Research on Rough Set Theory for Mobile Fraud Detection

Jinming Ma, Tianbing Xia, and Janusz Getta
(Corresponding author: Jinming Ma)

School of Computing and Information Technology, University of Wollongong
Northfields Ave Wollongong, NSW 2522 Australia
Email: jm662@uow.edu.au

(Received Dec. 19, 2021; Revised and Accepted May 6, 2022; First Online July 3, 2022)

Abstract

Since smartphones and mobile internet became popular, mobile advertisement fraud and anti-fraud are two competitors. One tries to suppress the other. Every time a new fraud method is utilized, a specially designed anti-fraud method will come out soon. We propose a mobile anti-fraud method that uses the rough set theory to end this circle. This method does not target any particular fraud method but observes the differences between user groups. As long as the fraudsters do not own the related data of natural user groups, it is almost impossible for fraudsters to avoid being detected by this method.

Keywords: Fraud Detection; Mobile Advertisement; Rough Set Theory

1 Introduction

Giving a proper definition of a mobile internet advertisement fraudster is never an easy job. It is not only because of the differences between different fraud methods, but also due to the differences in the intention of different fraudsters. Professional fraudsters aim to gain profit from advertisers. Competitors may also try to consume competitive advertiser's advertisement budget. Besides all the above, the user inducing apps make the definition and detection of mobile internet fraud even more complicated.

This work focuses on the most common and most damaging fraud actions and fraudsters.

Profit-aiming fraud actions: user actions (usually navigation over the links) on mobile advertisements that intend to gain profit from advertisements other than being attracted to learn more about the information from them.

As no fraudster will use single user account or IP address for cheating, they should be defined on multiple users, namely a group of users.

Profit-aiming fraudsters: groups of users that commonly perform profit-aiming fraud actions.

As addressed in the definitions above, one of the most important differences between normal user actions and fraudster actions are whether the user is attracted or interested in the advertisement. Thus, fraudsters can be detected as long as we know whether the users are interested in the advertisements they accessed.

Although it is almost impossible to know whether every user is interested when they click at a link, it is not hard to analyze whether a group of people are interested in an advertisement. For example, advertisements for video games are more attractive to young people than to elder people. Females should be more interested in makeup advertisements than males. It may not be correct for a single user, but it is almost true for a large group of users. In this case, if the click rate of a video game advertisement for older people is higher than it is for younger people in a given user group, then we might suspect such user group about cheating.

To analyze whether a user group is a fraudster, simply comparing click rate between older people and younger people or males and females is not accurate enough. It is because there could be many other attributes influencing the click rate. The best way is to analyze the dependence of the user actions on the attributes, such as age or gender.

The definition of 'dependent' requires knowledge of rough set theory. In rough set theory, an index called dependency can express how strong one group of attributes is influenced by another. The dependency of click actions on attributes, such as age, gender, platform or even price and brand of user's device can show the user's action. If the dependency of two user groups on one same advertisement is different, then it is very likely that one of the groups is a fraudster.

The advantage of using dependency on mobile anti-fraud is that this method focuses on the action of groups of people. One of the most serious hardships in anti-fraud is that fraudsters change their ID and IP frequently,

which makes it hard for an anti-fraud process to track them down. However, as long as the new IDs and IPs are classified in the same group (e.g. all users of an APP), changing ID or IP is meaningless for our method.

Another advantage of this method is that even if fraudsters are aware of the method, it is still hard for them to avoid being detected. The best way to avoid being detected by this method is to make the dependency similar to the dependency of a real user group. Fraudsters, however, usually do not have enough real traffic, which means it is hard for them to get the essential data.

In this paper, we describe the common fraud methods in Section 2, and we explain the rough set theory in Section 3. In Section 4, a limitation of rough set theory is discussed together with its solution, namely fuzzy rough set theory. In Section 5, we present the algorithm that applies fuzzy rough set theory to detect fraudsters. In Section 6, we demonstrate the algorithm by using synthetic testing data.

2 Common Fraud Methods

There are many different methods of cheating for mobile advertisement fraudsters. These methods can be generally categorized into three classes.

- **False Users**

  In a class of false users reporting logs to the advertising server do not come from a real human user but are generated either automatically by a fraudster's server or manually by a smartphone device operated by a professional fraudster. Fraudsters who use a server to generate such cheating logs are called server-based fraudster. Fraudsters who use real devices to cheat are called real-device-based fraudster.

