A Location Spoofing Detection Method for Social Networks

Chaoping Ding[1], Ting Wu[2], Tong Qiao[2], Ning Zheng[1,2], Ming Xu(✉)[1,2], Yiming Wu[2], and Wenjing Xia[1]

[1] Internet and Network Security Laboratory, School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou,China
{161050056,mxu,nzheng,161050051}@hdu.edu.cn
[2] School of Cyberspace, Hangzhou Dianzi University, HangZhou, China
{wuting,tong.qiao,ymwu}@hdu.edu.cn

Abstract. It is well known that check-in data from location-based social networks (LBSN) can be used to predict human movement. However, there are large discrepancies between check-in data and actual user mobility, because users can easily spoof their location in LBSN. The act of location spoofing refers to intentionally making false location, leading to a negative impact both on the credibility of location-based social networks and the reliability of spatial-temporal data. In this paper, a location spoofing detection method in social networks is proposed. First, Latent Dirichlet Allocation(LDA) model is used to learn the topics of users by mining user-generated microblog information, based on this a similarity matrix associated with the venue is calculated. And the venue visiting probability is computed based on user historical check-in data by using Bayes model. Then, the similarity value and visiting probability is combined to quantize the probability of location spoofing. Experiments on a large scale and real-world LBSN dataset collected from Weibo show that the proposed approach can effectively detect certain types of location spoofing.

Keywords: Location spoofing, Social networks, Semantic analysis

1 Introdcution

Several location-based social networks have increased exponentially during the last two decades. The growing popularity of LBSN has brought about the appearance of an economy where users announce. To attract more users, the LBSN provide both virtual or real-word rewards to the user, when a user has certain number of check-in at a venue. Unfortunately, this reward gives users incentives to spoof their location information so that they can check into POI(point of interest) far away from where they really are.

In recent years, the diffusion of location spoofing has stirred debate about the credibility of spatial-temporal data. It can use as an effective countermeasure to protect user privacy, and also has a negative impact on the credibility of LBSN.

However, these two views cannot have a comprehensive understanding of location spoofing. [1,2] indicated that users make more friends or impress others by claiming a false location in social media. [3] thought the business model of Foursquare make it as a lucrative target to attack by location cheating, a business man may use location spoofing to check into a competing business, and badmouth that business by leaving bad comments. To prevent location spoofing in social networks. Foursquare has adopted the "cheater code" to prevent location spoofing attacks.[4,5] proposed Wi-Fi, QR-code and near field based location verifications solutions to solve location fraud problems. Although above-mentioned measures have been applied, once the user knows the verification mechanism, they can launch location cheating attack.

To further validate and understand the discrepancies between user-generated check-ins and their actual movements. [6,7] quantified the value of geosocial datasets by comparing GPS traces against Foursquare check-ins, they find that a large portion of visited locations are forged. A few researchers have been studying how to detect location spoofing. [8] used tensor decomposition to spot anomalies in the check-in behavior of users. [9] used the Bayesian model to detect location spoofing by using millions of geo-tagged tweets. However, aforementioned approaches only consider check-ins data and ignore the location information contained in the user's microblog text, the method is not suitable for detecting fake check-in that continuously post from the same POI.

Considering the reliability of social network check-in data or the credibility of LBSN, effectively detecting spoofing location check-in data is an urgent problem need to be solved. Thus in this report, we use Sina Weibo [1] to investigate the location spoofing in social networks. Unlike the existing literature, we take both the user-generated content and check-ins data into consideration to detect location spoofing in social networks. Our contributions in this paper are twofold:(1) We introduce a practical way of location spoofing in Weibo, which can easily bypass the current location verification mechanisms. Our finding indicates that the current location verification mechanisms might not sufficient to prevent location spoofing. (2) We develop a novel location spoofing detection approach for quantizing the probability of fake check-in in social network, which can promote the credibility of LBSN and the reliability of spatial-temporal data.

2 Location Spoofing on Sina Weibo

Weibo has already launched a location service called Weibo Place, which allows users to check into special locations in Weibo, the check-in is done by hand, which means a user is able to determine where he/she wants to check in. When users post geo-tagged information, they are actually advertising the place. The check-in data is the concentrated reflection of customers to the POI.

