Computers and Mathematics with Applications 65 (2013) 1384–1395

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Online game bot detection based on party-play log analysis

Ah Reum Kang [a], Jiyoung Woo [a], Juyong Park [b], Huy Kang Kim [a,*]

[a] *Center for Information Security Technologies (CIST), Graduate School of Information Security, Korea University, 5-Ga Anam-Dong, Seoungbuk-Gu, Seoul, 136-701, Republic of Korea*

[b] *Department of Physics, College of Sciences, Kyung Hee University, 1 Hoegi-Dong, Dongdaemun-Gu, Seoul, 130-701, Republic of Korea*

ARTICLE INFO

Keywords:
Online game security
Game bot
User behavior analysis
MMORPG

ABSTRACT

As online games become popular and the boundary between virtual and real economies blurs, cheating in games has proliferated in volume and method. In this paper, we propose a framework for user behavior analysis for bot detection in online games. Specifically, we focus on party play which reflects the social activities among gamers: in a Massively Multi-user Online Role Playing Game (MMORPG), party play is a major activity that game bots exploit to keep their characters safe and facilitate the acquisition of cyber assets in a fashion very different from that of normal humans. Through a comprehensive statistical analysis of user behaviors in game activity logs, we establish threshold levels for the activities that allow us to identify game bots. Based on this, we also build a knowledge base of detection rules, which are generic. We apply our rule reasoner to AION, a popular online game serviced by NCsoft, Inc., a leading online game company based in Korea.

1. Introduction

As online games become popular and the boundary between virtual and real economies blurs, illegal activities in online game environments have drastically increased and become more diverse. Thus, security has become an important issue in the online game industry. The aim of online game security is to protect the assets, not only of online game publishers (systems, networks, database, and applications), but also of gamers (personal information and cyber assets), from cheaters and hackers. Halim et al. [1] classified computer games into four genres, i.e., classical perfect information games, classical imperfect information games, video games, and real world games such as MMORPGs. Among these genres in real world games, illegal activities frequently occur. Since items and currency acquired in games can be sold to other players for profit in real money, cheaters employ game bots strictly outlawed by game publishers [2]. Kim [3] classified game bots into two types, i.e., physical types and running types, as shown in Table 1.

A Game Bot is an automated program that plays the game on behalf of human players. Since they can play without break, game bots can accumulate money and items much faster than normal human players. Providers of game bots lure users with the promise of convenience, such as remote software installation, and avoid copyright law enforcement by operating in obscure territories.

The cheaters destroy the game balance by rapidly depleting in-game contents and resources. Honest human gamers may thus feel deprived, lose interest, and eventually leave the game. This poses a huge problem for game publishers, since the number of gamers is related directly to profit. However, efforts on the part of game publishers to detect game bots via network monitoring and security enhancement have so far shown to have room for improvement, and sometimes they cause other problems (high maintenance costs and the false identification of real users as bots). To overcome these potential

[*] Corresponding author. Tel.: +82 2 3290 4898; fax: +82 2 928 9109.
E-mail addresses: armk@korea.ac.kr (A.R. Kang), jywoo@korea.ac.kr (J. Woo), perturbation@khu.ac.kr (J. Park), cenda@korea.ac.kr (H.K. Kim).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.01.034

Table 1
Game bot taxonomy [3].

Taxonomy	Characteristics
Categorized by bot's physical types: software type, USB type and mouse type	- As of recently, most commercial game bots provide an evasion function for CAPTCHA authentication.
Categorized by bot's running types: OOG bot (known as out-of-game client bot) and IG bot (known as in-game-client bot)	- Most commercial game bots provide a chatting response function to react to the game master (GM) who monitors and detects bot programs by human observation.

drawbacks, the game industry is actively seeking highly accurate bot detection methods that reduce inconvenience to both the user and the system.

In this study, we propose a bot detection framework based on user behavior analysis. In Section 2, we review previous studies and some actual bot detection strategies employed by the game industry, highlighting their problems and limitations. In Section 3, we address the details of our proposed bot detection framework based on user behavior analysis. In Section 4, we evaluate our proposed method on actual in-game log data from a popular MMORPG. Finally, in Sections 5 and 6 we discuss our findings and explore possible future directions.

2. Literature review

2.1. Bot detection methods in the game industry

Hu and Zambetta [4] characterized Massively Multi-user Online Games (MMOGs) security into attributes, threats and means from the CIA (confidentiality, integrity, availability) perspective. Park and Lee [5] classified online game attacks into four categories: server attacks, network attacks, client attacks and user attacks. Since modern bots are integrated with these categories, we classified current bot detection methods into three classes, i.e., client-side, network-side, and server-side, as shown in Table 2.