- **False Actions on Real Users**

  In this class reporting logs to the advertising server come from a real user but are created against the user's will. There are many different ways of achieving this, like leaving users no other option than clicking on an advertisement or sending click reports without real clicks.

- **Induced Real Actions on Real Users**

  Some APPs nowadays would encourage the users to click advertisements by offering a small amount of profit. To gain profit, users using such APPs may click on advertisements that they are not interested. Such actions should also be considered as fraud actions since they bring no benefit to the advertisers.

Though there are many differences between fraudsters, there are two things in common.

- The fraudsters intend to gain profit, which is different from normal users.
- A fraudster must run an APP to generate fake data. That means it is not necessary to identify which user is a fraudster, but to find suspicious APPs are good enough for anti-fraud.

Bots install is a fraud method that happens when fraudsters simulate real user behavior. This fraud method is hard to detect because its behavior not only include installation but also in-app behaviors, such as add something to the cart. Yao, *et al.* [10], and Zhu *et al.* [12] developed an ensemble model specially designed against the fraud method. The fraud detection model in [12] is a remarkable contribution to mobile anti-fraud as it dealts with a hard problem in the area.

In [2], Dou *et al.* introduced their understanding and detecting method of mobile APP download fraud. Using a centralized management system, Tarmazakov and Silnov [8] presented a fraud preventing method. Oluwagbemi [4], developed a neural network algorithm to predict mobile fraud actions. Pooranian *et al.* [5] introduced different fraud and anti-fraud methods. Tian *et al.* [9] described a specially designed anti-fraud method against crowd fraud (device-based fraud in this article) and Oentaryo *et al.* [3] used a data mining approach method to detect click fraud on online advertisements.

3 Basic Rough Set Theory

Rough set theory has been developed for decades. It has shown great value in describing uncertainty. Suraj [7] explained the rough set theory with simple language and easy-understanding examples. Zhang *et al.* [11] gave a survey about the development of rough set theory until 2016. Tsumoto and Shusaku [6] talked not only about the history but also possibilities in the future of rough set theory.

Of all the references, [1] is the most impressive one. Cornelis *et al.* [1] combined fuzzy set theory and rough set theory, which inspired us on this paper and our plan for future works.

3.1 Basic Concepts of Rough Set Theory

Approximation Space. Given a set of objects U and R a subset of U^2. Then U and R can be called the universe and an indiscernibility relation. If R is an equivalence relation, the pair (U, R) can be called an approximation space. In an approximation space (U, R), given any $x \in U$, we use $R(x)$ to denote the equivalence class determined by x.

Lower Approximation. Given an approximation space (U, R) and a subset X of U, the lower approximation of X with respect to R is

$$R_* (X) = \{x : R(x) \subseteq X\}$$

Upper Approximation. Given an approximation space (U, R) and a subset X of U, the upper

approximation of X with respect to R is

$$R^*(X) = \{x : R(x) \cap X \neq \emptyset\}$$

Accuracy of Approximation. Given an approximation space (U, R) and a subset X of U, the accuracy of approximation of X with respect to R is

$$\alpha_R(X) = \frac{|R_*(X)|}{|R^*(X)|}$$

Membership Degree. Given an approximation space (U, R) and an element x and a subset X of U, the membership degree of x on X is

$$\mu_X^R(x) = \frac{|R(x) \cap X|}{|R(x)|}$$

3.2 An Example of Lower and Upper Approximation

Suppose we have a universe U and its subset S (Figure 1).

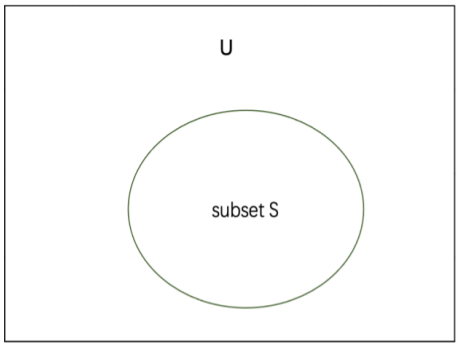

Figure 1: Universe and its subset S

Give an equivalence relationship R of U that divides U into several equivalence classes. These equivalence classes are the small squares in Figure 2.

