IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON MOBILE MULTIMEDIA: METHODOLOGY AND APPLICATIONS

Received November 20, 2019, accepted December 14, 2019, date of publication December 25, 2019,
date of current version January 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962134

Privacy Protection: An Anti-Fraud Scheme Based on Improved Bayesian Game Model in Multimedia Communication System

RUI ZHANG[1,2], HUI XIA[2], (Member, IEEE), FEI CHEN[1], LI LI[2], (Student Member, IEEE), AND XIANG-GUO CHENG[1]

[1]College of Computer Science and Technology, Qingdao University, Qingdao 266000, China
[2]College of Information Science and Engineering, Ocean University of China, Qingdao 266000, China

Corresponding authors: Hui Xia (xiahui@qdu.edu.cn) and Xiang-Guo Cheng (15964252399@163.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872205, in part by the Shandong Provincial Natural Science Foundation under Grant ZR2019MF018, in part by the Project of Shandong Province Higher Educational Science and Technology Program under Grant J16LN06, in part by the Source Innovation Program of Qingdao under Grant 18-2-2-56-jch, and in part by the Open Research Fund from Shandong Provincial Key Laboratory of Computer Networks under Grant SDKLCN-2018-07.

ABSTRACT In the multimedia communication system, users are often exposed to fraud information from malicious applications (such as providing fake bids and false content), which can easily cause privacy leakage. And multimedia information pluralism make it more difficult to protect users' privacy. To solve these problems, this article proposed an anti-fraud scheme based on improved Bayesian game model. First, we designed a Bargain-bayesian game model for modeling the interaction between applications and regular users. We used the Two-round bargain method to establish payoff matrix for two players and adjust it through regular user's detection rate dynamically. Then we obtained the application's best bid in the first round of bargain by backward induction to prevent malicious applications from sending fake bids. Second, we customized a group of test users and developed another interaction model between applications and users (i.e., regular and test) as the Two-side Bayesian game model based on sliding adaptive logistic regression method, then we put three influencing factors into the payoff matrix: test user's ratio, malicious application's ratio, and regular user's ratio. Through Bayesian Nash Equilibria analysis, we obtained values of three influencing factors when malicious applications provide true content, and thus solved the problem of malicious applications providing false content. Finally, experiment results proved that the new scheme had effectively raised expected payoffs for both players and their transaction achievement rate, and lowered the probability of users being deceived by malicious applications, which successfully solved the issue of users' privacy disclosure.

INDEX TERMS Multimedia communication system, privacy protection, Bayesian game, logistic regression.

I. INTRODUCTION

Quick development of technology such as mobile communication and intelligence terminal accelerated the information dissemination process, which managed to satisfy users' need for fast and convenient access to multimedia information. By the end of 2018, the unit number that is connected to the Global Mobile Internet surpassed 22 billion. And due to its openness, dynamic, and sharing features, multimedia information is prone to be attacked by eavesdropping, interception, and tampering, which has increasingly become a serious issue. Mobile terminal equipment is widely used nowadays, and intelligence terminal applications have been increasing rapidly. But related security policy and the formulation and execution of law falls behind, which causes a massive outbreak of malicious applications, seriously threatening internet users' privacy and data safety. For example, privacy incidents occurred for many times on Facebook. Google tried to adjust its strategy of analysis on sensitive data to prevent users' information disclosure.

Currently, in the multimedia communication system, researches on privacy protection mainly focused on

The associate editor coordinating the review of this manuscript and approving it for publication was Qing Yang.

multimedia copyright protection, recognition of malicious information, tamper-proof and anti-theft of privacy, etc. Multimedia information copyright protection primarily relies on information hiding, digital watermarking, splicing detection, and perceptual hash, etc. In [1], [2], two kinds of coverless information hiding methods were introduced. In [3], [4], researchers first analyzed the design theory of spread spectrum watermark, and then proposed a watermarking algorithm based on the transform domain. In [5], [6], the method of detecting scale and the trace of re-sampling were introduced to protect the copyright of multimedia information. In [7], [8], researchers first introduced a privacy protocol for perceptual image hashing, and then proposed a image perceptual hashing schemes. However, the above proposals could not protect copyright effectively. The cognition of malicious information mainly depends on designing a model of cognition in the multimedia information system. In [9], the Distance-Based Discrimination Detector was proposed to distinguish malicious information. In [10], researchers identified the malicious edge device by using the Markov model, Intrusion Detection System, and Virtual Honeypot Device in the fog computing environment. In [11], a small target detection scheme based on the weighted local difference measure was proposed for the detection of small targets. Moreover, a new attack-defense game model base on the repeated game approach was proposed for detecting malicious nodes in [12],. However, the model of cognition is too complicated and errors can easily occur in the above methods. The tamper-proof and anti-theft for multimedia information mainly rely on tamper detection or adjusting attackers and defenders' payoff. In [13], researchers summarized the typical image tamper types, published image tamper data sets, and recent tamper detection methods. In [14], the location privacy protection scheme based on the routing protocol was proposed to prevent information leakage. In [15], researchers proposed an oblivious watermarking technique to detect the tampering of digital images. In [16], a novel tamper detection, localization, and recovery scheme were proposed based on ***DWT*** and ***CS***. In [17], a passive blind video similar tamper detection algorithm based on multi-scale normalized mutual information is proposed, which realizes video frame replication, frame insertion and frame deletion tamper detection. Meanwhile, some researchers apply game theory to prevent information from being tampered or theft in [18]–[20]. In [21], the interaction between attacker and defender was model as a non-cooperative security game, players adjust their strategy by analyzing and predicting the opponent's strategy. In [22], researchers first studied a mixed strategy game of subjective storage defense, and then proposed a *APT* defense scheme based on Q-learning. In [23], the optimal transmission strategy and interference strategy of the legal transmission and eavesdropping were determined by analyzing the equilibrium of Stackelberg game. Moreover, there are also some emerging security technologies to protect users' privacy. For example, the privacy preservation framework of CPSSs [24], sensitive information protection based on differential privacy [25], and Hiding participants' abnormal behaviors [26]. However, errors could occur easily during the tampered area detection process, and the attacking and defending participants have a lower payoff in the above proposals.

How to solve the issue of malicious applications providing fake bids and false content to deceive users which caused the problem of internet users' privacy leakage problem is the biggest challenge we are facing nowadays. Though most of the researchers mentioned about users' privacy protection, they did not analyze from the perspective of preventing malicious applications from deceiving users. Therefore, this article proposed an Anti-fraud privacy protection scheme based on an improved Bayesian game model. Our main contributions are as follows:

(1) To address the issue of malicious applications sending fake bids, we first proposed the ***B***argain-***b***ayesian game model, i.e., ***BB***, for modeling the interaction between applications and regular users. Second, we used Two-round bargain method to establish the payoff matrix for both players and then formed dynamic regulation of this payoff matrix through regular user's detection rate. Third, we obtained the application's best bid in the first round of bargain by backward induction, thus prevented malicious applications from sending fake bids. Moreover, through Bargain-bayesian Nash Equilibria analysis, we found out that by raising regular user's detection rate, we can solve the problem of malicious applications providing false content to a certain degree.