Client-side security software often causes collisions in the operating system, resulting in inconvenience for users. For instance, intentional game designs such as invisible items that only a game bot can detect can in principle sometimes be revealed to human users, becoming taxing but worthless to all involved parties. Network-side detection methods such as network traffic monitoring or network protocol change cause network overload and lags in game play, a significant annoyance in the online gaming experience. To overcome these drawbacks of client-side and network-side detection methods, game publishers have begun showing interest in server-side detection methods, especially log mining to identify bot-specific activity pattern from in-game logs. It is believed that server-side log mining methods can produce highly accurate, effective rules to detect game bots since they do not interfere with game play, by relying on pre-defined detection algorithms.

2.2. Previous server-side detection research

Table 3 summarizes and compares bot detection schemes. We present key bot detection methods, classified into five categories, namely user behavior analysis, moving path analysis, traffic analysis, human observation proofs analysis, and CAPTCHA analysis.

CAPTCHA (Complete Automated Public Turing Test to Tell Computers and Humans Apart) analysis requests answers that can be easily solved by humans, but are hard to solved by bots. Yampolskiy and Govindaraju [6] proposed integrating the testing procedure as part of the game step performed by the player during the game to distinguish bots from legitimate human players.

Golle and Ducheneaut [7] demonstrated embedding CAPTCHA test into games to minimize the disruption compared to the out-of-band use of CAPTCHAs.

Traffic analysis uses network traffic information such as command packets timing, traffic explosiveness, network response speed, data length and traffic interval time. Chen et al. [8] studied the traffic regularity in the release time of client commands, the traffic burstiness in multiple time scales, and the traffic sensitivity to different network conditions as a means to identify bots. Hilaire et al. [9] observed that bots send less information than humans and exhibit regular and fast packet arrivals patterns.

User behavior analysis relies on the idea that there are differences between human behaviors and programmed bot behaviors such as idle time or social connection. Thawonmas et al. [10] studied discrepancies in action frequencies and action types in the log between humans and bots. Chen and Hong [11] proposed a relative entropy test scheme based on the Kullback–Leibler divergence between idle time distributions. Yeung et al. [12] implemented a prototype multiplayer game system based on the dynamic Bayesian network for cheating detection. Varvello and Voelker [13] analyzed the social connections of avatars to detect bots in the social graph. Yan [14] developed a Dynamic Bayesian Network-based statistical inference approach to distinguish humans from bot cheaters in FPS games. Ahmad et al. [15] used several machine learning binary classification techniques, such as BayesNet, NaiveBayes and KNN, to identify gold farmers.

Table 2
Bot detection methods in the game industry.

Category	Method	Examples	Merits/demerits
Client side detection	Game publisher's anti-cheat system	- Blizzard's warden system for World of Warcraft - NCsoft's NC guard for AION - TenGuard by tencent - SNDC by SNDA	- Demerit: collision with other security programs such as DRM and antivirus
	Third party game security solution	- INCA Internet's GameGuard - AhnLab's Hackshield	- Demerit: collision with other security programs such as DRM and Antivirus
	In-game design	- Detection with invisible NPC (non-player character)s: the normal human player cannot see these NPCs, only a bot program can notice and attack them - Detection with invisible items: the normal human player cannot see these items, only a bot program can be aware of these items and obtain them	- Demerit: temporal method (useless when the design is revealed)
	Detection method based on user reputation	- Kount, iOvation	- Demerit: high false positive rate
Network side detection	Traffic monitoring: - TTL (time to live) values, RTT (round trip time) values in the response	Not implemented in the field yet.	- Demerit: high false positive rate, low availability and heavy analysis cost
	Network protocol change: - Key exchange and encryption algorithm	Lineage, LineageII and AION by NCsoft	- Demerit: continuous update and network traffic cost led by client program update; more computing power for encryption and decryption in realtime
Server side detection	In-game log analysis		- Merit: high accuracy rate, high detection rate with out-game log
	In-game CAPTCHA analysis	- NCsoft's AION	- Demerit: reducing immersion of players in online game Low availability

Table 3
Recent research on bot detection.