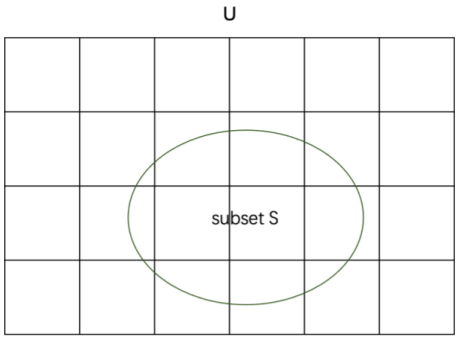

Figure 2: Equivalence classes and subset S

Thus, the green squares in Figure 3 are the lower approximation of subset S. The upper approximation of subset S are the shaded squares in Figure 3.

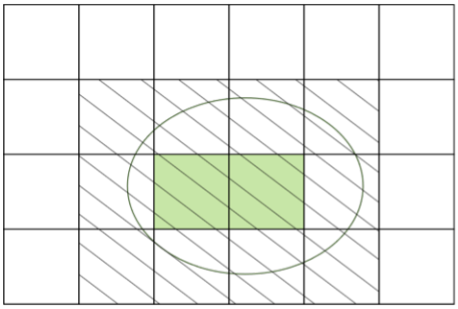

Figure 3: Lower and Upper approximation of subset S

3.3 Rough Set Theory and Information Systems (Relational Tables)

In the real world, data is usually stored in relational tables. To apply rough set theory to real-world data analysis, we need to define relational tables, namely information systems.

Information System: An information system S is a pair $S = (U, A)$, where U is a non-empty finite set of elements, and A is a non-empty finite set of attributes with the value map from U to the value of all attributes in A. U is usually called the universe.

This definition may be kind of abstract, but it should be understandable if assuming S be a relational table, U be the set of all rows in S and A to be all the columns.

To apply rough set theory to an information system, we need to build an approximation space structure on it first. Thus, the definition of indiscernibility in relation to information systems is required.

Indiscernibility Relation: Given an information system $S = (U, A)$ and B is a subset of A, the B-indiscernibility relation (written as $\mathrm{IND}_S(B)$) is defined as

$$IND_S(B) = (x,\ x') \in U^2 | \forall a \in B,\ a(x) = a(x')$$

where a(x) is the value of attribute a on element x.

As $\mathrm{IND}_S(B)$ is a subset of U^2, $\mathrm{IND}_S(B)$ suits the definition of indiscernibility relation. More importantly, it is obvious that $\mathrm{IND}_S(B)$ is an equivalence relation. Thus, we have successfully built an approximation space $(U, \mathrm{IND}_S(B))$ on the information system $S = (U, A)$.

Given an information system $S = (U, A)$ and a subset B of A, we define the following concepts.

Equivalent Class: For any x in U, the equivalent class x of B-indiscernibility relation is

$$[x]_B = \{x' \in U | (x, x') \in IND_S(B)\}$$

Lower Approximation:

$$B_*(X) = \{x | [x]_B \in X\}$$

Upper Approximation:

$$B^*(X) = \{x | [x]_B \cap X \notin \emptyset\}$$

Accuracy of Approximation:

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|}$$

Membership Degree:

$$\mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

3.4 Dependency of Attributes

The main goal of fraud detection is to discover 'unusual people'. Theoretically, the behavior of different user groups should be similar as long as the user groups are large enough. Thus, if the behavior between two user groups is quite different, at least one of the two user groups is 'unusual', which means very likely to be cheating.

This was only a theoretical method for anti-fraud in the past, as there was no index qualified for this job. In rough set theory, an index is suitable to express user behavior for fraud detection. This index is called the dependency of attributes.

Dependency of Attributes: Given an information system $S = (U, A)$, and C, D are subsets of A, the dependency of attributes C on attributes D in the universe of U is defined as

$$k_U(C, D) = \frac{|\cup_{X \in U/D} C_*(X)|}{|U|}$$

The dependency of attributes shows the accuracy when using one set of attributes to represent another set of attributes. In the real world, it could describe how strong one set of attributes can influence another set of attributes. If the dependency of D on C equals one, then the values of D are fully determined by the values of C. If the dependency equals zero, then C does not influence the value of D.

As mentioned, the behavior of different user groups is supposed to be similar. It is safe to assume that the influence of some attributes, like age or gender of users, on the behavior of users, like whether they like to click on an advertisement is stable in different user groups. With this assumption, we can use the dependency to detect fraudsters. If the dependency of a click event on a group of attributes is quite different in different user groups, then one of the user groups contains some fraudsters. This is the basic idea of this work.