Every check-in on Weibo is associated with a Weibo place page, and the microblogs containing the geo-tags related to the specified POI will display on the same page, the "total people" and "total check-ins" of the POI will also display

[1] https://weibo.com/

on the place page. According to our survey, if users want to share their real location information on Weibo, they can only check in with their mobile phones. But in fact Weibo uses short URL to represent POI, like "http://t.cn/z8Af3JR". As long as users get a URL of POI, he can publish check-in information on any client, which gives the user the possibility of spoof their location .

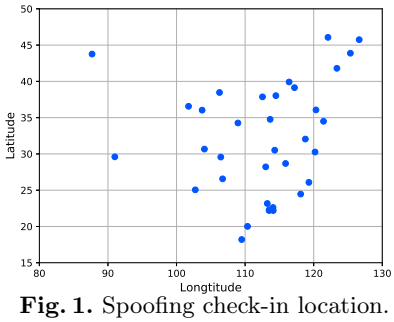

Fig. 1. Spoofing check-in location.

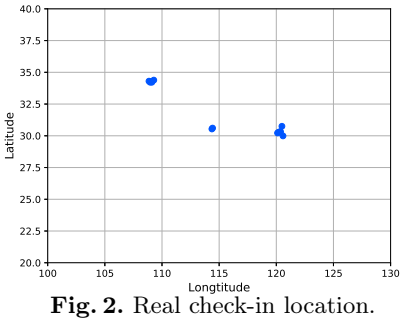

Fig. 2. Real check-in location.

To evaluate the location detection mechanisms deployed by Weibo, we reproduce the location spoofing scenario. First, we crawl 30 different venues' short URL throughout China, then we post microblog contained the short URL of POI. In this way, we can check into the same venue at any time interval. And we can also continuously check into different venues that are located far away from each other in just a few minutes. Meanwhile, we model a user to appear as traveling at any speed. Fig. 1 shows the check-in location of location spoofing within a few minutes. As we can see in Fig. 1, those venues are scattered pretty far apart and spread over 30 different cities throughout China. Fig. 2 shows the recent check-in record of a normal check-in volunteer, and the volunteer's check-in locate in three cities. Obviously, the distance elapsed between two consecutive check-ins cannot be so far in a very short time interval, due to the temporal and spatial constraints. Our experiments show that the location detection mechanisms of Weibo can be bypassed through our method. Unlike Foursquare, Weibo place do not set a limit on the number of check-ins in a given window and the user travel speed between two POI.

3 The Framework for Location Spoofing Detection

In this section, we introduce our approach for location spoofing detection in Weibo. As show in Fig. 3, there are three major parts in our scheme, contend proximity calculation, visiting probability computing and location spoofing estimation.

3.1 Problem Statement

The problem of location spoofing detection is to detect fake check-in data on specified POI. To simplify this problem, we set the detection target POI to the commercial center, we define the commercial center as a heterogeneous structure

that consists of parts which are different from each other, including shopping mall, restaurant and so on. For ease of presentation, we use POI, venue, and location interchangeably in this work. Formally, we are given a specified POI V, let U=$\{u_1, u_2, \ldots, u_M\}$ be a set of M users, where each user has the same geo-tag of V. Let G=$\{u_{1g}, u_{2g}, \ldots, u_{Mg}\}$ be a set of the microblog posted by user contained the same geo-tag of V. Our goal is to detect the fake geo-tags of G, the fake geo-tags are generated by user who intentionally falsifying their actual location information. For each POI, we have textual information in terms of geo-tag words, and the regional popularity of V in terms of how many people visited V, the popularity of POI is derived from the "total people" and "total check-ins".

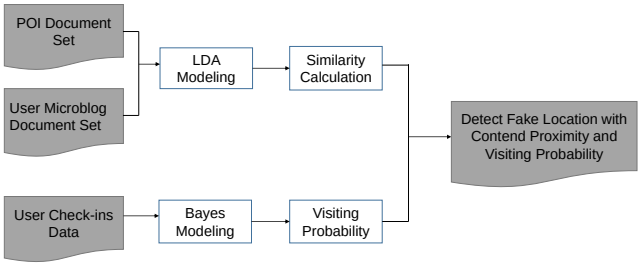

Fig. 3. The framework of location spoofing detection.

