World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:8, 2021

The Application of Fuzzy Set Theory to Mobile Internet Advertisement Fraud Detection

Jinming Ma, Tianbing Xia, Janusz R. Getta

Abstract—This paper presents the application of fuzzy set theory to implement of mobile advertisement anti-fraud systems. Mobile anti-fraud is a method aiming to identify mobile advertisement fraudsters. One of the main problems of mobile anti-fraud is the lack of evidence to prove a user to be a fraudster. In this paper, we implement an application by using fuzzy set theory to demonstrate how to detect cheaters. The advantage of our method is that the hardship in detecting fraudsters in small data samples has been avoided. We achieved this by giving each user a suspicious degree showing how likely the user is cheating and decide whether a group of users (like all users of a certain APP) together to be fraudsters according to the average suspicious degree. This makes the process more accurate as the data of a single user is too small to be predictable.

Keywords—Mobile internet, advertisement, anti-fraud, fuzzy set theory.

I. Introduction

MOBILE internet advertisement fraud is a series of user actions, usually aiming to gain advertising expenses from advertisers. The major difference between fraudulent actions and real user actions is that real user actions on advertisements usually show the attention and even interest of users in the advertisement. It means that there is a probability that advertisers may gain profit from the real users. On the other side, fraudulent actions do not bring any user's attention to the advertisers, and it is nearly impossible for advertisers to gain any profit. This is a serious damage to the interests of advertisers and in the same time to the industry environment of mobile internet. An example below may help to get a better understanding of the problem. Let's suppose that, Alice manages a mobile APP company with a large amount of traffic. As Alice is rich in traffic, a lot of advertisers wish to sever their advertisement on Alice's APP. This brings Alice much profit. Although Alice has already gained a lot from what she already have, she always wants to get more. Unfortunately, it is not possible because the profit Alice could earn is limited by the traffic. Alice cannot make more money without increasing her traffic. Thus, to gain more profit, Alice comes up with two methods of increasing her 'traffic'.

The first method is to cheat through generation of false traffic. Alice may choose to do so, but the false traffic will increase the input-output rate of the advertiser, making advertisers unwilling to cooperate in the future. In the long run, this is not a good choice.

Jinming Ma, Tianbing Xia, and Janusz R. Getta are with The School of Computing and Information Technology, The University of Wollongong, Wollongong, Australia (e-mail: jm662@uowmail.edu.au, txia@uow.edu.au, jrg@uow.edu.au).

The second method is to build an advertisement serving platform. With such a platform, Alice may sever advertisements to many other APPs that are willing to gain advertisement budget from her. This method can bring real traffic and income to Alice. As a matter of fact, not all the APPs have enough traffic to attract advertisers. An advertisement serving platform is now so popular that almost every large company builds one of their own.

After comparing the two options, Alice chooses the second. She easily found that many of APP runners would be happy to cooperate with her. Bob is one of them. Bob is an APP runner with little traffic. Like Alice he wishes to gain more profit. Unlike Alice, he cannot build a advertisement serving platform. This is because he has little traffic and less customers are willing to advertise their products. His company is not large enough to run a platform. Thus, Bob chooses to cheat. Bob generates fake traffic and gains more profit than he should from Alice's platform. This results in the increase of input-output rate of Alice's customers. The advertisers loose their money for advertising on fake traffic. Alice looses the trust of her customers. All because of the fraudster Bob. This is why mobile fraud is a damage to the interests of advertisers and the industry environment of mobile internet at the same time. Unfortunately, mobile advertisement fraud is not only harmful but also attractive. Gaining much profit by doing only a few simple technical tricks without worrying about traffic at all is much easier than managing a real honest APP. More importantly, such technical tricks are easy to run, but hard to identify.

The mobile fraud and anti-fraud are like a competition. Each side of this competition is trying their best to create the new methods and defeat the competitor. One side of this competition is a group of fraudsters, the other one is the mobile app companies. Mobile internet advertisement anti-fraud systems are mostly developed by mobile app companies. This is because almost all app companies cooperate and trade traffic and advertisement budget with the other companies. These companies, however, who develop their own methods of fraud detection, are usually not willing to share their progress in this area. This is reasonable as the fraudsters could learn how to avoid being identified when an anti-fraud method is known to public.