(2) To address the issue of malicious applications providing false content, we first customized a group of test users and developed the ***T***wo-side ***B***ayesian game model based on sliding adaptive logistic regression method, i.e., ***TB***, to model the interaction between applications (i.e., normal and malicious) and users (i.e., regular and test). Second, we constructed a sliding adaptive logistic regression method to calculate the probability of malicious applications providing false content and then put this probability as malicious application's ratio. Third, we adjusted the payoff matrix by the following three influencing factors dynamically: test user's ratio, malicious application's ratio, and regular user's detection rate. Through Two-side Bayesian Nash Equilibria analysis, we obtained the value range of three influencing factors when malicious applications chose to provide true content as its best strategy.

(3) The experiment used seven contrast vectors, including the expected payoff of malicious applications and regular users, the transaction achievement rate and so on, to compare and analyze the our new scheme and other schemes. The experiment results proved that in the ***BB*** model, the application's best bid price in the first round of bargain could raise payoffs for both players effectively. It also encouraged malicious applications to send real bids, which successfully solved the problem of malicious applications providing fake bids. In the ***TB*** model, by setting a smaller ratio of test users, the ratio of malicious applications providing true content increased remarkably, which managed

TABLE 1. Two-round bargain method.

Bargain round	Bid		Payoff	
	Application	User	Application	User
1	P_a^0		$P_a^0 - C_a$	$P_u^a - P_a^0$
		P_a^1	$\sigma(P_a^1 - C_a)$	$\sigma(P_u^a - P_a^1)$
2	P_a^2		$\sigma^2(P_a^2 - C_a)$	$\sigma^2(P_u^a - P_a^2)$
		P_a^3	$\sigma^3(P_a^3 - C_a)$	$\sigma^3(P_u^a - P_a^2)$

to prevent malicious applications from providing false content.

II. BARGAIN-BAYESIAN GAME MODEL (BB)

We build an interaction model between applications (malicious and normal) and regular users as Bargain-bayesian game model to solve the problem of malicious applications providing fake bids.

A. TWO-ROUND BARGAIN METHOD

To prevent malicious applications from providing fake bids, we use the Two-round bargain to determine the transaction price for two players and define σ as the discount coefficient of each bargain. Thus, the more rounds of the bargain between two players, the more the loss of payoff. The detailed bargain process can be described as follows:

In the first round, the application i makes its first bid of P_a^0, meanwhile, the payoff of the application i and the user j are $P_a^0 - C_a$ and $P_u^a - P_a^0$, where P_a^i, $i = 0, 1, 2, 3$ is the bid of each round bargain, C_a is the application i's the cost of providing content, P_u^a is the user's payoff of reaching agreement in the bargain. If the user j accepts it, and then the bargain ends with making a deal; otherwise, the user j makes its first bid of P_a^1, and then the payoff of two players are $\sigma(P_a^1 - C_a)$ and $\sigma(P_u^a - P_a^1)$. The bargain ends if the application i accepts P_a^1; otherwise, it enters the second round.

In the second round, the user j makes its second bid of P_a^2, the payoff of two players are $\sigma^2(P_a^2 - C_a)$ and $\sigma^2(P_u^a - P_a^2)$. If the application i accepts P_a^2, the bargain ends; otherwise, the user j makes its second bid price of P_a^3, the application i must accept it, and then the bargain ends. The Two-round bargain method is shown in Table 1.

We set $P_u^a > C_a$, $\frac{C_a}{P_u^a} < \sigma < 1 - \frac{C_a}{P_u^a}$, and use the backward induction to deduce the application i's optimal bid in the first round of bargain.

In the second round, since the application i must accept the user j's second bid of P_a^3, the user j must send the bid of 0, e.i., $P_a^3 = 0$ in its second bid. we define P_a^3 as the user i's general bid and $P_a^3 \neq 0$. Actually, two players are reluctant to send the second bid due to the decay of payoff. So, the user j accepts the bid as long as the application i's second bid makes the user j's payoff is not less than its payoff in the second bid, that is, the application i's second bid should be satisfying (1):

$$\sigma^2(P_u^a - P_a^2) \geq \sigma^3(P_u^a - P_a^3) \tag{1}$$

The maximum of P_a^2 satisfying (1) is $P_a^2 = (1-\sigma)P_u^a + \sigma P_a^3$. Then, the payoff of the application i and users j are:

$$\sigma^2(P_a^2 - C_a) = \sigma^2((1-\sigma)P_u^a + \sigma P_a^3 - C_a) \tag{2}$$

$$\sigma^2(P_u^a - P_a^2) = \sigma^3(P_u^a - P_a^3) \tag{3}$$

From (2), at this time, the application i's payoff is higher than that in the user j's second bid. From (3), the user j's payoff is equal to that in its second bid. Therefore, the bargain ends if $P_a^2 = (1-\sigma)P_u^a + \sigma P_a^3$ in the application i's second bid.

So similarly, we get the bid of the application i and the user j in the first round. The minimum of first bid P_a^1 for user j is $P_a^1 = (\sigma - \sigma^2)P_u^a + \sigma^2 P_a^3 + (1-\sigma)C_a$, meanwhile, the payoff of the application i and user j are $\sigma(P_a^1 - C_a) = \sigma((\sigma - \sigma^2)P_u^a + \sigma^2 P_a^3 - \sigma C_a)$ and $\sigma(P_u^a - P_a^1) = \sigma(P_u^a - (\sigma - \sigma^2)P_u^a - \sigma^2 P_a^3 - (1-\sigma)C_a)$. The maximum of first bid P_a^0 for the application i is $P_a^0 = (1+\sigma^2-\sigma^3-\sigma)P_u^a + \sigma^3 P_a^3 + (\sigma - \sigma^2)C_a$, meanwhile, the payoff of the application i and user j are $P_a^0 - C_a = (1+\sigma^2-\sigma^3-\sigma)P_u^a + \sigma^3 P_a^3 + (\sigma - \sigma^2 - 1)C_a$ and $P_u^a - P_a^0 = P_u^a - (1+\sigma^2-\sigma^3-\sigma)P_u^a - \sigma^3 P_a^3 - (\sigma - \sigma^2)C_a$. Therefore, we obtain the application's best bid in the first round of bargain, and thus solve the problem of malicious applications sending fake bids.

B. BARGAIN-BAYESIAN GAME ANALYSIS

In actual operations, there are two types of applications, i.e., malicious applications and normal applications, and one type of users, i.e., regular users. Malicious application has two strategies: False strategy, i.e., F, means that malicious applications are providing false content to users; True strategy, i.e., T, means that malicious applications are providing true content to users. Normal applications only have one true strategy, i.e., T. Regular users have two strategies: Accept strategy, i.e., A, which refers to regular users accepting the content provided by the application; Refuse strategy, i.e., R which refers to regular users not accepting content provided by applications. Regular users can make their judgments based on their experience to decide whether the content is true or not. We define notations in the payoff matrix are as follows:

C_a^m: the malicious application's cost of playing F;

C_a^n: the malicious application's cost of playing T;

P_a^0: the application's payoff after the successful bargain, it also refers to the user's transaction cost, i.e., $P_a^0 = (1+\sigma^2-\sigma^3-\sigma)P_u^a + \sigma^3 P_a^3 + (\sigma - \sigma^2)C_a$;

P_u^a: the regular user's payoff after the successful bargain

P_e: the malicious application's abnormal payoff by deceiving users. We assume $P_e > C_a^m > C_a^n$; otherwise, the malicious application does not have sufficient motivation to provide false content to deveive the user, and the regular user does not have sufficient motivation to distinguish the authenticity of the content;

α: the probability of the regular user identifying false content, where $\alpha \in [0, 1]$.