Category	Adapted method	Definition/key papers	Key idea	Merits/demerits
Client side detection	CAPTCHA analysis	Detection method with challenge–response test [6,7]	- Challenge–response method	- Merit: high speed - Demerit: reducing immersion of players in online game Low feasibility
Network side detection	Traffic analysis	Detection method based on network traffic analysis [8,9]	- Command packets timing analysis - Traffic explosiveness analysis - Networks response analysis - Data length analysis - Traffic interval time analysis	- Merit: high utilization of the other algorithms like decision tree - Demerit: low accuracy rate
Server side detection	User behavior analysis	Detection method based on user behavior pattern in game play [10–15]	- Idle time analysis - Social connection analysis (chatting, trade)	- Merit: high accuracy rate, high detection rate, high availability
	Moving path analysis	Detection method based on patterns and zones of moving path analysis [16–18]	- Coordinate analysis - Zone analysis	- Merit: high feasibility - Demerit: low accuracy rate
	Human observation proofs (HOP) analysis	Detection method with keyboard and mouse input patterns analysis [19,20]	- User inputs observation - Windows event sequence analysis	- Merit: high accuracy rate - Demerit: low feasibility

Moving path analysis uses the fact that most bots have pre-scheduled moving paths, whereas humans have a variety of moving patterns. Kesteren et al. [16] differentiated bots and humans based on the difference in their movement patterns.

Table 4
Classification of user behaviors in MMORPG.

Exploration	Combat	Craft	Socializing
- Training - Healing - Employment (i.e., quests) - Gathering resources - Gaining experience [11,15]	- Hunting mobiles - Player-versus-player (PvP) combat	- Producing items - Combining materials	- Communication using text messages - Trade - Haggling - Arguing - Socializing through group, guild, clan - Friendships [10,13]

Mitterhofer et al. [17] identified characters controlled by a script with movement patterns repeated frequently. Thawonmas et al. [18] proposed a method for detecting landmarks from player trails based on the weighted entropy of the distribution of visiting players in a game map of interest and visualized player clusters based on the transition probablilities.

Human observation proofs analysis uses keyboard and mouse input patterns. Kim et al. [19] analyzed the window event sequences produced by the game players to detect the auto programs. Gianvecchio et al. [20] identified bots by passively monitoring user input actions such as keystroke, mouse cursor position changes, non-action, and mouse drag and drop travels.

2.3. User behavior classification

User behaviors can be categorized into four classes according to Kelly [21], as shown in Table 4. Previous studies on game-log analysis focus on a single type of activity, such as exploration-related or socializing-related; to the best of our knowledge, so far there has been no serious attempt to encompass various types of user behaviors simultaneously.

From the review of previous works, we have identified several significant avenues for research. First, bot detection based on data mining and statistical inference is still in its infancy. Second, previous studies have dealt with a limited number of actions, ignoring a potentially significant portion of data. Third, the effect of "collective" actions involving multiple users has been largely neglected, even though it appears that the social aspect is a major part of the MMORPG experience for humans.

3. User behavior analysis for bot detection

Here we propose a user behavior analysis framework for online game bot detection. We specifically focus on party play, which reflects the social activities of gamers. We define party play to mean two or more players forming a group to undertake quests or missions together. Users in party play typically share experience points, money and items acquired upon completion of successful quests. The reason we focus on this is that most MMORPGs require and encourage some degree of party play. The goal of party play for humans is different from that of game bots: while humans seek party play to complete quests that are too difficult for a single player, thereby maximizing the fun one may have in a game, game bots, whose goal is acquisition of items, may continue to hunt or harvest even under party play. Typically, two bots form a group (party), in which one hunts or harvests while the other acts as the bodyguard. Thus we expect any repetitive nature of activities under party play to indicate whether the party members are humans or bots, and this forms the basis of our analysis.

We schematize our proposed framework in Fig. 1. First, we extract party-play records from the in-game log, and identify individuals and their detailed activities. Second, we measure the duration of parties, and identify outlier parties that last an exceedingly long time: since parties are to be disbanded once their goals are met, bot parties whose goal is to accrue as many resources as possible are expected to last indefinitely.

To validate our hypothesis that bot parties have different goals to those of human parties, we identify the major actions taken by parties, and proceed to characterize the details of action logs based on the following two metrics: (1) The proportion of an action over total actions, and (2) The ranking of an action log. We adopt these two because some actions have very small occurrences, so they cannot be used to differentiate between bot parties and normal parties. However, their rankings show large differences between the two, so they can be important classifiers. Through the analysis of the action log, we can derive significant classifiers that distinguish between the bot party and the normal party. Threshold values are set on selected classifiers as the proportion of an action over the total actions performed by outlier parties. Threshold values can be easily updated in changing environmental conditions [22].