We deem that the rating of user launch location spoofing in POI is determined by the visit probability[9], the locational information of user's check-in with a lower visit probability is more likely to be fake. The probability of a user accessing the POI can be inferred by the follow factors: (1) each user's history microblogs contain information related to the POI. As semantic information is embedded in the microblog on LBSN that reflect the relevance of the user to the POIs, e.g, a restaurant name, environment, service, etc. It could help us estimate the probability of a user visiting a venue. (2) each user's history check-ins data associated with the POI can reflect the user's visiting probability at the POI. For these two reasons, we propose a method for location spoofing detection in social media by exploring the user's microblog textual content similarity with the POI and the check-ins data associated with the venue.

3.2 Microblog Textual Content Similarity

As aforementioned, users spoof their location by checking into a POI far away from where they really are so that they can get the rewards, which means those who launch a location cheating attack must be far away from the special POI.

We observed that the similarity of textual content between the user and POI is inversely proportional to geographic distance. Because user-generated content in social media mainly comes from user's lives in the physical world, the user tends to post content related to the geographically nearby POI. To validate this assumption, we study about 120,000 users sampled from Weibo, the sample consists of users with different distance from POI, we collect microblog generated by those sampled users. In Fig. 4, we plotted the similarity of the user-generated

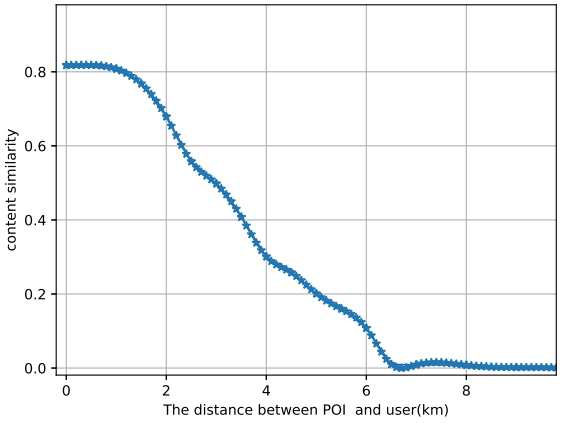

Fig. 4. The user-generated content similarity of different distance.

content and the target POI versus distance. We can see that the similarity of the user-generated content and the target POI monotonically decreases with the distance between them. Thus, the content similarity is a key factor to quantize the probability that a user launch location spoofing in the target POI.

For the above reasons, we caculate the similarity between user microblog content and POI information based on topic model, which provides methods to infer the latent topics from visible words, maps textual information corpus to each topic space. We use LDA to model textual information, and then obtain the topic-word probability distribution and document-topic probability distribution from the model, based on this calculate the content similarity by using the symmetric Jensen-Shannon divergence[10] between user u_i and the specified POI.

Topic Models Unlike the existed similarity calculation method, the similarity calculation based on topic model can mine the latent semantic location information. The LDA model [11] is a popular technique to identify latent information from large document collection. The basic idea of LDA is that every document is represented as the probability distribution of topics, and each topic is represented as the probability distribution of words, the generative process of LDA is as shown in Fig. 5.

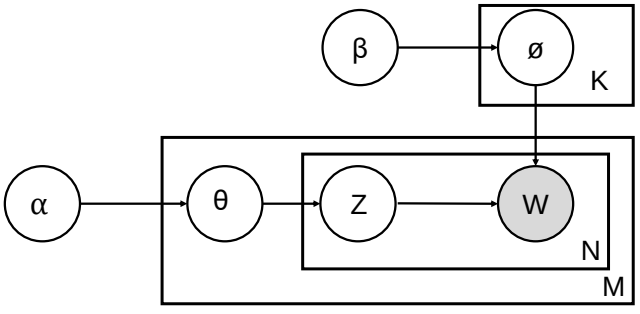

Fig. 5. LDA generation probability diagram.

To explore topics related to POI location of LBSN users, we propose to aggregate each user's history microblog into a user document d_{ui}, let D=$\{d_{u1},d_{u2},\ldots,d_{um}\}$, be a set of user document, one document of D is $d_{ui}=\{w_1,w_2,\ldots,w_n\}$, unique N words set of all the associated textual information. We combine all the microblog contained the same geo-tags of specified POI into a POI document d_v. To calculate the similarity between the user microblog and POI, we need to extract more semantic information of a given POI. Thus we crawl the location information matched POI from Meituan[2] to enrich the knowledge of the POI, then we conflate POI information of Weibo with Meituan. In this way, we get each user's microblog documents d_{ui} and the document d_v of POI V. By using LDA model, we get the topic distribution both of the user's microblog and POI. Each user u is associated with multinomial distribution over topic, represented by θ, each location-related topics is associated with a multinomial distribution over textual terms, represented by ϕ, the generation process of LDA model is as follows:

1. For each topic k,a word multinominal distribution ϕ_k is obtained from the Dirichlet(β) distribution.
2. For the document d_{ui} in D=$\{d_{u1},d_{u2},\ldots,d_{um}\}$,a topic distribution $\theta_{d_{ui}}$ is obtained from the Dirichlet(α) distribution.
3. For each word w_i in document d_{ui}:(a).extract a topic t from topic multinomial distribution $\theta_{d_{ui}}$.(b).extract a word w_i from word multinomial distribution ϕ_k of the topic.

From the LDA topic model, we can get the probability θ_{ij} that a user i associated with the topic k_j, and the probability distribution ϕ_i of topic i over the number of unique terms N in the dataset. We futher infer the topic distribution η of the specified POI based on the learned user topic term distribution. Therefore, we can compute the content similarity between the user and POI. According to Fig. 5, the joint distribution probability of all parameters of the whole model is obtained:

$$P(W,Z,\Theta,\Phi \mid \alpha,\beta)=\prod_{n=1}^{N_m} P(w_{m,n} \mid \phi_{z_{m,n}})P(z_{m,n} \mid \theta_m)P(\theta_m \mid \alpha)P(\Phi \mid \beta) \quad (1)$$

LDA Modelling In the process of LDA modelling, there are two latent variables that can be infered from the data: the user level document-topic distribution Θ , and the topic-word distributions Φ. We also need to infer the topic distribution η for the specified POI through the learned model. α and β are super parameter. In this paper, the method used for parameter estimation is Gibbs sampling [12]. Two matrixes can be obtained from Gibbs sampling, the calculation method is as follows :

$$\phi_{k,w}=\frac{n_k^{(w)}+\beta}{\sum_{w=1}^{v} n_k^{(w)}+N\beta}, \theta_{i,k}=\frac{n_i^{(k)}+\alpha}{\sum_{k=1}^{k} n_i^{(k)}+K\alpha} \quad (2)$$

[2] http://meituan.com/

where $n_k^{(w)}$ represents the number of words assigned to topic k, $n_i^{(k)}$ represents the topic observation counts for document d_{ui} of user u_i.N is the number of the unique words and K is the number of topics. Then we infer the topic distribution for POI is :$\eta_{jk} = \frac{n_j^{(k)}+\alpha}{\sum_{k=1}^{K} n_j^{(k)}+K\alpha}$,where $n_j^{(k)}$ is the topic observation count for POI document d_v.

Content Similarity Calculation After obtaining the document-topic probability distribution and topic-word probability distribution by constructing LDA model, the similarity calculation between user-generated documents d_{ui} and the POI document d_v can be realized by computing the corresponding topic probability distribution. This paper we use the symmetric JS (Jensen-Shannon)distance formula which can measure the distance of probability distribution to calculate the similarity between two documents. It is based on the Kullback-Leibler divergence. By applying Jensen-Shannon divergence to the topic assignment for the documents d_{ui} and document d_v, it will allow us to measure the similarity between the user and the POI. The distance vector p=(p_1,p_2,...,q_k) to q=(q_1,q_2,...,q_k) is computed as follows Equation(3):

$$D_{js}(p,q) = \frac{1}{2}[D_{KL}(p \,||\, M) + D_{KL}(q \,||\, M)], \tag{3}$$

where $M = \frac{1}{2}(p+q)$ and $D_{KL}(p \,||\, M) = \sum_{j=1}^{T} p_j \log \frac{p_j}{M_j}$ is Kullback-Leibler distance. p represents topic probability distribution of each user's documents d_{ui}, and q represents topic probability distribution of the POI document d_v. Then we define the similarity between the user-generated microblog and the POI information as formula:

$$S(u_i, V) = 1 - D_{js}(p,q) \tag{4}$$

3.3 POI Visiting Probability Based on Bayes Model

In addition to the user's microblog content information and the POI textual information, we have the user's historical check-in data. User's history check-in data can reflect the visiting probability at the POI. Rather than estimate the visiting probability by using historical check-in record directly, we take the popularity of POI into consideration. And use Bayes models to calculate the user's probability of accessing the POI. More concretely, let P(V) represents the visiting probability of user at POI, which can be calculated as the following equation:

$$P(V) = P(A|H) = P(A) \cdot \frac{P(H|A)}{P(H)} \tag{5}$$

where P(A) is the prior beliefs, given a user's historical check-in data, the total number of check-in data is N, and n is the number of user's check-in at POI, then $P(A) = \frac{n}{N}$. P(H) indicates the probability of an arbitrary user check into POI, it can be estimated by the popularity of POI. We can get the "total people" and

the "total check-ins" at POI from the POI page, then $P(H) = \frac{total \quad people}{total \quad check-ins}$. $P(H|A)$ is determined by C_n, C_n is the "total number" check-ins associated with POI. We define $P(H|A)$ as a piecewise function:

$$P(V) = \begin{cases} P(H), & C_n > p \\ P(A), & 0 < C_n \leq p \end{cases} \tag{6}$$

where p is the threshold to distinguish the densely populated areas and sparsely populated areas. When C_n>p, meaning the POI is a crowded place, human movements P(H) can be barely interfered by the activity of an individual P(A). Therefore, $P(H|A)$=P(H). When C_n<p, meaning the POI is a sparsely populated area, human movements P(H) can be affected by the activity of an individual P(A). Therefore, $P(H|A)$=P(A). Ultimately, P(V) can be formulated as follows:

$$P(V) = \begin{cases} P(A), & C_n > p \\ \frac{P(A)^2}{P(H)}, & 0 < C_n \leq p \end{cases} \tag{7}$$

For ease of evaluation, we count the total number of check-ins on different densely populated POI in Weibo, the statistical results show that the average number of check-ins on these POIs is no less than 1000, thus we set p=1000.

3.4 Location Spoofing Detection

Further, to quantize the probability of user's check-in data is fake in social network, we need to consider both (1) the similarity between user-generated content and the target POI information, and (2) the probability of the user accessing the POI. The similarity value can reflect the latent location information between user and POI, and the visit probability reflects the correlation between users and POI. Both the low similarity value and visit probability is more likely to be fake location. Thus we combine the similarity value and visit probability to quantize location spoofing as follows:

$$\vartheta = \lambda * D_{js}(p, v) + (1 - \lambda) * P(V) \tag{8}$$

where ϑ is the quantized value to estimate the probability of spoofing location information. The lower the value of ϑ, the greater the probability of forgery. λ is a factor to balance these two factors.

4 Experiments and Ananlysis

The experiments were performed on real-world social network dataset collected from Weibo. First, we collect all the microblogs containing the geo-tags related to the specified POI, including user ID, microblogs content, Weibo source, microblogs posting time. Then we use the user ID that collect in the specified POI to collect microblogs generated by these sample user, and filter out those users

whose historical microblogs is fewer than 20 and check-in data is less than three times on the target POI. Filter out users with less than three check-ins at the target POI for the following two reasons:(1) If the users want to be rewarded by falsifying their location to the target POI, his number of check-ins related to the POI must accumulate to the specified value. (2) We set the specified value to 3, in this case, we can effectively avoid misjudgment of users who have checked into the target once. After applying this filter, we crawled over 2,000 user's historical data, a total of 120,000 microblogs.

There is no formal ground truth to label LBSN user's check-in data in fake. Hence, we invite volunteer to label these users who post fake geo-tags manually according to the forgery features. As mentioned in section 3, if the user's check-in data is legitimate, he can only post the check-in information via the mobile client, and the check-in data should not violate the temporal and spatial constraints. For these reasons, we manually selected the microblogs that contain the target POI geo-tag posted by non-mobile client. And pick out microblogs that obviously violates the space-time constraints. Then we mark those checked-in messages as fake, which can be treated as ground truth.

4.1 Evaluation Metrics

For the evaluation, accuracy, precision, recall and F1-score are the most common used evaluation metrics. Thus in our experiments, we adopt these four metrics defined below to measure performances of our method. Where TP (True Positive) represents the fake geo-tags correctly identified as fake, FP (False Positive) represents the true geo-tags incorrectly identified as fake, TN (True Negative) represents the true geo-tags correctly identified as true, FN (False Negative) represents the fake geo-tags incorrectly identified as true.