The payoff matrix of malicious applications and regular users when they play different strategies, as shown in Table 2.

TABLE 2. The payoff matrix of malicious applications and regular users.

Payoff	*R*	*A*
T	$(\alpha-1)C_a^n$, 0	$\alpha(P_a^0-C_a^n)$, $\alpha(P_u^a-P_a^0)$
F	$-\alpha(C_a^m+P_e)$, αP_e	$(1-\alpha)(P_a^0+P_e-C_a^m)$, $(\alpha-1)(P_a^0+P_e)$

TABLE 3. The payoff matrix of normal applications and regular users.

Payoff	*R*	*A*
T	$(\alpha-1)C_a^n$, 0	$\alpha(P_a^0-C_a^n)$, $\alpha(P_u^a-P_a^0)$

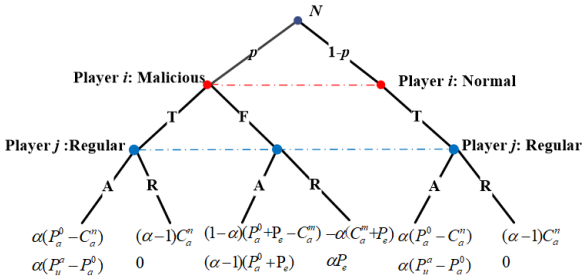


FIGURE 1. The extensive form of bargain-bayesian game.

When malicious applications play T, and regular users play A only when making the right decision. The malicious application with the probability of α gets normal payoff P_a^0 at the cost of C_a^n; meanwhile, the regular user with the probability of α obtains the normal payoff P_u^a at the cost P_a^0. When the malicious application plays F, and the regular user plays A when they misjudge the situation. The malicious application can successfully deceie users with the probability of $(1-\alpha)$, gets payoff P_a^0 and P_e at the cost of C_a^m for providing false content. Meanwhile, the regular user with the probability of $(1-\alpha)$ pays extra losses P_e at the cost P_a^0. This theory also applies to the payoff of regular users play R and malicious application play T and F. The payoff matrix of normal applications and regular users, as shown in Table 3, which follows the same theory as Table 2.

C. BARGAIN-BAYESIAN NASH EQUILIBRIUM ANALYSIS

Based on the type of application i and strategies of two players, we can derive an extensive form of Bargain-bayesian game, as shown in Fig. 1. On top of the tree, the root node N represents "Nature". p is the probability that the application's type selected by "Nature" is malicious, where $p \in [0, 1]$. The first layer shows the application's type, the second layer describes two types of application's strategy, the third layer represents the regular user's strategy, and the tuple of each branch terminal refers to payoffs of both players.

The application i is to play F if it is malicious, the application i is to play T if it is normal. From Fig.1, the expected payoff of the regular user j playing R and playing A are:

$$\begin{aligned} E_{\mu_u^R} &= p\alpha P_e \\ E_{\mu_u^A} &= p(\alpha-1)(P_a^0+P_e)+(1-p)\alpha(P_u^a-P_a^0) \end{aligned} \quad (4)$$

From (4), We can derive that if $0<\alpha<\frac{1}{2}$, $P_a^0>\frac{P_e+\alpha P_u^a}{2\alpha-1}$ and $\frac{\alpha(P_a^0-P_u^a)}{(2\alpha-1)P_a^0-P_e-\alpha P_u^a}<p<1$; or, $\frac{1}{2}<\alpha<1$, $P_a^0<\frac{P_e+\alpha P_u^a}{2\alpha-1}$, and $\frac{\alpha(P_a^0-P_u^a)}{(2\alpha-1)P_a^0-P_e-\alpha P_u^a}<p<1$, that is, $E_{\mu_u^R}>E_{\mu_u^A}$, which means the regular user j's optimal strategy is to play R. Meanwhile, the expected payoff of the malicious application i playing F and playing T are:

$$\begin{aligned} E_{\mu_a^F} &= -\alpha(C_a^m+P_e) \\ E_{\mu_a^T} &= (\alpha-1)C_a^n \end{aligned} \quad (5)$$

From (5), we can derive that if $0<\alpha<\frac{C_a^n}{C_a^n+C_a^m+P_e}$, then, $E_{\mu_a^F}>E_{\mu_a^T}$ which means the malicious application j's optimal strategy is to play F. And since $\frac{C_a^n}{C_a^n+C_a^m+P_e}<\frac{1}{2}$, the strategy profile ((Malicious application F, Normal application T), R, p) is a Bargain-bayesian Nash Equilibrium if $0<\alpha<\frac{C_a^n}{C_a^n+C_a^m+P_e}$, $P_a^0>\frac{P_e+\alpha P_u^a}{2\alpha-1}$, and $\frac{\alpha(P_a^0-P_u^a)}{(2\alpha-1)P_a^0-P_e-\alpha P_u^a}<p<1$. However, if $\frac{C_a^n}{C_a^n+C_a^m+P_e}<\alpha<1$, the malicious application j's optimal strategy is to play T, it will shift his strategy is to play T, therefore, the strategy profile ((Malicious application F, Normal application T), R, p) is not a Bargain-bayesian Nash Equilibrium.

So similar, we can derive that if the strategy profile ((Malicious application F, Normal application T), A, p) is a Bargain-bayesian Nash Equilibrium, there are three cases: *i)* $\frac{1}{2}<\alpha<\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$, $P_a^0<\frac{P_e+\alpha P_u^a}{2\alpha-1}$, and $0<p<\frac{\alpha(P_a^0-P_u^a)}{(2\alpha-1)P_a^0-P_e-\alpha P_u^a}$; *ii)* $\frac{1}{2}<\alpha<\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$, $P_a^0>\frac{P_e+\alpha P_u^a}{2\alpha-1}$, and $0<p<1$; *iii)* $0<\alpha<\frac{1}{2}$, $P_a^0>\frac{P_e+\alpha P_u^a}{2\alpha-1}$ and $0<p<\frac{\alpha(P_a^0-P_u^a)}{(2\alpha-1)P_a^0-P_e-\alpha P_u^a}$. The malicious application i is to play T if it is malicious, the normal application i is to play T if it is normal, the strategy profile ((Malicious application T, Normal application T), A, p) is a Bargain-bayesian Nash Equilibrium if $\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}<\alpha<1$ and $0<p<1$.

To sum up, it is possible to prevent malicious applications from sending fake bids by using Two-round bargain, and encourage malicious applications to play T by setting the regular user's detection rate $\alpha \in (\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}, 1]$. However, in actual operations, the regular user's detection rate is usually bellowed $\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$, and the regular user only distinguishes the authenticity of the content that provided by the malicious application based on historical experience, which is not accurate enough. Thus, we customize a group of test users to solve the problem of malicious applications providing false content.