Although this is not an open research area, there are still a few references, that offer the descriptions about mobile fraud and anti-fraud methods. Pooranian et al. [4] introduced different fraud and anti-fraud methods; Tian et al. [3] described a specially designed anti-fraud method against crowd fraud (device-based fraud in this article) and Oentaryo et al. [2]

used a method of data mining approach to detect click fraud on online advertisement. Besides these references, there are also a few websites, that offer suggestions about related to fraud detection.

The major difference between the anti-fraud method proposed in this work and the methods developed in the past is such that we determine for each user a measure of how likely the user is a fraudster. The other works, like [3] divided users into 3 groups with two different indices, and only one of the groups was assumed to be suspicious. It is reasonable to do that, but our method can be more accurate.

The reason behind this research is the hardship in the exact detection of fraudsters. As the behavior of normal users is unpredictable, there is no way to very precisely distinguish fraudsters from honest users. For example, a click rate can be used in fraud detection, because a normal user's click rate is usually less than 5%. But we cannot say, that a user is a fraudster if his or her click rate is higher than 5%. There is no number that could categorically distinguish fraudsters from normal users. To solve this problem, we propose to apply a method based on fuzzy statistics. If it is hard to identify a fraudster, then we can calculate a suspicious degree of how likely a user is cheating. It can be done for every user instead of dividing the users into two categories.

This work mainly focuses on using fuzzy set theory to measure how suspicious the activities of a user or app can be. This measurement is called suspicious degree. With the calculation of suspicious degree, we can avoid the hardship in making a conclusion on whether a certain user is a fraudster. As a matter of fact, the benefit of this method is such that we can avoid making conclusion on a small data sample, like one single user. Instead, we use the suspicious degree to represent the most important feature of one certain log or user and make a conclusion after collection of all suspicious degrees of a certain app or source.

In this article, some past work and some fraud and anti-fraud methods will be introduced in Section II. Section III contains the topic of fuzzy set method. In Section IV, we give the descriptions of the implementations by using the fuzzy set method.

II. Methods of Mobile Internet Advertisement Fraud and Anti-Fraud

A. Methods of Mobile Fraud

The mobile advertisement fraud methods have been developed almost since mobile internet became popular. As a result, there is a large number of different methods for implementations of fraud advertisements. These methods can be classified into three types below.

1) Fake Users: The first method creates the fake users. The fake users are not real human users generated by fraudsters. There are two ways to generate a fake user.

Real device based fake user method, also called as a device farm, is usually a group of people working together, such that each person operates tens of devices at the same time. Such a method of fraud usually results in a high cost event rate, density, and very short time difference between cost events. Sometimes, to avoid being detected, fraudsters may reset the phone number, user ID, device ID, IP address frequently. This will result in a high percentage of the fake new users.

Sever based fake users method is a method that requests and reports directly from the fraudster's server to the advertiser's or advertising server platforms without using any smart phone device. In such a fraud method, nearly all the information related to a user is generated by the server. Some fraudsters can even generate users so perfectly that the data looks even more 'normal' than real user's data. The weak point of such a method is that to avoid being identified, the fraudsters usually generate new users too frequently. In this method a 'user' may last less than one hour. Thus, monitoring the density of the arrivals of new users can be used against such a fraud method.

2) Fake Actions of Real Users: Fake actions of real users is a type of fraud method that happens on real users' smartphones. Fraudsters usually use their APPs installed on real users' smartphones to collect data and to generate the reports from fake actions to gain profit from the advertisers. The fake actions may or may not actually happen, but they are always not what the real users intend to do. This type of fraud methods were popular few years ago and it became rare now. It is because such methods usually require the users to install APPs on their devices with thew pretty awful user experience. However, one of the methods called attribution misleading is still frequently seen.

