPs within a /24 segment are usually at the same location. According to [6], 97% of /24 IP segments are at nearly the same location. Moreover, the other 3% of /24 IP segments that disperse for a long distance are usually from backbones of ISPs. Note that we are interested in user-end IPs instead of backbone IPs, given the demand of precise geolocation applications such as location-based advertising. It is true that for residential users, their IP addresses are sometimes dynamically assigned by ISPs. In this case, the location of an IP address or a /24 IP segment can vary across an area depending the structure of IP addresses for ISPs. Fortunately, the high estimation precision in our large-scale

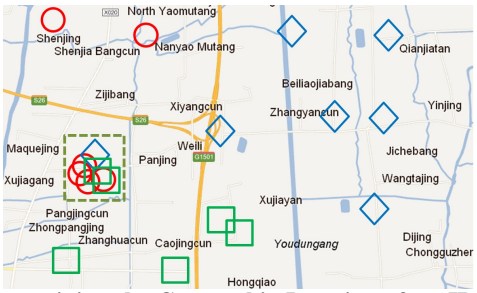


Fig. 2: Determining the Geographic Location of an IP Segment. The location of an IP segment is estimated to be at the intersection area of the home/office candidates of users in this IP segment.

deployment (Section IX) indicates that ISPs tend to assign their IP addresses in a structured way and an /24 segment tends to be assigned to a small geographic area (seems to due to the fact that it is easy for ISPs to maintain and manage their IP addresses with a structured and hierarchical IP assignment).

Our sampled test IPs (see Section VIII-A) show that after the segmenting, Checkin-Geo covers 80.7% of IPs. In this case, we can use Checkin-Geo to provide fast and fine-grained IP geolocations for a large proportion of IPs and reuse existing geolocation techniques for the left IPs. Note that after increasing IP coverage through IP segmenting, Checkin-Geo can estimate the locations of IPs in a large range instead of only home/office IPs or IPs used by users in our data set.

VII. IP Geolocation

From the previous two sections, we get mappings from users to their home and office locations, i.e., $u \leftrightarrows R_{u,h}$ and $u \leftrightarrows R_{u,o}$, and mappings from users to their home and office IPs, i.e., $u \leftrightarrows P_{u,h}$ and $u \leftrightarrows P_{u,o}$. In this section, we build the mappings from IPs to locations to estimate the location of any given IP.

A. Estimation Rationale

Definition 5. (*Users and Home/Office Candidates in a Segment*). The users in an IP segment S denote the users that have used an IP in the range of S, i.e., $U_S = \{u | l_u.p \in S\}$. The home/office candidates for a segment S are $R_S = \cup(R_{u,h} \cup R_{u,o}), u \in U_S$.

We estimate the location of a given IP in the following two conditions: 1) *single user case*: if the /24 segment S that this IP belongs to has only been used by a single user u, i.e., $|U_S| = 1$, then we estimate the location of this IP to be at $r_{u,h,1}$ or $r_{u,o,1}$ depending on whether this IP is tagged to be from *home* or *office*; 2) *multiple users case*: if multiple users have used an IP in the range of S, i.e., $|U_S| > 1$, we estimate the location of this IP considering all home/office location candidates R_S.

Figure 2 shows the idea of estimating the location of an IP p and the corresponding /24 IP segment S in the *multiple users case*. Assuming there are three users u_1, u_2 and u_3 who have used an IP in S, we use the square, diamond and circle to indicate the home/office candidates of the three users respectively. As shown in Figure 2, the home/office candidates of the three users are dispersed. However, each of the three users has at least one home/office candidate in the dashed rectangle region. In this case, there is a great chance that S is located at that region. Note that we consider all home/office candidates R_S instead of only home candidates $R_{u,h}$ or office candidates $R_{u,o}$ according to the tagged source (home or office) of an IP, so that Checkin-Geo can tolerate errors of *inverse estimation* as mentioned in Section V.

An IP segment S is probably at a region x if many users in U_S have home/office candidates in x. However, the reliability of different home/office candidates is different. In this case, the number of checkins in x is also important when we consider the chance that S is at a region x.