III. TWO-SIDE BAYESIAN GAME MODEL (TB)

We build an interaction model between applications (i.e., malicious and normal) and users (i.e., test and regular) as

TABLE 4. The payoff matrix of malicious applications and test users.

Payoff	A
T	$P_a^0 - C_a^n$, $P_u^a - P_a^0 - C_u$
F	$P_a^0 - P_u^e - C_a^m$, $P_u^e - P_a^0 - C_u$

TABLE 5. The payoff matrix of normal applications and test users.

Payoff	A
T	$P_a^0 - C_a^n$, $P_u^a - P_a^0 - C_u$

Two-side Bayesian game model based on sliding adaptive logistic regression method to solve the problem of malicious applications providing false content.

A. TWO-SIDE BAYESIAN GAME ANALYSIS

Since we customize a group of test users, there are two types of applications and users in the game. The definition of applications and regular users are the same as in Section II, the test user is only to play A and it can simultaneously send the same content request to multiple applications and compare the received multiple pieces of content to distinguish the authenticity of the content. Thus, test users have higher accuracy in distinguishing the authenticity of the content.

The payoff matrix of the malicious application and the regular user, as well as the normal application and the regular user, as shown in Table 2 and Table 3. Table 4 and Table 5 show the payoff matrix of applications (i.e., malicious and normal) and test uses. We define new notations are as follows:

u: the probability that the user is test, that is, the test user's ratio in total users, where $u \in [0, 1]$;

C_u: the test user's cost of distinguishing the authenticity of the content;

P_u^e: the test user's payoff when it identifies a malicious application, we assume $P_u^e \geq C_u$; otherwise, the regular user does not have sufficient motivation to shift to test user for distinguishing the authenticity of the content.

Since the test user has only one strategy (i.e., A), we need only to analyze the payoff of the test user plays A, as shown in Table 4. If the malicious application plays F and the test user plays A. The malicious application gets the payoff P_a^0 at the cost C_a^m and P_u^e; the test user gains the payoff P_u^e at the cost C_u and P_a^0. If the malicious application plays T and the test user plays A. The malicious application obtains payoff P_a^0 at the cost C_a^n; the test user gains the payoff P_u^a at the cost C_u and P_a^0. The payoff analysis of the normal application and the test user follow the same theory as Table 4.

B. PREDICTING THE RATIO OF MALICIOUS APPLICATIONS

It is necessary to determine the malicious application's ratio in the system before analyzing the Two-side Bayesian Nash Equilibrium. However, which is extremely difficult in real operations. For this, we define the probability that the malicious application plays F as the malicious application's ratio, and the probability calculation belongs to the binary classification problem which is suitable for solving by a logical regression method. In this article, we use the previous two payoff of malicious applications in iterations to construct a sliding adaptive logistic regression method for predicting the probability that malicious application playing F. To the best of our knowledge, the larger size of training data set, the higher accuracy of the prediction, which means that the data set needs to cover large-scale historical data of all the time series. To solve this problem, Algorithm 1 gives the specific implementation method of the adaptive prediction window.

Algorithm 1 Sliding Window Size

```
function WindowSize()
WindowSize=1000
F1=2*precision*recall/(precision+recall)
if F1>0.9 then
WindowSize= WindowSize-50
else if F1<0.8 then
WindowSize= WindowSize+50
else
WindowSize= WindowSize
end if
return WindowSize
end function
```

If the accuracy of the method is high, decreasing the size of the prediction window; otherwise, increasing the size of the prediction window. For the data set length t in iterations, if $t \leq Windowsize$, the new data goes directly to the sliding window; otherwise, we remove the leftmost unit data of the sliding window, and the new data enters the sliding window to form a new sliding window data set.

C. TWO-SIDE BAYESIAN GAME MODEL BASED ON SLIDING ADAPTIVE LOGISTIC REGRESSION METHOD NASH EQUILIBRIUM ANALYSIS

Based on the type and strategy of applications and users, we derive an extensive form of Two-side Bayesian game model based on sliding adaptive logistic regression method, as shown in Fig. 2.

The application i is to play F if it is malicious, the application i is to play T if it is normal, the user j is to play A if it is test. From Fig. 2, the expected payoff of regular users playing R and playing A are:

$$\begin{aligned} E_{\mu_u^R} &= p\alpha P_e \\ E_{\mu_u^A} &= p(\alpha - 1)(P_a^0 + P_e) + (1 - p)\alpha(P_u^a - P_a^0) \end{aligned} \quad (6)$$

If $E_{\mu_u^R} > E_{\mu_u^A}$, the expected payoff of the malicious application i playing F and playing T are:

$$\begin{aligned} E_{\mu_a^F} &= u(P_a^0 - C_a^m - P_u^e) - (1 - u)\alpha(C_a^m + P_e) \\ E_{\mu_a^T} &= u(P_a^0 - C_a^n) + (1 - u)(\alpha - 1)C_a^n \end{aligned} \quad (7)$$

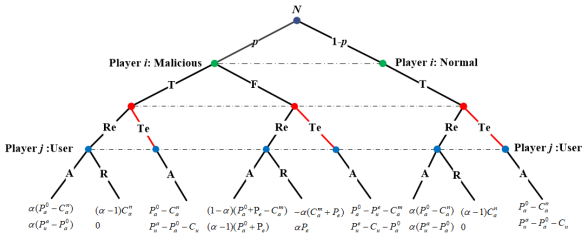


FIGURE 2. The extensive form of two-side Bayesian game model based on sliding adaptive logistic regression method.

From (7), If $\frac{C_a^m+P_u^e}{C_a^m+P_e+C_a^n} < \alpha < 1$, and $\frac{\alpha P_e+\alpha C_a^n+\alpha C_a^m-C_a^n}{(\alpha-1)C_a^m+\alpha P_e-P_u^e+\alpha C_a^n} < u < 1$, then, $E_{\mu_a^F} > E_{\mu_a^T}$. which means the malicious application's optimal strategy is to play F. From Section II, if $\frac{\alpha P_e+\alpha C_a^n+\alpha C_a^m-C_a^n}{(\alpha-1)C_a^m+\alpha P_e-P_u^e+\alpha C_a^n} < u < 1$, the strategy profile ((Malicious application F, Normal application T),(Regular user R, Test user A), p, u) is a Two-side Bayesian Nash equilibrium when the regular user j's optimal strategy is to play R. However, if $\frac{C_a^m+P_u^e}{C_a^m+P_e+C_a^n} < \alpha < 1$ and $0 < u < \frac{\alpha P_e+\alpha C_a^n+\alpha C_a^m-C_a^n}{(\alpha-1)C_a^m+\alpha P_e-P_u^e+\alpha C_a^n}$; or, $0 < \alpha < \frac{C_a^m+P_u^e}{C_a^m+P_e+C_a^n}$ and $0 < u < 1$, that is, $E_{\mu_a^F} < E_{\mu_a^T}$ which means the malicious application i's optimal strategy is to play T, the malicious application i will shift his strategy to play T, therefore, the strategy profile ((Malicious application F, Normal application T),(Regular user R, Test user A), p, u) can not be a Two-side Bayesian Nash equilibrium.