The attribution misleading method is a method focusing on cost per download or cost per install advertisements. Cost per download (CPD) or cost per install (CPI) are billing modes of advertisements, which is a billing mode that the advertisers pay for only successful downloads or installs of users. One of the reasons why advertisers would like to choose this billing mode is such that advertisers usually have the exact data of downloads and installs of their own APPs. In the other words, data of downloads and installs are usually trustworthy for the advertisers. Such billing mode requires an logical attribute. When a download or install happens, the advertisers would check if there is any click event reported from the same user a little bit earlier, usually no more than 24 hours. If there is no 'click' reported, then advertisers would assume, that the download or install happened naturally. If there is one 'click' report, then advertisers should pay for the downloads and installations on the medias who sent the report. If there is more than one media who sent 'click' reports, then advertisers will choose one of the media to pay, usually choose the last media who sent the report. In this case, it is not wise for fraudsters to make fake downloads and installation. Because it is not hard for advertisers to notice, that there is a number of users from the same media who downloaded their APP but not active at all. Thus, fraudsters tend to choose to report a fake click when a real download is detected. Achieving this would mean that they are almost definitely the last media who reported. To deal with this fraud method, each cost event of download type advertisement should be checked to see if a cost event origin exists.

Some other fraud methods of this type include *click redirection* and *forced click*.

Click redirection happens mostly on mobile web, where a sscript could redirect a user's first click to the page or a hidden link to load an unexpected page.

Forced click is a fraud method that happens when the whole screen is covered by an advertisement without a clear close option forcing the users to click on the advertisement.

In advertisement stacking method, fraudsters hide advertisements behind an advertisement, article, video, or anything displayed on their APPs. When a user clicks what has been shown above, click events of all advertisements behind are reported.

3) Induced Real Actions of Real Users: Ever since 2018, some mobile technology companies in China found it profitable to develop a type of APP which offers a little bit of cash (a user gets less than 1 CNY a day, if the user works really hard on the APP) to the users in exchange for the users to click and read the advertisements on their APPs. This kind of APP, usually called *user inducing APPs*, has been proven to be attractive to so called 'sinking users'. These are the users commonly assumed to be large in number and with low consumption power. Since the user inducing APPs have a really large group of target users, and also truly inspired the users to be active, such APPs are much more profitable than normal APPs. Such APPs are profitable for the developers, maybe also for the users, but not for the advertisers. As mentioned, the target users usually have low consumption power, they are clicking the advertisement to gain money, not for the interest of the product. The advertisers would like to pay for clicks because they believe that clicking shows a chance, that a user would like to pay for the product, which is meaningless in user inducing APPs. Thus, even though both the actions and a user inducing APP are real, the APP developer is still a fraudster. As all the users are real, this fraud method is the hardest to identify. Overactive monitoring might help, but clever developers have learnt to encourage the users to act 'normally'. Choosing not to cooperate with them may seem to be a good choice, but as they are rich in traffic, they can always get a budget from other media or advertisement serving platforms. In another word, advertisers will never know who 'sold' their advertisement budget to a user-inducing APP company.

III. Application of Fuzzy Set Theory

As it has been mentioned earlier in this paper, the exact identification of fraudsters is usually impossible. In this case, fuzzy set theory can be helpful.

There are many articles and books about fuzzy set theory and fuzzy statistics. Mordeson [8] defined and explained fuzzy set, fuzzy mapping, fuzzy logic and other contents. Uga-Rebrovs [7] made a specific description about the fuzzy random variable. Kandel et al. [5] introduced fuzzy set, algebra and statistics detailly. In this work, we will follow the definitions in [1] by Buckley.

A. Definitions

We repeat the definitions of fuzzy subset, fuzzy number and α-cuts in [1] below.

1) Fuzzy Subset: Given a set A, a fuzzy subset B of A is defined by its membership function $B(x)$ with values in $[0,1]$ for all x in A. If $B(x_0) = 1$, then x_0 belongs to B. If $B(x_0) = 0$, then x_0 does not belong to B. If $B(x_0) = h$, where $0 < h < 1$, then the membership degree of x_0 in B is h.