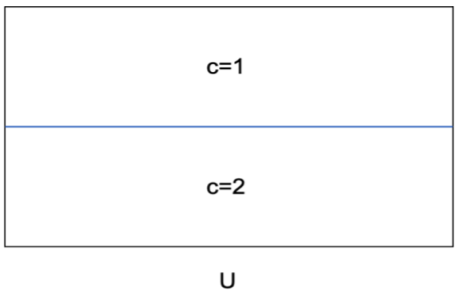

Figure 4: Attribute c on the universe

3.5 An Example of Dependency of Attributes

Given an information system $S = (U, A)$. Let c, d be elements of A.

As shown in Figure 4 the attribute c has two possible values 1 and 2.

The attribute d has three possible values 1, 2 and 3 (Figure 5).

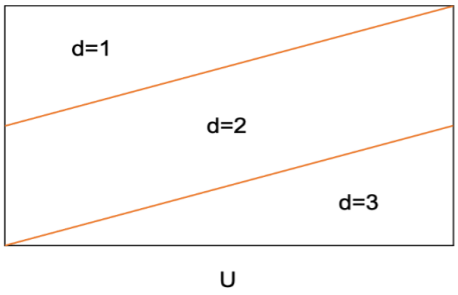

Figure 5: Attribute d on the universe

To put attribute c and d together (Figure 6).

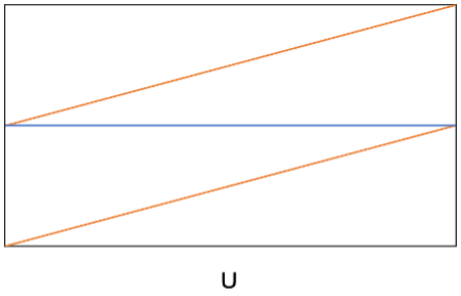

Figure 6: Attributes c and d in the universe

Let C = c and D = d. Then the dependency of C on D in U is the cardinality of sets in grey divided by the cardinality of universe in Figure 7.

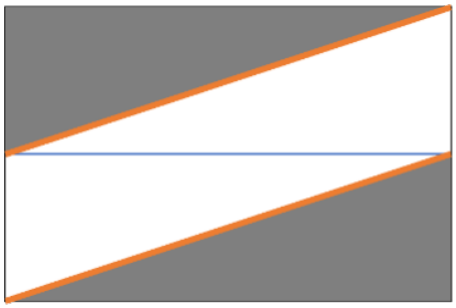

Figure 7: The dependency of c on d in the universe

4 Part Rough Set and Part Approximation

In the previous chapter, we introduced some contents of rough set theory and the definition of the dependency of attributes. We also introduced the brief idea of how to use dependency to detect fraudsters. However, the dependency of attributes cannot be used directly in big data analysis, as there is one serious problem.

The definition of the dependency used the concept of lower approximation. In the definition of lower approximation, an equivalence class is a part of the lower approximation only when it is a subset of the target set, which means that there can be no element outside of the target set. However, in big data analysis, since the data is too large, any possibility could happen. This could cause errors when calculating the dependency.

Suppose there is an advertisement for PC games. As it is a game advertisement, youngsters are more likely to click it while older people usually ignore it. In this case, the dependency of whether a user clicks the advertisement on the age of users should be high. If the data sample is large enough, there's a possibility that some youngsters did not click the advertisement and some elder people clicked the advertisement. In this case, the dependency could be zero only because a few people behave differently from other people. This is the limitation of the rough set theory.

An imaged example of this issue is in Figure 8.

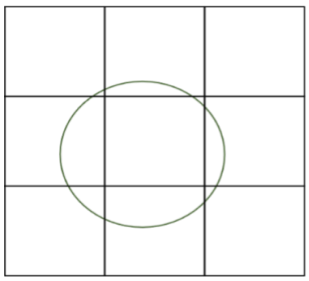

Figure 8: An imaged example of this issue

In Figure 8, suppose each small square in the picture is an equivalence class. Then the lower approximation of the circle is empty. However, the square in the middle is almost a subset of the circle. There is only a small part out of the circle. In this case, using the vanilla definition of lower approximation is not good enough to solve problems.

Thus, an improvement to the definitions is essential.

Part of Set: Given a set A and a real number $a \in [0,\ 1]$, then the pair (A, a) is called a fuzzy part of A.