B. Optimization Modeling

Following the analysis in Section VII-A, we formulate the problem with the objectives of *maximizing users* and *maximizing checkins*, respectively. We will show in Section VIII-B that *maximizing checkins* achieves a better geolocation precision.

1) *Maximizing Users (MU).* Given the users U_S in the segment S, maximize the number of users who have a home/office location candidate in R'_S, which satisfies that R'_S is a subset of R_S, and each home/office location candidate r_1 in R'_S is within a predefined distance D_0 of any another candidate r_2 in R'_S. Formally,

$$\begin{cases} \max |\{u | r_{u,h} \in R'_S \ or \ r_{u,o} \in R'_S\}| \\ s.t. \\ dist(r_1.g, r_2.g) \leq D_0, \forall r_1, r_2 \in R'_S, R'_S \subseteq R_S \end{cases}$$

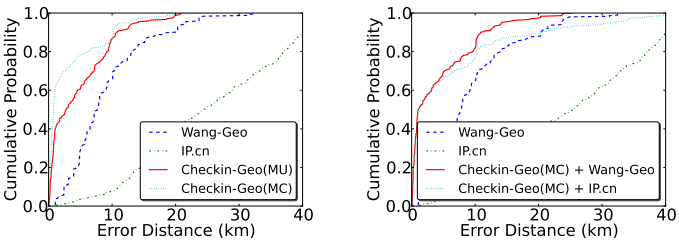
2) *Maximizing Checkins (MC).* Given the users U_S in the segment S, maximize the number of checkins in R'_S, which satisfies that R'_S is a subset of R_S, and each home/office candidate r_1 in R'_S is within a predefined distance D_0 of any another candidate r_2 in R'_S. Formally,

$$\begin{cases} \max \sum_{r \in R'_S} r.n \\ s.t. \\ dist(r_1.g, r_2.g) \leq D_0, \forall r_1, r_2 \in R'_S, R'_S \subseteq R_S \end{cases}$$

A complete-linkage hierarchical clustering method [24] can solve both the two objectives. For a given segment S, we cluster all the home/office candidates in R_S with a distance threshold of D_0, and get a cluster set $\{R'_S\}$. In this case, home/office candidates in the same R'_S are within a distance of D_0, and home/office candidates from different R'_S have a distance farther than D_0. We select R'_S with the most users from $\{R'_S\}$ for the objective of maximizing users, and select R'_S with the most checkins from $\{R'_S\}$ for the objective of maximizing checkins. The location of IP segment S is then estimated to be at the center of the selected R'_S, which is defined as the average latitude and longitude coordinates of all home/office candidates in R'_S, i.e., $Avg\{r.g | r \in R'_S\}$.

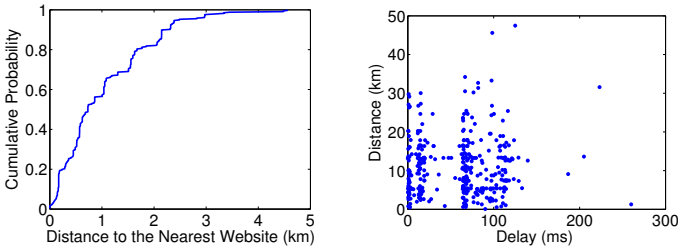
VIII. Evaluation

Database-driven geolocations are widely used in location-based applications, while delay measurement based geolocations are not used in real applications due to their high deployment cost and long response time [1]. We propose Checkin-Geo to be a feasible commercial IP geolocation technique with good scalability, which can replace/complement existing database-driven geolocations and provide fine-grained and fast geolocations for real applications. In this section, primarily we compare Checkin-Geo with IP.cn [35], which is based on database-driven techniques and is also the most popular IP geolocation service in China. We also compare Checkin-Geo with delay measurement based geolocations to 1) show Checkin-Geo achieves even better estimation precision than



(a) 243 IPs. (b) 301 IPs.

Fig. 3: Error Distances for (a) the 243 IPs that Checkin-Geo Covers and (b) All the 301 IPs in Our Test Dataset.



(a) Nearest Website. (b) Delay vs. Distance.