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|}, \qquad Precision = \frac{|TP|}{|TP| + |FP|} \tag{9}$$

$$Recall = \frac{|TP|}{|TP| + |FN|}, \qquad F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{10}$$

4.2 Experimental Results

For the probabilistic framework of detecting spoofing location in the dataset, we further need to set the parameter λ in the formulate $\vartheta = \lambda * D_{js}(p, v) + (1 - \lambda) * P(V)$. When λ=1, it means that the quantized value ϑ is only determined by the user-generated microblogs, when λ=0, it means that the quantized value ϑ is only determined by the user check-in record. and $0<\lambda<1$ means that the probabilistic framework is determined by these two factors.

In the LDA modeling process, the accuracy of LDA clustering results will be affected by the number of topics k, so before setting the value of λ, we

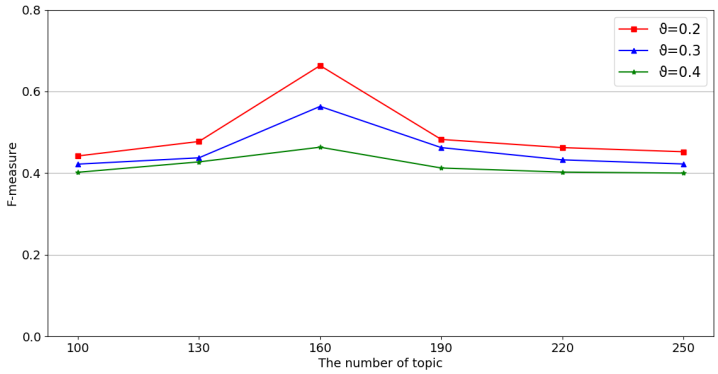

Fig. 6. Relationship between the number of topics and F-measure .

need to determine the value of k. We use Gibbs sampling algorithm to evaluate parameter, the number of iterations of Gibbs sampling is 2000. According to the empirical value, we set α =50/k and β = 0.01. As the topic k and quantized value ϑ will directly affect the precision of LDA model, which will affect the detection result, we determined the value of k and ϑ by experiment, and set λ=1 . The F-measure is higher, the experiment result is better. As shown in Fig. 6, different number of topics generates different F-measure.

Here the abscissa represents the number of topics, the vertical coordinate represents the value of the F-measure. The change of F-measure with topic K is shown in Fig. 6. Since most of the content similarity values are around 0.6, we set the ϑ as 0.2,0.3,0.4. In Fig. 6, we can see that under different values of ϑ, F-measure values change with respect to topic K. When ϑ =0.2, K=160, the F-measure reaches the maximum. So the number of topics is determined as 160, and the value of ϑ is determined as 0.2 in our case.

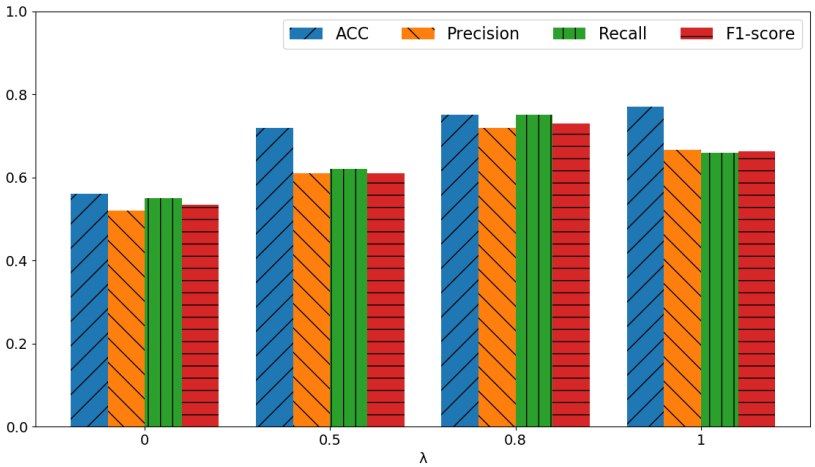

Fig. 7. Accuracy,precision, recall and F1-score with different λ .

To further investigate the effect of the user-generated microblog and the user history check-in record on the probabilistic framework, we perform the experiment by adjusting λ, which controls the weight of this two different factors.