The Cardinality of a part of set: Given a fuzzy part (A, a), then the cardinality of this fuzzy part is

$$|(A,\ a)| = |A| * a$$

Union of parts of sets: Given (A, a) and (B, b) to be two fuzzy parts. If $A \cap B = \emptyset$, then the union of the two fuzzy parts is

$$(A \cup B,\ \frac{|A| * a + |B| * b}{|A| + |B|})$$

Inner part: Given a set U and A, B to be its subsets. The inner part of B in A is

$$I_{(A,B)} = (B,\ \frac{|A \cap B|}{|B|})$$

where A is called the source-set and B is called the cut-set.

Outer part: Given a set U and A, B to be its subsets. The outer part of B in A is

$$O_{(A,B)} = (B, 1 -\ \frac{|A \cap B|}{|B|})$$

The subtraction of inner and outer parts: Given a set U and A, B to be its subsets. The subtraction of the inner and outer parts of B in A is

$$\begin{aligned} I_{(A,B)} - O_{(A,B)} &= (B, \max(0, \frac{|A \cap B|}{|B|} - (1 - \frac{|A \cap B|}{|B|}))) \\ &= (B, \max(0,\ \frac{2 * |A \cap B| - |B|}{|B|})) \end{aligned}$$

$$\begin{aligned} O_{(A,B)} - I_{(A,B)} &= (B, \max(0, 1 - \frac{|A \cap B|}{|B|} - \frac{|A \cap B|}{|B|})) \\ &= (B, \max(0,\ \frac{|B| - 2 * |A \cap B|}{|B|})) \end{aligned}$$

Part Lower Approximation: Given an approximation space (U, R) and a subset A of U, the fuzzy lower approximation of A with respect of R is a pair of sets defined below

$$\underline{R}\,(A) =\ U_{X \in \frac{U}{R}}(I_{(A,X)} - O_{(A,X)})$$

Part Upper Approximation: Given an approximation space (U, R) and a subset X of U, the fuzzy upper approximation of X with respect of R is a pair of sets defined as below

$$\bar{R}\,(A) =\ U_{X \in \frac{U}{R}}(O_{(A,X)} - I_{(A,X)})$$

Part Dependency of Attributes: Given an information system $S = (U, A)$ and C, D being subsets of A, the dependency of attributes C on attributes D in the universe of U is defined as

$$k\left(C, D\right) = \frac{\sum_{X \in \frac{U}{C}} \left(|\underline{D}(X)|\right)}{|U|}$$

We define part dependency in this way because for any $X \in \frac{U}{C}$ and $Y \in \frac{U}{D}$, if Y is a subset of X, then the two definitions are the same in the case of X and Y, but if Y is not a subset of X, the part dependency will ignore Y and X like the dependency in rough set theory. This would result in part dependency being more accurate, as it also considers the cases of most of Y is in X, but Y is not a subset of X.

4.1 Example of Part Rough Set Dependency

Given an information system $S = (U, A)$. Let c, d be elements of A. As shown in Figure 9 the attribute c has two possible values 1 and 2.

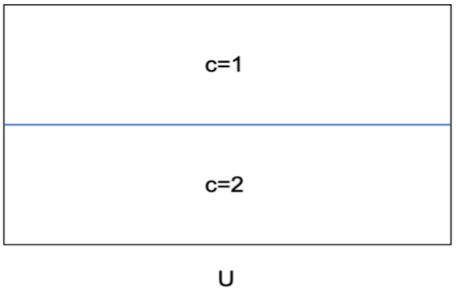

Figure 9: Attribute c in the universe

The attribute d has three possible values 1, 2 and 3 (Figure 10).

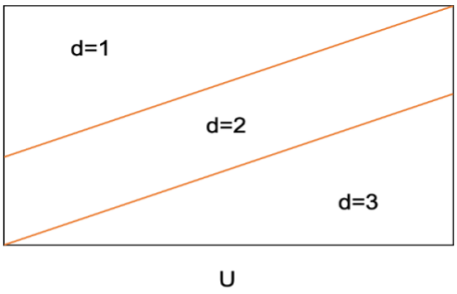

Figure 10: Attribute d in the universe

To put attributes c and d together (Figure 11).

Let C = c and D = d. Then the part dependency of c on d in U is the subsection of the cardinality of sets in green and sets in yellow divided by U in Figure 12, Figure 13 and Figure 14.