Fig. 4: The Relationship between Network Delay and Geographic Distance.

delay measurement based techniques; and 2) confirm that delay measurement based techniques suffer from a long response time and it is difficult to be used at large scale in commercial applications. As shown in Table I, Wang-Geo [2] achieves the best estimation precision among existing delay measurement based techniques. Thus we compare Checkin-Geo with Wang-Geo. For Wang-Geo, we use all the 17 available servers in PlanetLab [33] in China as probing servers.

A. Datasets

We use IPs with known ground-truth locations to compare Checkin-Geo with Wang-Geo. In order to collect the ground-truth dataset, we setup a web page with a map to enable users to mark their locations. This web page records the latitude and longitude coordinates of the marked point, as well as the IP address that the user uses. We then request volunteers widely dispersed in Shanghai to access our web page and contribute their IPs and locations. We filter out IPs from mobile devices based on the user-agent field of HTTP request, and get a total of 301 IPs with ground-truth locations widely dispersed in Shanghai. Note that it is difficult to collect IPs with ground-truth locations, and a dataset with hundreds of IPs is sufficient compared with the evaluations in existing IP geolocation techniques. For example, Wang-Geo [2] uses a dataset with 160 IPs.

Checkin-Geo covers 243 of the 301 IPs (80.7%) after the step of increasing IP coverage (Section VI-B). We first compare Checkin-Geo with IP.cn and Wang-Geo using the 243 IPs that Checkin-Geo covers. We then use the whole 301 IPs as the test data, to evaluate the scenario where Checkin-Geo is combined with IP.cn (use IP.cn for IPs which Checkin-Geo does not cover) to provide a fast and precise IP geolocation, which we expect to be the model for Checkin-Geo to be used for general location-based applications. We also combine Checkin-Geo with Wang-Geo to see the precision as a reference, although delay measurement based geolocations such as Wang-Geo are difficult to be used at scale in reality due to the long response time and high deployment cost [1].

B. IP Geolocation Error

We use the error distance, i.e., the distance from the estimated location to the actual location, to quantify the geolocation error. Figure 3a demonstrates the cumulative probability of the error distance for the 243 IPs that Checkin-Geo covers. We use legends Checkin-Geo (MU) and Checkin-Geo (MC) to represent Checkin-Geo with the objectives MU and MC, respectively. It is not surprising that IP.cn has the worst estimation precision. Wang-Geo has better precision and Checkin-Geo achieves even better precision.We use the metric *median error* to further compare Checkin-Geo and Wang-Geo. Since the average error can be influenced by abnormal or large errors from few IPs, the median error is widely used in geolocation systems [2], [3], [5], [8] to indicate the average case for estimation error. The median errors of IP.cn, Wang-Geo, Checkin-Geo (MU) and Checkin-Geo (MC) are 24,783 m, 7,735 m, 2,829 m, and 799 m, respectively, which indicates that for an average case, Checkin-Geo (MC) achieves an estimation error with an order of magnitude smaller than Wang-Geo and IP.cn.

We have investigated the causes for the performance difference between Checkin-Geo (MC) and Checkin-Geo (MU). Let us consider a case in which several users in the same IP segment S have an intersection area with small number of checkins, but this area is not the home/office location of anyone of these users. In Checkin-Geo (MU), these intersection areas are likely to be estimated to be the location of S. Therefore, Checkin-Geo (MC) is more applicable in real applications.

We next consider the scenario where Checkin-Geo is expected to be used in reality: we reuse existing database-driven geolocations for IPs which Checkin-Geo does not cover. Therefore, Checkin-Geo can complement existing widely used database-driven techniques by improving the precision significantly without increasing response time or deployment cost. However, to make the evaluation more complete, we also consider the case where we reuse Wang-Geo for the remaining IPs that Checkin-Geo does not cover, which is expected to provide the highest precision, but might not be good for real applications as it takes a long response time for some IPs and the deployment cost is also high. Figure 3b shows the results for these cases for all the 301 IPs in our dataset. The median errors of IP.cn, Wang-Geo, Checkin-Geo(MC) + IP.cn, and Checkin-Geo(MC) + Wang-Geo are 25,732 m, 7,786 m, 1,079 m and 1,001 m, respectively. Therefore, if we combine Checkin-Geo with IP.cn, we achieve a median estimation precision of about 1 km for all IPs, which is promising for a number of location-based applications such as targeted advertising.