Fig. 7 shows the experimental results at different λ. When λ=0, it means that the experimental result is determined by the user check-in record, and the detection of spoofing location information is based on Bayes Model. When λ=1, it means that the experimental result is determined by the user-generated microblogs, and the detection of spoofing location information is based on "LDA+JS". When $0<\lambda<1$ means that the experimental result is made by combining both the user-generated microblogs and the user history check-in record, and the detection of spoofing location information is based on "LDA+JS+Bayes Model".

Table 1. Experimental results of different methods.

Experimental method	***Precision Ration***	***Recall Ration***	***F Metic***
Bayes Model	0.523	0.551	0.537
LDA+JS	0.667	0.660	0.663
LDA+JS+Bayes Model	0.724	0.752	0.738

Table 1 shows the experimental results of different methods. We can see that the combination of user-generated microblog and the user history check-in record achieves the highest precision, which means the combination of user-generated microblog compensate the user history check-in record to improve the experimental results. The results demonstrate that we can achieve 20 percent relative improvement over state-of-the-art approaches, it proves that combining these features together can achieve a better performance.

5 Conclusion and Future Work

A method for location spoofing detection in social networks is proposed in this article, and the results show that our approach can detect certain types of spoofing location. Meanwhile, we launch location cheating attack that enables us to check into a venue far away from our real location, and demonstrate the LBS are vulnerable and the true impact of fake locations in social networks.

As the counter measures against location cheating can be bypassed, a method is proposed to detect location spoofing in social networks. Even though our method is to detect the target POI of commercial center, we believe that the method we proposed can open the door to further research in this field. Regarding the future work, we will try to obtain more social context information (e.g., user's social groups) because geographically nearby users are more likely to publish similar geo-related content. The proposed model adopts LDA (based on bag-of-words model) to get the topic distribution of each users and POI, however, it is not based on context. Therefore, extracting location information from POI and user's generated-content by utilizing neural networks (e.g.,Word2Vec [13]) is a part of our future work.

Acknowledgment This work is supported by the cyberspace security Major Program in National Key Research and Development Plan of China under grant 2016YFB0800201, Natural Science Foundation of China under grants 61572165

and 61702150, State Key Program of Zhejiang Province Natural Science Foundation of China under grant LZ15F020003, Key Research and Development Plan Project of Zhejiang Province under grants 2017C01062 and 2017C01065, and the Scientific Research fund of Zhejiang Provincial Education Department under grant Y201737924, and Zhejiang Provincial Natural Science Foundation of China under Grant No. LGG18F020015.

References

1. Janne Lindqvist, Justin Cranshaw, Jason Wiese, Jason Hong, and John Zimmerman. I'm the mayor of my house: examining why people use foursquare-a social-driven location sharing application. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2409–2418. ACM, 2011.
2. Sameer Patil, Gregory Norcie, Apu Kapadia, and Adam Lee. Check out where i am!: location-sharing motivations, preferences, and practices. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 1997–2002. ACM, 2012.
3. Wenbo He, Xue Liu, and Mai Ren. Location cheating: A security challenge to location-based social network services. In *Distributed computing systems (ICDCS), 2011 31st international conference on*, pages 740–749. IEEE, 2011.
4. Feng Zhang, Aron Kondoro, and Sead Muftic. Location-based authentication and authorization using smart phones. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1285–1292. IEEE, 2012.
5. Iasonas Polakis, Stamatis Volanis, Elias Athanasopoulos, and Evangelos P Markatos. The man who was there: validating check-ins in location-based services. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 19–28. ACM, 2013.
6. Zengbin Zhang, Lin Zhou, Xiaohan Zhao, Gang Wang, Yu Su, Miriam Metzger, Haitao Zheng, and Ben Y Zhao. On the validity of geosocial mobility traces. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 11. ACM, 2013.
7. Gang Wang, Sarita Yardi Schoenebeck, Haitao Zheng, and Ben Y Zhao. " will check-in for badges": Understanding bias and misbehavior on location-based social networks. In *ICWSM*, pages 417–426, 2016.
8. Evangelos Papalexakis, Konstantinos Pelechrinis, and Christos Faloutsos. Spotting misbehaviors in location-based social networks using tensors. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 551–552. ACM, 2014.
9. Bo Zhao and Daniel Z Sui. True lies in geospatial big data: detecting location spoofing in social media. *Annals of GIS*, 23(1):1–14, 2017.
10. Jianhua Lin. *Divergence measures based on the Shannon entropy*, volume 37. IEEE, 1991.
11. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
12. Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.
13. Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.