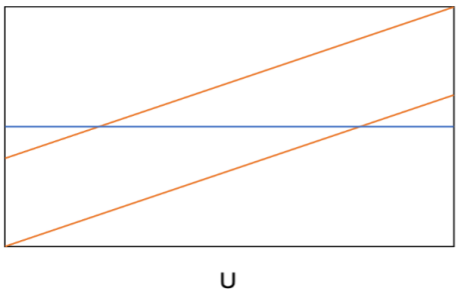

Figure 11: Attribute c and d in the universe

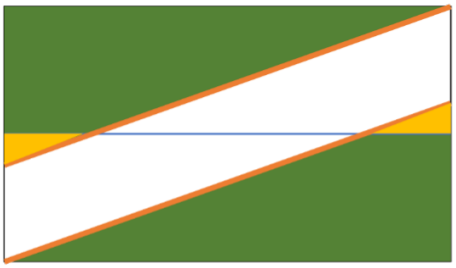

Figure 12: Part dependency (d=2)

Figure 13: Part dependency (d=1)

Figure 14: Part dependency (d=3)

5 Application of Part Rough Set Theory to Mobile Fraud Detection

As it is explained in Chapter 1, the mobile fraud actions are the actions that intend to gain profit. It is easy to see from this definition that the main difference between normal user action and fraud action is the intention. The normal user actions come from the user's interests. Such actions influenced by attributes of the user, like age or gender. On the contrary, the fraud actions are not dependent on user's attributes as normal user actions. In other words, the dependency of a click event on attributes, such as age, gender or location should be different in the universe of normal users and the universe of fraudsters.

To make the result more accurate, we will use a dependency metric for fraud detection.

Given a data set with data of user action of click or not and attributes of users of age, gender, platform and location. The dependency metric of user group A is in Table 1.

Table 1: Dependency metric

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	xxx	xxx	xxx	xxx	xxx
0002	xxx	xxx	xxx	xxx	xxx
0003	xxx	xxx	xxx	xxx	xxx

According to the definition of part dependency of attributes, two attribute sets C, D and the universe U all influence the dependency. In the dependency metric, D_xxx means the dependency of a click event on attribute xxx detailed information can be seen in Table 2. The first column ad ID represents the universe. For example, if an ad ID equals 0001, then the universe is of the user group A on advertisement 0001.

Table 2: Dependency metric of attributes

D_age	dependency of click rate on attribute age
D_gen	dependency of click rate on attribute gender
D_pla	dependency of click rate on attribute platform
D_loc	dependency of click rate on attribute location
D_main	dependency of click rate on all attributes above

6 Calculation of the Dependency Metric

Given a user group A and an ad ID, the process first enquiry the database for logs of the user group and advertisement. After obtaining the logs, the following algorithm can be used to compute the dependency metric (Figure 15).

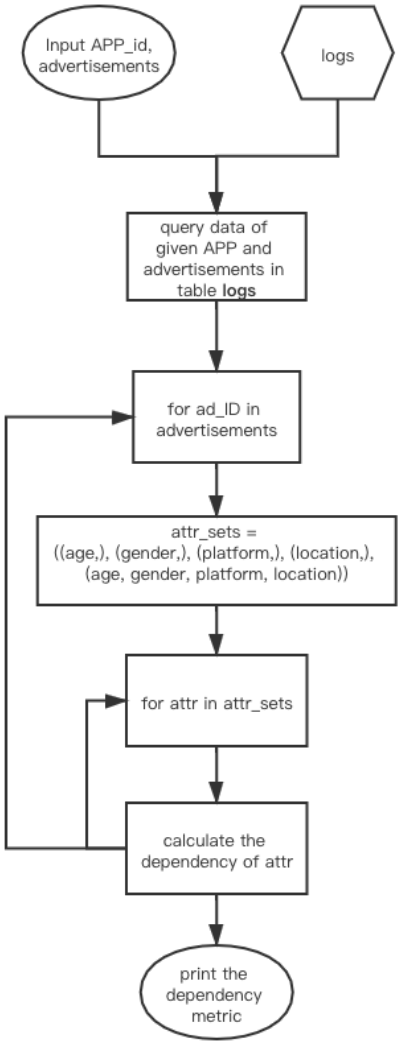

Figure 15: The process

To calculate the dependencies in the dependency metric, we use **Algorithm 1** Dependency Calculation.