2) Triangular Fuzzy Number: Given three real numbers $a < b < c$, then a triangular fuzzy number $N = (a/b/c)$ can be defined as a fuzzy subset of R with membership function defined as:

$$f(n) = \begin{cases} 0 & (x \le a) \\ \frac{b-x}{b-a} & (a < x \le b) \\ 1 - \frac{c-x}{c-b} & (b < x \le c) \\ 0 & (x > c) \end{cases}$$

In this paper, other types of fuzzy number will not be discussed. Thus, when referring to a fuzzy number, we assume that it is always triangular fuzzy number. Buckley also used the application of triangular fuzzy number in [6].

3) $\alpha - Cuts$: Given a fuzzy number N, an $\alpha - cut$ of N is defined as $N(\alpha) = \{x \in R | N(x) \ge \alpha\}$ where $0 < \alpha \le 1$.

B. Application of Fuzzy Statistics on Anti-Fraud Methods

For example, we consider the overactive check method. Let U be a set of all users recorded in an hour. Let set O be the fuzzy subset of all overactive users in U. Then the membership function of a user in a certain hour can be defined as

$$O(u) = max(min(1, max(0, 10s - MTD)), \\ min(1, costeventdensity \times costeventrate)), \quad (1)$$

where MTD is the minimum time difference between two cost events (if exist). This membership function will be called suspicious degree and it will be written as sups-degree.

suspicious degree of a user in an hour defined as above is used to define the suspicious degree of a source or APP on a day. The suspicious degree of a source or APP on a day is an $\alpha - cut$ of a fuzzy number: $(min(sups-degree)/avg(sups-degree)/max(sups-degree))$. This is actually not typically a real $\alpha - cut$ of a fuzzy number, as the fuzzy number and the α are both undefined. But given the circumstance, it is safe to assume that there exist such a number and α satisfying the suspicious degree. This suspicious degree of a source or APP on a day is the final result that determines the validity of a user. The $avg(sups-degree)$ will show how likely the whole source or APP is cheating, while the $min(sups-degree)$ and $max(sups-degree)$ will show the behavior of normal users and fraudsters in the source or APP.

IV. The Anti-Fraud Data Analyzing Process and Database

The main target of this paper is to find a method that can avoid making conclusion on user level. The idea of the method, as introduced previously, comes from the fuzzy set theory. To apply this method, we created an anti-fraud data analyzing process and its supporting database.

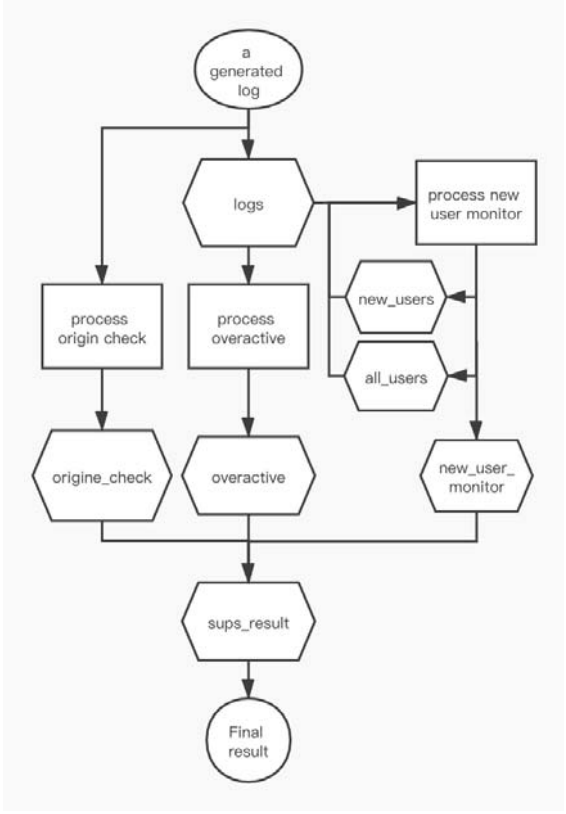

Fig. 1 Anti-fraud data analysis process

A. Introduction of The Main Process

There are three main processes analyzing different types of fraudsters. The processes include: *origin check process*, *overactive monitor process* and *new user monitor process*. All data obtained from the recording of the users' activities are saved in 7 relational tables: logs, origin-check, all-users, new-users, new-users-monitor and sups-result.