So similar, we can derive that if $0 < \alpha < \frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$ and $0 < u < \frac{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^m-\alpha C_a^n}{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^n-\alpha C_a^m-P_u^e}$, the strategy profile ((Malicious application F, Normal application T),(Regular user A, Test user A), p, u) is a Two-side Bayesian Nash equilibrium when the regular user i's optimal strategy is to play A. The malicious application i is to play T if it is malicious, the normal application i is to play T if it is normal, the user j is to play A if it is test. We can derive that if $0 < \alpha < \frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$, $0 < p < 1$ and $\frac{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^m-\alpha C_a^n}{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^n-\alpha C_a^m-P_u^e} < u < 1$; or, $\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n} < \alpha < 1$, $0 < p < 1$ and $0 < u < 1$. the strategy profile ((Malicious application T, Normal application T),(Regular user R, Test user A), p, u) is the Two-side Bayesian Nash equilibrium.

To sum up, we can set the test user's ratio $u \in [\frac{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^m-\alpha C_a^n}{(2\alpha-1)P_a^0+(\alpha-1)P_e+(1-\alpha)C_a^n-\alpha C_a^m-P_u^e}, 1]$ if $0 < \alpha < \frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n}$; or set the test user's ratio $u \in [0, 1]$ if $\frac{P_a^0+P_e-C_a^m}{2P_a^0+P_e-C_a^m-C_a^n} < \alpha < 1$ to encourage the malicious application to play T.

IV. SIMULATION RESULTS

A. EXPERIMENT SETTING

This article used an integrated development tool "anaconda" to do the simulation verification for our proposed scheme. The process can be divided into two stages:

TABLE 6. Experimental parameters.

Parameter	C_a^n	C_a^m	P_e	P_a^0	P_u^a	P_u^e	C_a	C_u
Value	0.5	0.8	1	1	2	1	0.5	1

TABLE 7. Comparison of the payoff in the two-round bargain method and the application's best bid.

σ	Payoff				Payoff difference	
	Two-round bargain		Application's best bid			
	Application	User	Application	User	Application	User
0.1	0.0005	0.001	1.364	0.136	1.3635	0.135
0.2	0.004	0.008	1.252	0.248	1.248	0.24
0.3	0.0135	0.027	1.158	0.342	1.1445	0.315
0.4	0.032	0.064	1.076	0.424	1.044	0.36
0.5	0.0625	0.125	1	0.5	0.9375	0.375
0.6	0.108	0.216	0.924	0.576	0.816	0.36
0.7	0.1715	0.343	0.842	0.658	0.6705	0.315
0.8	0.256	0.512	0.748	0.752	0.492	0.24
0.9	0.3645	0.729	0.636	0.864	0.2715	0.135
1	0.5	1	0.5	1	0	0

First, we compared and analyzed payoff difference value of applications and users in the Two-round bargain method and the application's best bid which deduced by backward induction, the expected payoff of malicious applications and regular users under the circumstances of different contrast vectors, such as malicious application's ratio, the regular user's detection rate, the transaction achievement rate in three schemes, namely, ***BB***, ***NB*** and ***TS***.

Next, we checked the effectiveness of the Two-side Bayesian game model based on sliding adaptive logistic regression method's capability in solving the problem of malicious applications deceiving users. We compared and analyzed values of prediction by using sliding adaptive logistic regression method and logistic regression method, three influencing factors' influence on the expected payoff of malicious applications and regular users, namely, the malicious application's ratio, the test user's ratio, and the regular user's detection ratio, the probability of malicious applications playing T in the following four schemes, ***TB***, ***PD***, ***HZD*** [27] and ***HIS*** [28]. Experimental parameters setting were shown in Table 6.

B. VERIFICATION OF FAKE BID SOLUTION

1) PAYOFF COMPARISON OF TWO-ROUND BARGAIN AND APPLICATION'S BEST BID

In Table 7, we separately calculated payoffs of applications and users under the condition of the Two-round bargain method and the application's best bid.

From the payoff of the application's best bid in Table 7, we can see that as σ increases, payoffs of applications

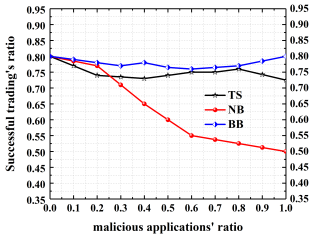


FIGURE 3. Changing trend of transaction achievement rate.

decreases, and payoffs of users increases. From the payoff difference of the Two-round bargain and application's best bid, we can see that payoffs of both players in the application's best bid are higher than that in the Two-round bargain method. And the increase in applications' payoff is more noticeable. Therefore, we can encourage applications to adopt the best bid strategy in the first round of bargain. By using this scheme, we can successfully solve the issue of malicious applications providing fake bids.

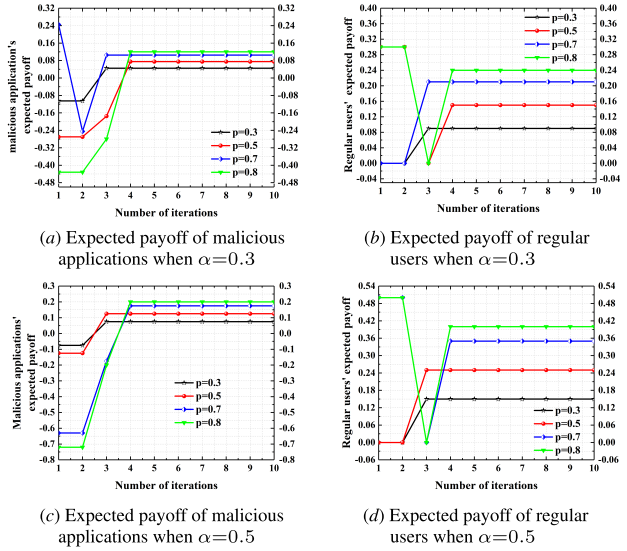
2) COMPARISON OF THE TRANSACTION ACHIEVEMENT RATE

Fig. 3 describes the malicious application's ratio influence on the transaction achievement rate between applications and users. ***BB*** scheme and ***TS*** scheme stay relatively stable when the malicious application's ratio increases, but the ***BB*** scheme is more stable than the ***TS*** scheme. The lowest value of ***BB***'s transaction achievement rate is 0.76, and its value floats between 0.76 to 0.8. But the ***TS*** scheme has a wider floating range, and the lowest value of the transaction achievement rate is 0.72. The reason is that this articled used the method of the Two-round bargain and the application's best bid in the first round of bargain to determine the transaction price. Therefore, even if the malicious application's ratio increases, the transaction achievement rate would not change to a large degree. Comparing the ***NB*** scheme with the ***BB*** scheme and the ***TS*** scheme, the ***NB*** scheme does not prevent malicious applications from giving fake bids and thus causes the failure of transaction and low transaction achievement rate. By using our proposed scheme, the issue of malicious applications sending fake bids can be successfully solved.

3) THE PAYOFF ANALYSIS OF BARGAIN-BAYESIAN GAME

Fig. 4(a)-4(d) describe the trend of expected payoff of malicious applications and regular users when the regular user's detection rate α equals to 0.3 and 0.5.