When dealing with real data, number of logs with click=1 will be much smaller than logs with click=0. That would cause the dependency to be too small to study. The solution to this problem is easy: Put a higher weight on the logs with click=1 will fix it.

7 Analysis of Simulation Results

As mentioned in Chapter 2, identifying suspicious APPs will be enough for fraud detection. Thus, we designed five different APPs in our test. Some of them are normal, and others are fraudsters. Users in different APPs have different possibilities to 'click' on advertisements.

Algorithm 1 Dependency Calculation

input logs and attributes
for a in all possible values of given attributes **do**
c[a] = number of logs with attributes=a and click=1

nc[a] = number of logs with attributes=a and click=0

m[a] = abs(c[a] - nc[a])
end for
return dependency = sum(m)/(number of given logs)

In the real world, different advertisements have different targeted users. Different attributes also have different influences on whether targeted users will click or not. Such actions are not easy to be simulated by fraudsters. The sever based fraudster may identify the targeted users of an advertisement, but it is almost impossible for fraudsters to know the difference between different attributes.

To simulate this phenomenon, we designed a series of columns for advertisements called the power of attributes. There is a column for each attribute to describe how strong the attribute can influence the possibility of normal users clicking the advertisement. We designed three advertisements for the simulation. The values of power of attributes for each advertisement are listed in Table 3.

Table 3: Advertisements

ad ID	age	gender	platform	location
0001	10	10	1	1
0002	10	10	10	10
0003	1	1	1	1

To verify our fraud detecting process, we designed five different APPs to represent normal APPs or fraudsters. Related information on each APP and the test results are in Table 2.

APP1 is a normal APP. Users in APP1 are more likely to click if they are targeted users of an advertisement. Also, different attributes have different powers to influence whether targeted users click or not. These rules are both applied in APP1.

The dependency metric of APP1 is in Table 4.

Table 4: Dependency Metric of APP1

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	0.388	0.324	0.042	0.055	0.435
0002	0.331	0.345	0.316	0.390	0.560
0003	0.056	0.058	0.027	0.053	0.141

APP2 is a sever based fraudster. The users in APP2 are false users. Their actions are different from normal users like APP1. Targeted users in APP2 are more likely to click, but there's no difference in the power of different attributes.

The dependency metric of APP2 is in Table 5.

Table 5: Dependency Metric of APP2

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	0.212	0.241	0.207	0.204	0.373
0002	0.213	0.226	0.237	0.190	0.375
0003	0.246	0.218	0.218	0.177	0.368

APP3 is a real-device based fraudster. All cheating actions operated by real humans. The actions of targeted users in APP3 are the same as other users.

The dependency metric of APP3 is in Table 6.

Table 6: Dependency Metric of APP3

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	0.021	0.012	0.018	0.018	0.057
0002	0.006	0.001	0.010	0.010	0.042
0003	0.013	0.001	0.009	0.009	0.047

APP4 is a user-inducing APP. Half of the users in APP4 behave like users in APP1, and the other half behave like users in APP3.

The dependency metric of APP4 is in Table 7.

Table 7: Dependency Metric of APP4

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	0.065	0.047	0.006	0.003	0.095
0002	0.052	0.064	0.074	0.055	0.113
0003	0.007	0.006	0.007	0.013	0.063

APP5 is also a normal APP. All the behaviors are set to be the same as APP1.

The dependency metric of APP5 is in Table 8.

According to the test results, APP1 and APP5 are similar. It suits our assumption that APP1 and APP5 are both normal APPs.

In the result of APP2, as APP2 is a server-based fraudster, it can identify the target users for each app. Thus the dependency of each attribute is not as small as APP3 or APP4. But as APP2 is not aware of the power of each attribute, different attributes have little difference in the result. APP3 and APP4 are similar, as they both contain real-device based fraudster, the influence of attributes are so small that the dependencies are all nearly zero.

Thus, the test result shows that our process has the potential ability to detect fraudsters. The difference between the power of different attributes and advertisements are too idealized. Whether one can find advertisements as good as they are in the simulation is still a question. In

Table 8: Dependency Metric of APP5

ad ID	D_age	D_gen	D_pla	D_loc	D_main
0001	0.338	0.339	0.039	0.062	0.404
0002	0.344	0.340	0.348	0.336	0.562
0003	0.017	0.073	0.066	0.057	0.149

conclusion, the algorithm is useful in mobile anti-fraud, but it is not a practical method yet. To be applied in the real world, there are more work and tests to be done.