Note that Wang-Geo is reported to achieve a median error of 690 to 2,250 m in [2]. However, Wang-Geo shows a median error of 7,735 m in our experiments. Although the main barrier stopping Wang-Geo to be used at scale is its *response time* and *deployment cost*, we still would like to discuss the precision of Wang-Geo in detail. The basic idea of Wang-Geo is 1) collecting a number of websites with known locations (through the postal addresses posted in some web pages) and 2) estimating the target IP to be at the location of the website that has the shortest delay to the target IP. However, we find that the shortest network delay does not simply mean the shortest distance. Figure 4a demonstrates the cumulative probability of the geographic distance from each testing IP to its nearest website in Wang-Geo. It shows that 50% of IPs have a website within 1 km and all IPs have a website within 5 km. If

Wang-Geo can identify the nearest website through the metric *shortest network delay*, Wang-Geo can achieve a median error of less than 1 km and a worse-case error of less than 5 km, instead of the estimation errors shown in Figure 3. We further use a server in PlanetLab to measure the network delays from this server to these websites. Compared with Wang-Geo in which only 13 network delays and corresponding distances are analyzed, here we conduct a more extensive experiment with about 400 delay and distance pairs. Figure 4b demonstrates the network delays from this server to these websites and corresponding geographic distances. It shows that in a city-level range (tens of kilometers), the relationship between the shortest network delay and the shortest geographic distance is quite weak. Reasons responsible for this weak relation include complicated network topology, queueing delay, indirect or circuitous routing, etc. For example, the location of an IP address IP_1 is near to that of an IP address IP_2, but IP_1 and IP_2 belong to different ISPs. In this case, the network delay from IP_1 to IP_2 is usually large. Another possible reason is that more and more websites are deployed in cloud services such as Amazon, in which case the postal address posted in a website is the actual address of a company while the IP address of this website belongs to Amazon.

C. Response Time

For delay measurement based geolocation, the slow response time is a fundamental limitation. It usually takes tens of seconds to several minutes to localize a single IP address [1]. However, Checkin-Geo and IP.cn belong to the database-driven category whose IP/Location mappings are precomputed and does not need any delay measurement. Therefore, the response time of Checkin-Geo and IP.cn is negligible. On the contrary, our experiments show that Wang-Geo has a median response time of 25.9 seconds.

The major time overhead for Wang-Geo comes from the large-scale *traceroute* measurements (typically tens of seconds to several minutes [1]). *traceroute* needs to probe each router hop in the path to the destination in turn, starting from the first one to the last. The probing time can be reduced if we probe each hop simultaneously. However, many routers limit probing packets from a single host, and a large number of simultaneous probes can create a storm of packages (especially in the reply direction) and lead to loss of some replies [34]. In this case, it is difficult to conduct large-scale measurements offline and store the IP/Location mappings for future queries (probing all IP ranges is almost an impossible task). On the contrary, database-drive geolocations such as Checkin-Geo and IP.cn only involve with intensive computation which can be easily speed up by parallelization, thus conducted offline and updated regularly. Due to the long response time, delay-measurement based geolocations are only studied in literature. Instead, database-driven geolocations are widely used commercially as their response time is negligible (although they have much larger estimation errors).

IX. Commercial Deployment

We reimplemented Checkin-Geo based on checkins and login logs from Tencent [31], a large Internet company in China with about 800 million users. We conducted an online survey to request users' postal addresses, to which we mailed the volunteer a gift and thus we have high confidence on these ground-truth locations. We were able to obtain the postal addresses for about 5000 users across China and thus the addresses of the corresponding IP addresses. Evaluation results show that Checkin-Geo achieves an estimation precision within 2.5 km (such precision can be of significant help in practice for location-based applications such as targeted advertising) for 68% of these users. Here, we respect the request from Tencent and do not disclose the detailed distribution of estimation precision. Therefore, we can expect Checkin-Geo to provide fine-grained geolocation for millions of users. We recalculate and update the location of each IP /24 segment every day (we use a cluster of 12 servers each with 8 cores which can finish all the computations involved with Checkin-Geo within one day) based on the data of the last 30 days, in which case Checkin-Geo can keep the computed locations up to date according to the changes of users' mobility (e.g., move to a different home/work place) or the reassignment of IP addresses.