The data flow diagram that overviews the entire anti-fraud data analysis process can be seen in Fig. 1.

B. Process Origin Check

To identify attribution misleading fraudsters, we designed the *origin check process*. Each time when a downloaded type log with $event = cost_event$, this process will check if a show event log with the same ads_id exists in the logs table. If not, the log is very suspicious to be cheating. Otherwise, the process will calculate the sups_degree accordingly to the time difference between show log and cost event log, as it is also suspicious if the time difference is rather small. This process starts each time when a log is updated to the logs table. Related details can be seen in Fig. 2.

C. Process Overactive Monitor

This process analyzes if any user is too active to be normal. Usually, users will not click or download on advertisements frequently. On the contrary, fraudsters would report cost event logs in a much higher density. That would result in higher

Algorithm 1 Origin Check

```
given a log updated to table logs
then log[x] is the value of element x of the given log
if log[event] = log[cost_event] != 'show'
   and log[ad_type] = 'download' then
   select timestamp from logs
   where ads_id=log[ads_id] and event=log[cost_event]
   if timestamp exist then
      show_exist ← 1
      timedif ← log[timestamp]-timestamp
      sups_degree ← min(1, max(0, 1 - timedif/1000))
   else
      show_exist ← 0
      timedif ← 0
      sups_degree ← 1
   end if
end if
insert into origin_check values
(log[log_id:event] + (show_exist, timedif, sups_degree))
```

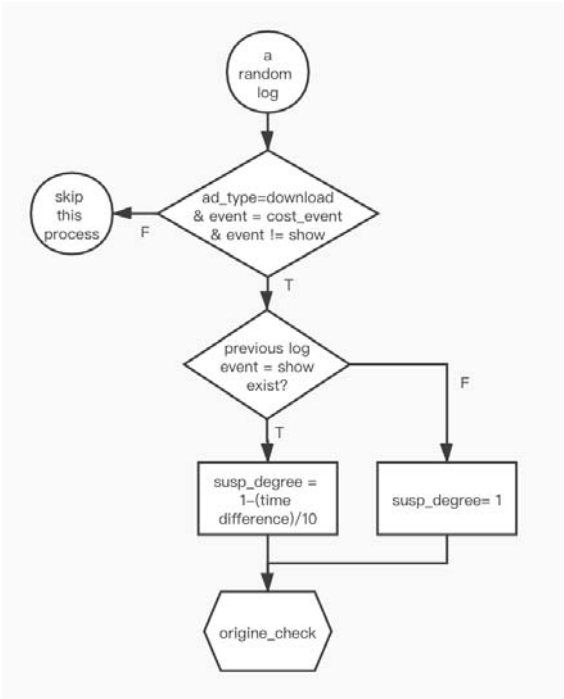

Fig. 2 Process origin check

cost event rate, density and smaller cost event time difference. The overactive monitor process will analyze these indexes to identify the fraudsters. Every hour, the overactive monitor process will query data from the table logs to analyze if users are overactive in the hour. Related detail can be seen in Fig. 3 (all operations are only applied to the data of the certain hour).

D. Process New User Monitor

The overactive process is effective against many fraudsters, but if the fraudsters, like server based fraudsters, change their

Algorithm 2 Overactive Monitor

query the database
let sel_1 be a tulple of cost event rate and density of every user in the logs table
each element of this tulple represents a user
for each tulple i in sel_1 **do**
  timestamp = 0, min_timedif = 1000000
  if i[costEvent_density] >= 2 **then**
    query the database
    let sel_2 be the tulple of timestamp of the given user's all cost event logs whose cost event is not show
    each element of this tuple represents a cost event
    for each tulple j in sel_2 **do**
      min_timedif ←
        min(min_timedif, j[timestamp] - timestamp)),
      timestamp ← j[timestamp]
    end for
    sups_degree ←
      max(min(1, max(0, 1 - min_timedif/10000)),
        min(1, i[costEvent_rate] * i[costEvent_density]))
  else
    sups_degree ← 0, min_timedif ← 1000000
  end if
  insert into overactive values
    (i+(min_timedif, sups_degree))
end for