8 Conclusion and Future Work

In this paper, we developed an anti-fraud method with the application of rough set theory. The method calculates and compares the part dependency metric of different user groups on different advertisements. By testing different user groups with different advertisements, the dependency metric can identify fraudsters without worrying about the differences between cheating methods. There is also another advantage of this method. As the main job of this method is to compare the dependency metric, it would be impossible for fraudsters to pretend to be a normal user group as long as they do not have the dependency metric data of real user groups. Since fraudsters usually have little real traffic, it is safe to assume that the fraudsters will not be able to gain the data. Besides, anti-fraud programmers can always use new advertisements for the dependency metric, which means the anti-fraud programmers will become the initiative with the help of the method in this paper.

Despite the fact that the method has significant advantages, it is still not practical enough for real application. When applied to the real world, we are not sure if the dependency metrics of different user groups is going to be as different as they are in the test. If the difference between dependency metrics is small, it won't be strong enough for fraud detection. In that case, adjustments like using advertisements with higher dependency or adding more attributes to the dependency metric may fix the problem.

In conclusion, the anti-fraud method in this paper has a revolutionary potential, as it not only detects fraudsters without worrying about their methods, but also helps anti-fraud programmers get an initiative position against fraudsters, but it needs big real data for testing and modification to suit the real-world application.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] C. Cornelis, M. De Cock, and A. M. Radzikowska, "Fuzzy rough sets: from theory into practice," in *Handbook of Granular computing*, pp. 533–552, 2008.

[2] Y. Dou, W. Li, Z. Liu, Z. Dong, J. Luo, and S. Y. Philip, "Uncovering download fraud activities in mobile app markets," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'19)*, pp. 671–678, 2019.

[3] R. Oentaryo, E. P. Lim, M. Finegold, D. Lo, F. Zhu, C. Phua, E. Y. Cheu, G. E. Yap, K. Sim, and M. N. Nguyen, "Detecting click fraud in online advertising: a data mining approach," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 99–140, 2014.

[4] O. Oluwagbemi, "Predicting fraud in mobile phone usage using artificial neural networks," *Journal of Applied Sciences Research*, vol. 4, no. 6, pp. 707–715, 2008.

[5] Z. Pooranian, M. Conti, H. Haddadi, and R. Tafazolli, "Online advertising security: Issues, taxonomy, and future directions," *IEEE Communications Surveys & Tutorials*, 2021.

[6] A. Skowron and S. Dutta, "Rough sets: past, present, and future," *Natural Computing*, vol. 17, no. 4, pp. 855–876, 2018.

[7] Z. Suraj, "An introduction to rough set theory and its applications," in *ICENCO*, Cairo, Egypt, vol. 3, p. 80, 2004.

[8] E. I. Tarmazakov and D. S. Silnov, "Modern approaches to prevent fraud in mobile communications networks," in *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus'18)*, pp. 379–381, 2018.

[9] T. Tian, J. Zhu, F. Xia, X. Zhuang, and T. Zhang, "Crowd fraud detection in internet advertising," in *Proceedings of the 24th International Conference on World Wide Web*, Conference Proceedings, pp. 1100–1110.

[10] T. Yao, Q. Li, S. Liang, and Y. Zhu, "Botspot: A hybrid learning framework to uncover bot install fraud in mobile advertising," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2901–2908.

[11] Q. Zhang, Q. Xie, and G. Wang, "A survey on rough set theory and its applications," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 4, pp. 323–333, 2016.

[12] Y. Zhu, X. Wang, Q. Li, T. Yao, and S. Liang, "Botspot++: A hierarchical deep ensemble model for bots install fraud detection in mobile advertising," *ACM Transactions on Information Systems*, vol. 40, no. 3, pp. 1–28, 2021.

Biography

Jinming Ma, Received his Master degree from School of Computing and Information Technology, University

of Wollongong, Australia. His research interest include data processing and mobile anti-fraud.

Tianbing Xia, Received his PhD degree from the Department of Computer Science, University of Wollongong, Australia. He is currently work at the School of Computing and Information Technology, University of Wollongong Australia. His research interests include combinatorial design, cyber security and data processing.

Janusz Getta, received his Master of Computer Science and PhD degrees from Warsaw University of Technology in Poland. Now, he is with the School of Computing and Information Technology, University of Wollongong in Australia. His research interests include database systems, big data systems, and performance aspects of data processing.