We integrated Checkin-Geo into the IP geolocation service in Tencent, which had used information from Whois, DNS and user contributions for IP geolocation before. We provided this service to a location-based targeted advertising service in Tencent [36]. Before the integration, the advertisements displayed in web pages were based on users' browsing histories and a city-level location information. After the integration, advertisements are also selected based on users' precise locations. Say, based on a user Bob's browsing histories, Tencent estimated his interest and displayed advertisements about fitness classes, music lessons, and cinema tickets in his city to him. With the precise location information from Checkin-Geo, Tencent can show these advertisements of interest nearby his location (neighboring business), which improves the conversion rate and result in more actual purchases.

X. Discussion

Limitations. Compared with existing approaches, Checkin-Geo achieves both high estimation precision and fast response time, and thus it is ready to be deployed at large scale for precise location-based applications, by replacing/complementing existing widely used database-driven geolocation techniques. However, the IP coverage of Checkin-Geo depends on the data we use and typically cannot cover all IPs. In this case, we can combine Checkin-Geo and existing database-based geolocation techniques to provide a city-level precision for IPs that Checkin-Geo cannot cover, as illustrated in Figure 3b.

Data Collection. Checkin-Geo needs to mine checkins and PC login logs from the same users. Collecting these data is not difficult given that most social network services (e.g., Twitter, Facebook, and Foursquare) provide both a mobile (app or web) and a PC version (typically through web) service. Moreover, most location-sharing services allow users to share checkins to other social networks. For example, checkins in Foursquare can be shared to Facebook and Twitter. In this case, both checkins and login logs can be collected from different services and associated through the same users. For example, Facebook can use its own login logs and checkins cross-posted from Twitter, Foursquare or Facebook itself. Note that account links among different services for a user are usually publicly posted on the user's profile page in social network services (See Footnote 2).

Alternatives for IP/Location Mapping. We map IPs to locations through the transition of home/office locations. In fact, we have tried other mapping alternatives. However, initial experiments show that these mapping alternatives are not promising. For example, we have tried to use checkins with a time close to a PC login time for estimating the corresponding

login IP's location. However, we find that only quite a small number of checkins of a user are shared in a time near to this user's PC login time (e.g., within half an hour), and thus the estimation accuracy is quite limited.

XI. Conclusion

In this paper, we propose Checkin-Geo, an accurate IP geolocation approach which exploits checkins from location-sharing services and login logs from PCs, fundamentally different from existing approaches. Both our experimental results and commercial deployment show that Checkin-Geo achieves fine-grained geolocations. Although the precision of Checkin-Geo is impacted by 1) the density of checkins and 2) the mechanisms ISPs use to allocate IPs, our large-scale deployment in Tencent including almost all cities in China where the two factors change in a large range shows promising estimation precision. It should also be noted that although Checkin-Geo initially leverages home/office IPs, it covers many more IP addresses in addition to the *intuitive* home/office IPs after IP segmenting (the IP range of an airport is the office IPs of employees in this airport, the IP range of an cafe can be used by nearby residents or employees as they are using the same IP /24 segment, etc.). Checkin-Geo demonstrates good scalability as it is only a computation intensive approach so that we can speed up the tasks by parallelization and update the IP/Location mappings at high frequency (every day in our deployment as described in Section IX), which makes Checkin-Geo a commercial-ready technique to complement existing database-driven techniques (de facto industry standard). On the contrary, delay-measurement based approaches are difficult to be used commercially as they 1) require to deploy widely dispersed servers and 2) cannot be parallelized at large scale so that the measurements cannot be done offline as the computation in Checkin-Geo (see Section VIII-C).

XII. Ackowledgement

This work is supported by International Science and Technology Cooperation Program of China under Grant No. 2013DFB10070 and EU FP7 IRSES MobileCloud Project (Grant No. 612212). We would like to thank the anonymous reviewers for their helpful comments on this paper.