In Fig. 4(a), when the malicious application's ratio p equals to 0.3, 0.5, and 0.8, the payoff of malicious applications increases first, it then goes stable, which is because before the game gets stable, the malicious application and the regular user adjust their strategies to increase their payoffs. And the regular user's adjusting strategy produces a favorable effect on malicious applications' expected payoff. After adjusting strategies for several times, the game goes stable. When p equals to 0.7, the malicious application's payoff drops from 0.2 to -0.24 in the second interaction, which is because of the



(a) Expected payoff of malicious applications when α=0.3

(b) Expected payoff of regular users when α=0.3

(c) Expected payoff of malicious applications when α=0.5

(d) Expected payoff of regular users when α=0.5

FIGURE 4. The expected payoff comparison.

regular user adjusted strategy in the second round of interaction causes the payoff of malicious applications dropping rapidly. However, in Fig. 4(b), although regular users adjusted its strategy, its payoff did not change, which is because the regular user's way of adjusting strategy is: if adjusting strategy does not lower its payoff, then it will adjust its strategy. Otherwise, it will continue to use the strategy that is used in the last round of interaction. When p equals to 0.5 and 0.8, the regular user's payoff decreases rapidly. The reasons for malicious applications and regular users' expected payoff changing in Fig. 4(c)-Fig. 4(d), Fig. 5(a)-Fig. 5(d) are the same as the above theory.

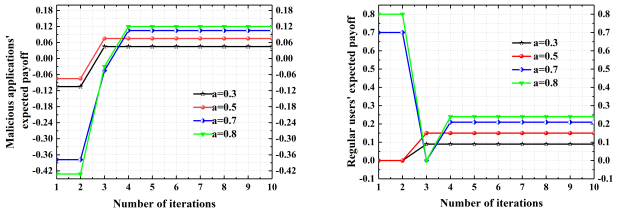
When the game is stable, by comparing Fig. 4(a)-4(b), we can see that when the regular user's detection ratio α is fixed, as p increases, the expected payoff of malicious applications increases, and the expected payoff of regular users decreases. Comparing Fig. 4(a) and Fig. 4(c), when the malicious application's ratio p is fixed, as α increases, the expected payoff of malicious applications and users increase, and the payoff of regular users increases more rapidly. To verify this rule, this article assume that $p = 0.3$ and $p = 0.5$, as shown in Fig. 5(a)-5(d).

Comparing Fig. 5(a) and 5(c), Fig. 5(b) and 5(d), when the game goes stable, as p increases, the payoff of malicious applications and users increase to a certain degree. As α increases, payoffs of both players increase and the regular user's payoff rise more rapidly. Therefore, if regular users want to increase payoff, they should increase their detection rates to prohibit the malicious application's ratio from growing. Besides, if regular users have a higher detection rate, the issue of malicious applications providing false content can be solved more effectively.

C. VERIFICATION OF FALSE CONTENT SOLUTION

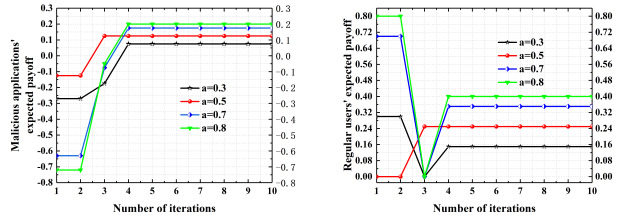
1) SLIDING ADAPTIVE LOGISTIC REGRESSION METHOD

Table 8 describes the predicted probability of malicious applications playing F (i.e., the malicious application provides



(*a*) Expected payoff of malicious applications when $p = 0.3$

(*b*) Expected payoff of regular users when $p = 0.3$



(*c*) Expected payoff of malicious applications when $p = 0.5$

(*d*) Expected payoff of regular users when $p = 0.5$

FIGURE 5. The expected payoff comparison.

false content to users) in two different logistic regression methods. Comparing the predicted probability of two models, the accuracy of predicting malicious applications to play F in our sliding adaptive logistic regression model is more high. For example, if $f = 0.14508$, $s = 0.15632$, and malicious applications play F, the predicted probability of sliding adaptive logistic regression method is 0.9912, and the predicted probability of logistic regression is 0.9873. When $f = 0.17502$, $s = 0.14655$, and malicious applications play T (i.e., the malicious application provides true content to users), the predicted probability of sliding adaptive logistic regression method is 0.1068, and the predicted probability of logistic regression is 0.1460. Comparing with the logistic regression method, when malicious application plays F, the predicted probability in this article is higher, and when malicious application plays T, the predicted probability in this article is lower. To sum up, the sliding adaptive logistic regression method in this article is better than the other model, and the predicted probability of malicious applications playing F is more accurate.

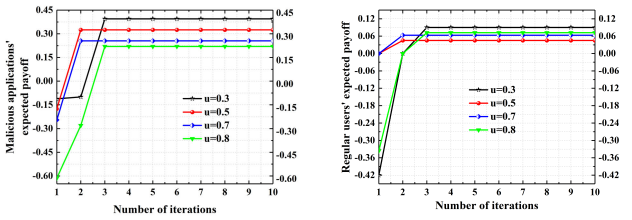
2) THE PAYOFF ANALYSIS OF TWO-SIDE BAYESIAN GAME

Fig. 6-Fig. 8 separately described three factors' relations: the regular user's detection rate α, the test user's ratio p, and the malicious application's ratio u. When two factors' value are fixed, how will the third factor affect the expected payoff of malicious applications and regular users.

From Fig. 6(a)-6(b), we can see that expected payoffs of malicious applications and regular users both increase at the beginning and then go stable. In the first several rounds of interactions, expected payoffs of malicious applications and regular users keep growing because their adjusting strategies produce positive effects on two players' payoffs. After their adjusting strategies several times, the game goes stable and their payoff do not change from then. Comparing Fig. 6(a)-6(b), as u increases, payoffs of two

TABLE 8. Comparison of the predicted probability.

Times	Test data		Strategy	The predicted probability	
	First payoff (f)	Second payoff (s)		Logistic regression	Sliding adaptive logistic regression
1	0.13462	0.17802	***F***	0.9624	0.9725
2	0.13029	0.14389	***F***	0.946	0.9602
3	0.13585	0.1729	***F***	0.9825	0.9833
4	0.16018	0.18631	***T***	0.006	0.001
5	0.17903	0.17534	***T***	0.0003	0.0003
6	0.14508	0.15632	***F***	0.9873	0.9912
7	0.16111	0.19651	***T***	0.0074	0.0004
8	0.17502	0.14655	***T***	0.146	0.1068
9	0.1761	0.18742	***T***	0.0001	0.0001
10	0.18443	0.14353	***T***	0.0266	0.0136
11	0.19586	0.13823	***T***	0.0029	0.0021
12	0.17501	0.1306	***T***	0.762	0.7618
13	0.18231	0.17648	***T***	0.0001	0.0001
14	0.16936	0.19772	***T***	0.1974	0.1974
15	0.13954	0.17604	***F***	0.8958	0.9058
16	0.15397	0.18921	***T***	0.0167	0.0101
17	0.16907	0.15274	***T***	0.1974	0.1974
18	0.16795	0.14668	***F***	0.5012	0.5012
19	0.17066	0.19293	***T***	0.0011	0.0011
20	0.17698	0.14758	***T***	0.0729	0.0729

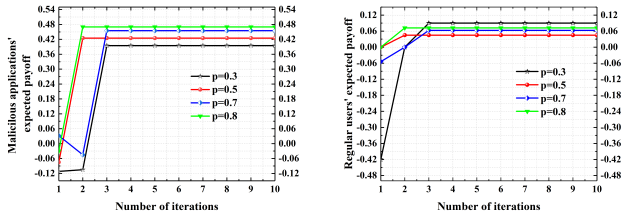


(*a*) Expected payoff of malicious applications (*b*) Expected payoff of regular users

FIGURE 6. The changing trend of expected payoff when $\alpha = 0.3$, $p = 0.3$.