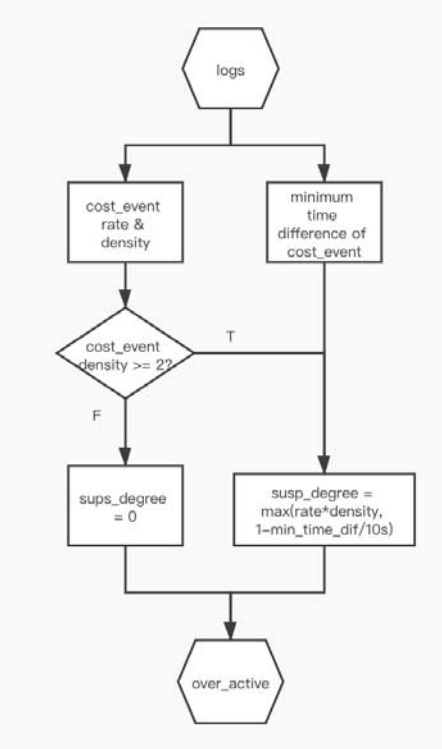

Fig. 3 Overactive monitor process

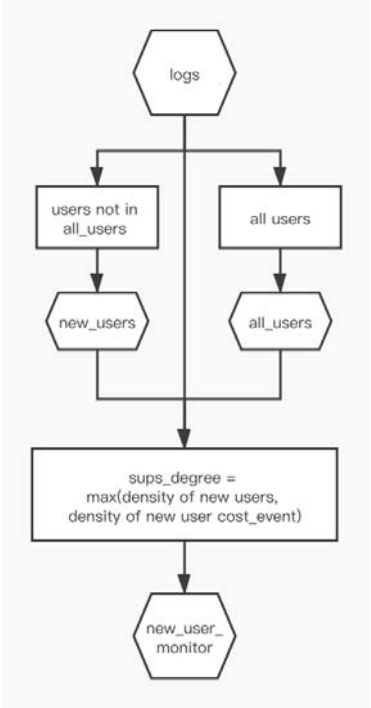

Fig. 4 Process new user monitor

ID frequently, it is going to be hard for the overactive process to collect enough data for analyzing. In this case monitoring the percentage of new users and cost event of new users, namely NU_density and NU_action_density in the process, could easily identify cheating APPs. Every hour, the New User Monitor Process will query data from table logs to analyze if any source or APP contains too many new users. Related detail can be seen in Fig. 4(all operations are only applied to data of the certain hour).

Algorithm 3 New User Monitor

query the database
let sel_1 be the tulple of users that showed up in logs but not in all_users
each element of sel_1 is a user
update sel_1 to new_users diagram
query the database
let sel_2 be the tulple of all users that showed up in logs
each element of sel_2 is a user
update sel_2 to all_users diagram
query the database
let sel_3 be the tulple of new user density and new user action density of each app
each element of sel_3 represent an app
for i in sel_3 **do**
  sups_degree ←
    max(i[NU_density], i[NU_action_density])
  insert into new_user_monitor values (i+(sups_degree,))
end for

E. Table Sups_Result

As a day passes, all results of the three processes will be collected and updated into the sups_result table, the table that contains the finial result and offers the summarised suggestion of the whole process. Since this table will be used in the next chapter to show the result of our text, the table will be specially introduced here in detail.

The columns of the sups_result table are listed in Table I.

TABLE I
DESCRIPTION OF COLUMNS IN SUPS_RESULT

Column	Description
app_id	Unique identity of each APP.
date	Date when the event happened.
OA_min_sups	The minimum supsDegree in table overactive.
OA_avg_sups	The average supsDegree in overactive.
OA_max_sups	The maximum supsDegree in overactive.
NU_min_sups	The minimum supsDegree in table new_user_monitor.
NU_avg_sups	The average supsDegree in new_user_monitor.
NU_max_sups	The maximum supsDegree in new_user_monitor.
OC_min_sups	The minimum supsDegree in table origin_check.
OC_avg_sups	The average supsDegree in origin_check.
OC_max_sups	The maximum supsDegree in origin_check.
OC_log_num	The number of logs in the origin_check.