Based on these two metrics we build a rule base using the domain knowledge of AION,[1] a popular online game served by NCsoft, Inc., a major Korean games provider. We then apply the rules to actual party-play action logs to identify online game bots. The performance of our rules is evaluated by cross-checking the bots identified by our method against the confirmed list of bots provided by the company.

[1] http://na.aiononline.com/.

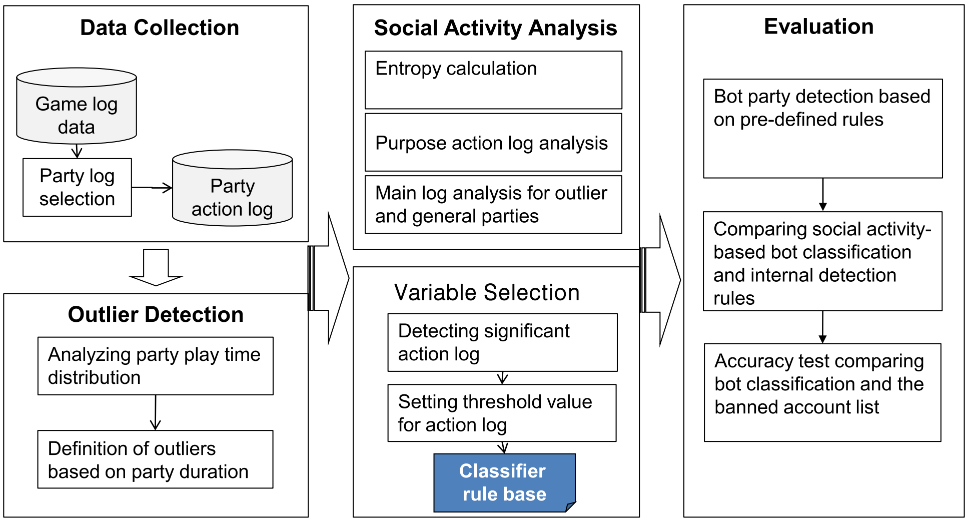

Data Collection
Game log data
Party log selection
Party action log
Outlier Detection
Analyzing party play time distribution
Definition of outliers based on party duration
Social Activity Analysis
Entropy calculation
Purpose action log analysis
Main log analysis for outlier and general parties
Variable Selection
Detecting significant action log
Setting threshold value for action log
Classifier rule base
Evaluation
Bot party detection based on pre-defined rules
Comparing social activity-based bot classification and internal detection rules
Accuracy test comparing bot classification and the banned account list

Fig. 1. Bot detection based on user behavior analysis.

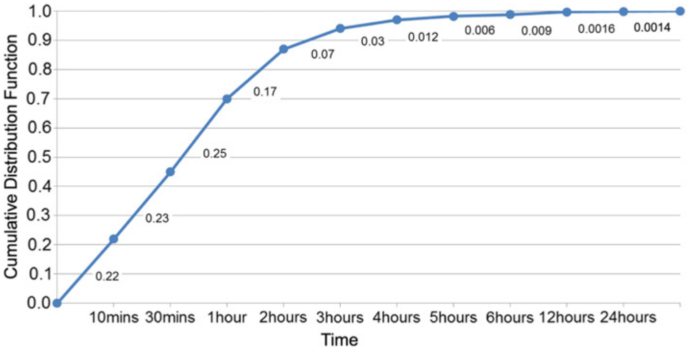


Fig. 2. The cumulative distribution function of party duration.