References

[1] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev*. vol.41, no.2, pp.53-56, 2011.

[2] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. Towards Street-level Client-independent IP Geolocation. In *NSDI*, 2011.

[3] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *NSDI*, 2007.

[4] C. Guo, Y. Liu, W. Shen, H. J. Wang, Q. Yu, and Y. Zhang. Mining the Web and the Internet for Accurate IP Address Geolocations. In *INFOCOM*, 2009.

[5] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking*, vol.14, no.6, pp.1219-1232, 2006.

[6] M. J. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic Locality of IP prefixes. In *IMC*, 2005.

[7] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *SIGCOMM*, 2001.

[8] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *IMC*, 2006.

[9] D. Choujaa and N. Dulay. Predicting Human Behaviour from Selected Mobile Phone Data Points. In *UbiComp*, 2010.

[10] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma. Understanding Mobility Based on GPS Data. In *UbiComp*, 2008.

[11] E. Cho, S. A. Myers, and J. Leskovec. Friendship and Mobility: User Movement in Location-based Social Networks. In *KDD*, 2011.

[12] S. Isaacman, R. Becker, R. Caceres, S. Kobourov, M. Martonosi, J. Rowland, and A. Varshavsky. Identifying Important Places in People's Lives from Cellular Network Data. In *Pervasive*, 2011.

[13] P. Andreas, Y. Xie, F. Yu, M. Abadi, G. M. Voelker, and S. Savage. How to Tell an Airport from a Home: Techniques and Applications. In *HotNets*, 2010.

[14] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An Empirical Study of Geographic User Activity Patterns in Foursquare. In *ICWSM*, 2011.

[15] C. Lu, P. Lei, W. Peng, and I. Su. A Framework of Mining Semantic Regions from Trajectories. In *DASFAA*, 2011.

[16] R. M. Needham. Denial of Service. In *CCS*, 1993.

[17] S. C. Johnson. Hierarchical Clustering Schemes. *PSYCHOMETRIKA*, vol.32, no.3, pp.241-254, 1967.

[18] APNIC - Query the APNIC Whois Database. http://wq.apnic.net/apnic-bin/whois.pl.

[19] GeoIP City Accuracy for Selected Countries. http://www.maxmind.com/en/city_accuracy.

[20] Host IP. My IP Address Lookup and Geotargeting Community Geotarget IP Project. http://www.hostip.info.

[21] IP2Location. IP Address Geolocation to Identify Website Visitor's Geographical Location. http://www.ip2location.com.

[22] MaxMind. Geolocation and Online Fraud Prevention from MaxMind. http://www.maxmind.com/.

[23] C. C. Chang and C. J. Lin. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, vol.2, no.3, pp.1-27, 2011.

[24] D. Defays. An Efficient Algorithm for a Complete Link Method. *The Computer Journal*, vol.20, no.4, pp.364-366, 1977.

[25] Foursquare. https://foursquare.com/.

[26] S. S. Intille, K. Larson, E. Tapia, *et al*. Using a Live-in Laboratory for Ubiquitous Computing Research. In *Pervasive*, 2006.

[27] Google Maps with My Location. http://www.google.com/mobile/gmm/mylocation/index.html.

[28] Location-based Advertising Network Developed. http://www.channelnewsasia.com/stories/technologyfeatures/view/1136532/1/.html.

[29] WiseTouch India's 1st Android based Interactive Media Platform. http://www.wisetouch.in/wp-content/uploads/2012/03/WiseTouch-Press-Release.pdf.

[30] Desktops Still Fit in Post-PC Era. http://www.lenovo.com/articles/us/en/news/desktops-still-fit-in-post-pc-era.html.

[31] Tencent. http://www.tencent.com/en-us/at/abouttencent.shtml.

[32] Skyhook. http://www.skyhookwireless.com/.

[33] PlanetLab. http://www.planet-lab.org/.

[34] Traceroute. http://man.he.net/man8/traceroute.

[35] IP.cn. http://www.ip.cn/.

[36] Tencent Online Advertising Service. http://www.tencent.com/en-us/ps/adservice.shtml.