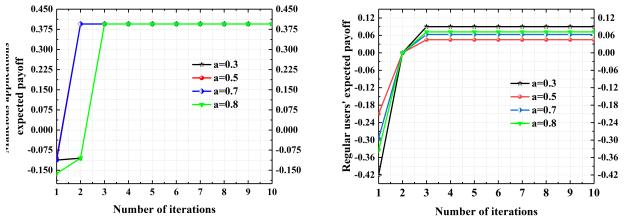
players increased. Therefore, when the values of α and p are fixed, by adjusting u we can successfully achieve the goal of increasing their payoffs.

In Fig. 7(a), if α equals to 0.3 and 0.5, the trend of expected payoff of malicious applications is the same. If α equals to 0.7 and 0.8, the trend of expected payoff of malicious applications is basically the same as well. In Fig. 7(b), before the game goes stable, the expected payoff of regular users have been increasing and in the second round of interaction, four expected payoff curves cross at the same point. This is because adjusting regular users' detection rate α does not effectively influence both players' payoff matrix, but only slightly changes its own expected payoff. After the game goes stable, by comparing Fig. 7(a)-7(b), when the values of p and u are fixed, only by adjusting the value of α can change regular users' expected payoffs.



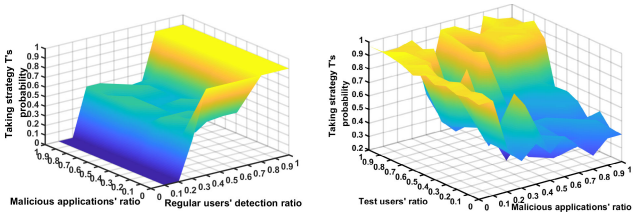
(*a*) Expected payoff of malicious applications (*b*) Expected payoff of regular users

FIGURE 7. The changing trend of expected payoff when $\alpha = 0.3$, $p = 0.3$.



a) Expected payoff of malicious applications (*b*) Expected payoff of regular users

FIGURE 8. The changing trend of expected payoff when $p = 0.3$, $u = 0.3$.



(*a*) Bargain-bayesian Game (*b*) Two-side Bayesian Game

FIGURE 9. Comparison probability of malicious applications playing T.

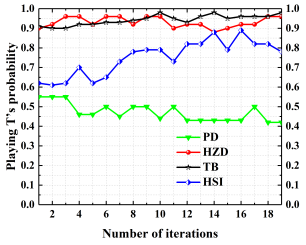


FIGURE 10. The probability comparison of playing T.

In Fig. 8(*a*), when $p = 0.7$, the expected payoff of malicious applications drops rapidly in the second round of interactions. In Fig. 8(*b*), the payoff of regular users increases. This is because regular users adjust their own strategy, which helps to increase their own payoff and lowers the payoff of malicious applications. By comparing Fig. 8(*a*)-8(*b*), after the game goes stable, when the value of α and u are fixed, as p increases, payoffs of malicious applications increases and regular users' payoffs decreases. Therefore, if users want to increase their own payoff, they should prevent the malicious application's ratio from growing.

3) PROBABILITY COMPARISON OF MALICIOUS APPLICATIONS PLAYING T

Fig. 9 analyzes the probability of malicious applications playing T in two game models, i.e., Bargain-bayesian game model and Two-side Bayesian game based on sliding adaptive logistic regression model.

Fig. 9(*a*) describes the relation between regular user's detection rate α, the malicious application's ratio p and the probability of malicious applications playing T in the *BB* model. When $\alpha = 0.2$, malicious applications start to play T, and the probability of playing T increases as increases α. When $\alpha \in [0.4, 0.7]$, under the influence of p, the probability of malicious applications playing T floats around 0.5. When $\alpha = 0.8$, the probability of malicious applications playing T increases again until the probability of playing T reaches 0.9. Therefore, we conclude that by adjusting regular users' detection rates, we can prevent malicious applications from providing false content to deceive users to a certain degree.

Fig. 9(*b*) describes the influence of the malicious application's ratio p and the test users' ratio u on malicious applications playing T in the ***TB*** model. When p < 0.2, even if the test user's ratio is 0, the probability of malicious applications playing T can reach 0.9, much better than its performance when $\alpha < 0.8$ in Fig. 9(*a*). However, when $p \in [0.2, 0.5)$, the probability of malicious applications playing T is not stable, this is because by adjusting ratios of test users and malicious applications, the payoff matrix changes as well as the malicious applications' best strategy. And when $p \in [0.5, 1)$, the probability of malicious applications playing T increases as u increases. Comparing with $\alpha \in (0.8, 1]$'s effects in Fig. 9(*a*), this model has weaker effects than the ***BB*** model when $u < 0.6$ and $p > 0.5$. But in real operation, α rarely reaches 0.8 and the minimum probability of the malicious application playing T is 0.4 in the ***TB*** model. As u increases, the ***TB*** model proves to be more effective. Therefore, we conclude that ***TB*** model is better than the ***BB*** model for encouraging malicious applications playing T.

Fig. 10 shows the changing trend of probability that the malicious application plays T in four different schemes, e.g., ***PD***, ***HSI***, ***HZD***, and ***TB***, when $u = 0.6$. As the ***PD*** scheme does not play any method in the inhibition of the malicious application playing F, the probability that the malicious application plays T is less than 0.5. In the ***HSI*** scheme, the probability that the malicious application plays T keeps increasing in the first 13 iterations and then drops down after reaching the peak value. Its probability that the malicious application plays T is a bit lower than that of the ***HZD*** scheme and ***TB*** scheme. In ***HZD*** scheme, the probability that the malicious application plays T fluctuates between 0.85-0.96. Comparing the ***HSI*** scheme and the ***HZD*** scheme, the probability that the malicious application plays T is more stable and keeps at a high level in ***TB*** scheme. Though the effect of ***TB*** scheme is lower than the ***HZD*** scheme within the first 8 iterations, its probability keeps growing after the 10th iterations and stays stable, and its value is always greater than 0.9 in each iteration. In conclusion, ***TB*** model can increase the probability that the malicious application plays T, and to further solve the problem of the malicious application providing the false content problem.