V. TESTING LOG GENERATOR AND TEST RESULT

A. Testing Log Generator

To test if the anti-fraud process in functional identifying fraudsters, we designed a generator that could simulate normal users and fraudsters. The generator will generate logs from 4 different apps, namely Alice, Bob, Chris, David.

Of the 4 apps, Alice is the only normal source, with only real human behavior like users. Bob is the attribution misleading fraudster, which has a lot of common users as Alice, and will always generate a cost event log if any of these common users reported a cost event log of a download type advertisement on Alice. Apart from this, Bob is all the same as Alice. Chris is a sever based fraudster, with the rate of both cost event and new user generation being much higher than Alice and Bob. David is a device based fraudster, with the new user generation rate a little bit higher and cost event generation rate much higher than Alice and Bob.

There's another difference between the 4 apps. Since Alice and Bob contain mostly human-like users, the time difference between different events of the 2 apps are a little bit longer than Chris and David. Also, as Chris is fully automatic sever based fraudster, and David is supposed to be operated by human, the time difference between events of Chris is even shorter than David, being the shortest of the 4 sources.

B. Test Result

Tables II and III are the results of the two tests. According to the test, the process is functional as intended.

TABLE II
TEST RESULT 1

app_id	Alice	Bob	Chris	David
OA_min	0	0	0	0
OA_avg	0	15.6%	4.9%	64.6%
OA_max	0	1	1	1
NU_min	4.9%	7.2%	98.3%	19.5%
NU_avg	4.9%	7.2%	98.3%	19.5%
NU_max	4.9%	7.2%	98.3%	19.5%
OC_min	0	0	0	0
OC_avg	41.0%	94.1%	2.2%	1.6%
OC_max	81.0%	1	76.5%	72.4%
OC_num	2	70	206	147

TABLE III
TEST RESULT 2

app_id	Alice	Bob	Chris	David
OA_min	0	0	0	0
OA_avg	0	11.0%	4.3%	66.8%
OA_max	0	1	1	1
NU_min	10.5%	8.5%	97.9%	25.6%
NU_avg	10.5%	8.5%	97.9%	25.6%
NU_max	10.5%	8.5%	97.9%	25.6%
OC_min	0	0	0	0
OC_avg	0	89.0%	2.4%	4.2%
OC_max	0	1	87.1%	85.2%
OC_num	7	51	199	185

VI. SUMMARY

One of the most severe hardships is that there is little evidence to prove whether a user is a fraudster. This paper aims to solve the problem using the application of fuzzy statistics. By measuring how likely a log, user or app is cheating, identifying fraudsters no longer has to be 100% accurate. Instead, the suspicious degree could offer a numerical view for advertisers to analyze fraudsters. This method can be a solution for mobile internet advertisement anti-fraud system.

REFERENCES

[1] Buckley, James J. "Fuzzy Probability and Statistics." *Studies in Fuzziness & Soft Computing* 196(2006):236.
[2] Oentaryo R., Lim E P., Finegold M. et al. Detecting Click Fraud in Online Advertising: A Data Mining Approach (J). *Journal of Machine Learning Research*, 1990, 15:99-140.
[3] Tian T., Zhu J., Xia F. et al. Crowd Fraud Detection in Internet Advertising (C). The 24th International Conference. *International World Wide Web Conferences Steering Committee*, 2015.
[4] Pooranian Z., Conti M., Hadaddi H. *Online Advertising Security: Issues, Taxonomy, and Future Directions (J).* arXiv preprint arXiv:2006.03986, 2020.
[5] Kandel A., Byatt WJ. Fuzzy sets, fuzzy algebra, and fuzzy statistics (J). *Proc IEEE*, 1978, 66(12):1619-1639.
[6] Buckley J J . Fuzzy statistics: regression and prediction (J). *Soft Computing*, 2005, 9(10):769-775.
[7] Uga-Rebrovs O, Kueova G. Specific Features of Descriptive Statistics with Fuzzy Random Variables (J). *Information Technology and Management Science*, 2018, 21:104-110.
[8] Mordeson J N . *Fuzzy Mathematics (M).* Physica-Verlag, 2001.