V. CONCLUSION

In the multimedia communication system, to solve the problem of user privacy leakage caused by malicious applications deceiving users is inevitable to promote efficient applications. We proposed an Anti-fraud scheme based on an improved Bayesian game model to solve this problem effectively. The Bargain-bayesian game model can improve the transaction achievement rate and payoffs of both players to address the problem of malicious applications sending fake bids. The Two-side Bayesian game model based on a sliding adaptive logistic regress method can adjust the test user's ratio to inhibit the malicious application to provide false content for deceiving users. Detailed simulation experiments verify the effectiveness of proposed proposals.

In future research, we wish to improve our scheme by designing new machine learning model to predict the malicious application's ratio. And we will continue to focus on the study of multimedia information security issues and apply game theory to solve more privacy protection problems.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

REFERENCES

[1] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," *Comput., Mater. Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[2] C. Yuan, Z. Xia, and X. Sun, "Coverless image steganography based on SIFT and BOF," *J. Internet Technol.*, vol. 18, no. 2, pp. 435–442, Mar. 2017.

[3] I. J. Cox, J. Kilian, T. Shamoon, and F. T. Leighton, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[4] E. Etemad, S. Samavi, S. M. Reza Soroushmehr, N. Karimi, M. Etemad, S. Shirani, and K. Najarian, "Robust image watermarking scheme using bit-plane of Hadamard coefficients," *Multimedia Tools Appl.*, vol. 77, no. 2, pp. 2033–2055, Jan. 2018.

[5] A. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[6] T. Pomari, G. Ruppert, E. Rezende, A. Rocha, and T. Carvalho, "Image splicing detection through illumination inconsistencies and deep learning," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, vol. 3, Oct. 1998, pp. 3788–3792.

[7] D. Hu, B. Su, S. Zheng, Z. Q. Zhao, X. Wu, and X. Wu, "Security and privacy protocols for perceptual image hashing," *Int. J. Sens. Netw.*, vol. 17, no. 3, p. 146, 2015.

[8] C. Qin, Y. Hu, H. Yao, X. Duan, and L. Gao, "Perceptual image hashing based on weber local binary pattern and color angle representation," *IEEE Access*, vol. 7, pp. 45460–45471, 2019.

[9] X. Lu, W. Zhang, and X. Li, "A hybrid sparsity and distance-based discrimination detector for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1704–1717, Mar. 2018.

[10] A. S. Sohal, R. Sandhu, S. K. Sood, and V. Chang, "A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments," *Comput. Secur.*, vol. 74, pp. 340–354, May 2018.

[11] H. Deng, X. Sun, M. Liu, C. Ye, and X. Zhou, "Small infrared target detection based on weighted local difference measure," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4204–4214, Jul. 2016.

[12] K. Wang, M. Du, D. Yang, C. Zhu, J. Shen, and Y. Zhang, "Game-theory-based active defense for intrusion detection in cyber-physical embedded systems," *ACM Trans. Embed. Comput. Syst. (TECS)*, vol. 16, no. 1, pp. 1–21, Oct. 2016.

[13] L. Zheng, Y. Zhang, and V. L. Thing, "A survey on image tampering and its detection in real-world photos," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 380–399, Jan. 2019.

[14] Q. Yang, A. Lim, X. Ruan, and X. Qin, "Location privacy protection in contention based forwarding for VANETs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[15] J. Fridrich, "Image watermarking for tamper detection," in *Proc. Int. Conf. Image Process.*, Nov. 2002, pp. 404–408.

[16] R. Zhang, D. Xiao, and Y. Chang, "A novel image authentication with tamper localization and self-recovery in encrypted domain based on compressive sensing," *Secur. Commun. Netw.*, vol. 2018, pp. 1–15, Mar. 2018.

[17] W. Wei, X. Fan, H. Song, and H. Wang, "Video tamper detection based on multi-scale mutual information," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27109–27126, Oct. 2019.

[18] H. Wu and W. Wang, "A game theory based collaborative security detection method for Internet of Things systems," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 6, pp. 1432–1445, Jun. 2018.

[19] Y. Li, Z. Zhang, H. Wang, and Q. Yang, "SERS: Social-aware energy-efficient relay selection in d2d communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5331–5345, Jun. 2018.

[20] Z. Ni and S. Paul, "A multistage game in smart grid security: A reinforcement learning solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2684–2695, Sep. 2019.

[21] A. Attiah, M. Chatterjee, and C. C. Zou, "A game theoretic approach to modelcyber attack and defense strategies," in *Proc. IEEE Int. Conf. Commun. (ICDCS)*, May 2018, vol. 56, no. 3, pp. 1–7.

[22] L. Xiao, D. Xu, C. Xie, N. B. Mandayam, and H. V. Poor, "Cloud storage defense against advanced persistent threats: A prospect theoretic study," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 534–544, Mar. 2017.

[23] X. Tang, P. Ren, and Z. Han, "Power-efficient secure transmission against full-duplex active eavesdropper: A game-theoretic framework," *IEEE Access*, vol. 5, pp. 24632–24645, 2017.

[24] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: Privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1868–1878, Dec. 2017.

[25] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6492–6499, Dec. 2019, doi: 10.1109/tii.2019.2911697.

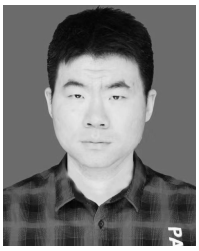
[26] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Trans. Netw. Sci. Eng.*, early access, 2018, doi: 10.1109/tnse.2018.2830307.

[27] Q. Hu, S. Wang, L. Ma, R. Bie, and X. Cheng, "Anti-malicious crowdsourcing using the zero-determinant strategy," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1137–1146.

[28] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1402–1410.

RUI ZHANG was born in 1995. She is currently pursuing the master's degree with the College of Computer Science and Technology, Qingdao University. Her current research interest includes privacy protection.

HUI XIA (Member, IEEE) was born in 1986. He received the Ph.D. degree in computer science from Shandong University, in 2013. Since December 2019, he has been a Professor and a Ph.D. Supervisor with the College of Information Science and Engineering, Ocean University of China. He has published over 43 articles. His research was supported by the Natural Science Foundation of China (NSFC) under Grant 61872205, the Shandong Provincial Natural Science Foundation under Grant ZR2019MF018, and the Source Innovation Project of Qingdao under Grant 18-2-2-56-jch. His research interests include the social IoT, the IoT security, vehicular ad hoc networks, crowdsourcing, edge computing, and privacy protection. He is a member of the CCF and the IEEE Computer Society.



FEI CHEN received the M.S. degree from the School of Electronics Engineering and Computer Science, Northeastern University, Shenyang, China, in 2009, and the Ph.D. degree from the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, in 2014. He is currently an Associate Professor with the College of Computer Science and Technology, Qingdao University, Qingdao, China. His research interests include cloud computing, peer-to-peer networks, crowd-sensing systems, and multimedia communications.

LI LI (Student Member, IEEE) was born in 1994. She is currently pursuing the Ph.D. degree with the Ocean University of China. Her research interests include network and information security, and the IoT security. She is a Student Member of the CCF.

XIANG-GUO CHENG was born in 1969. He is currently a Professor with the College of Computer Science and Technology, Qingdao University, China. His main research interests include cloud security, computer security, and public key cryptosystems.