4. Results

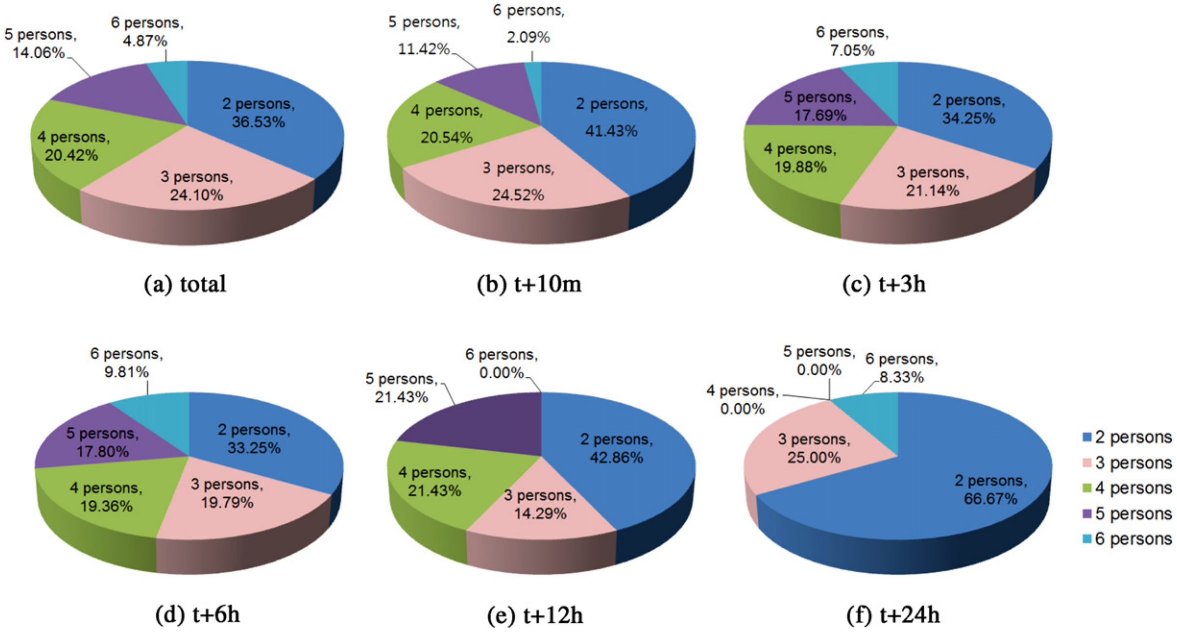
NCsoft, Inc., maintains forty-three servers dedicated to AION, and is capable of hosting nearly 240,000 concurrent users. About 12,000 players log into a server, and about 74,000,000 logs are recorded per day. AION's in-game log contains information on the party, legion (guild), hunt, trade, harvest, and chat activities of users [23]. We found 63,092 parties in a seven days' worth of logs.

The cumulative distribution of party duration (shown in Fig. 2) shows that 0.3% of all parties last longer than 12 h, which we judged to be bot parties. 22% of the parties lasted fewer than ten minutes, and these were excluded from analysis (their extremely short durations were probably due to system errors).

To be able to discriminate bot parties from the others, we examined the party members according to party duration. Fig. 3 presents the relative proportions of parties of a certain size for a given party duration. First, chart (a) shows the average proportion of party sizes for all parties. Charts (b) to (f) show the proportions for given durations: for instance, chart (b) shows parties whose durations are between ten minutes and three hours. As we illustrate, using the pie charts, normal (human) parties show similar sizes. Outliers, however, show markedly a different makeup: the fraction of 2-member parties is much higher in chart (f) and somewhat higher in chart (e) than for the other charts. This is highly consistent with our hypothesis that bots form parties in groups of two so that one can protect the other while it is harvesting or hunting for items.

This is further corroborated by our analysis of the purposes for the actions of the parties. Fig. 4 illustrates the proportion of main action logs embedding the purposes of the party according to the party duration, showing that normal parties have a similar makeup in the purposes of party play.

The ratio of 'Getting Race Point' (race points can be earned when defeating another player in the combat) log in normal parties is over 5%. The ratio of 'Harvesting Item' log in normal parties is below 2%, whereas it is around 3% for the outlier parties. Also, the ratio of 'Quest Completion' log in outlier parties is much lower than that of normal parties. This strongly supports that the completion of difficult quests is characteristic of human parties, not bot parties. Even 'Instance Dungeon



(a) total (b) t+10m (c) t+3h

(d) t+6h (e) t+12h (f) t+24h

Fig. 3. The average number of party members.

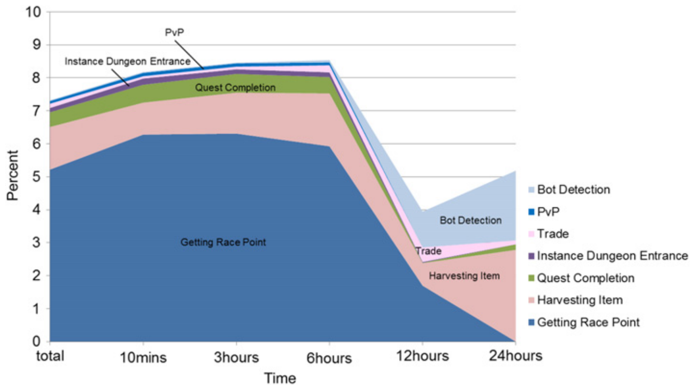
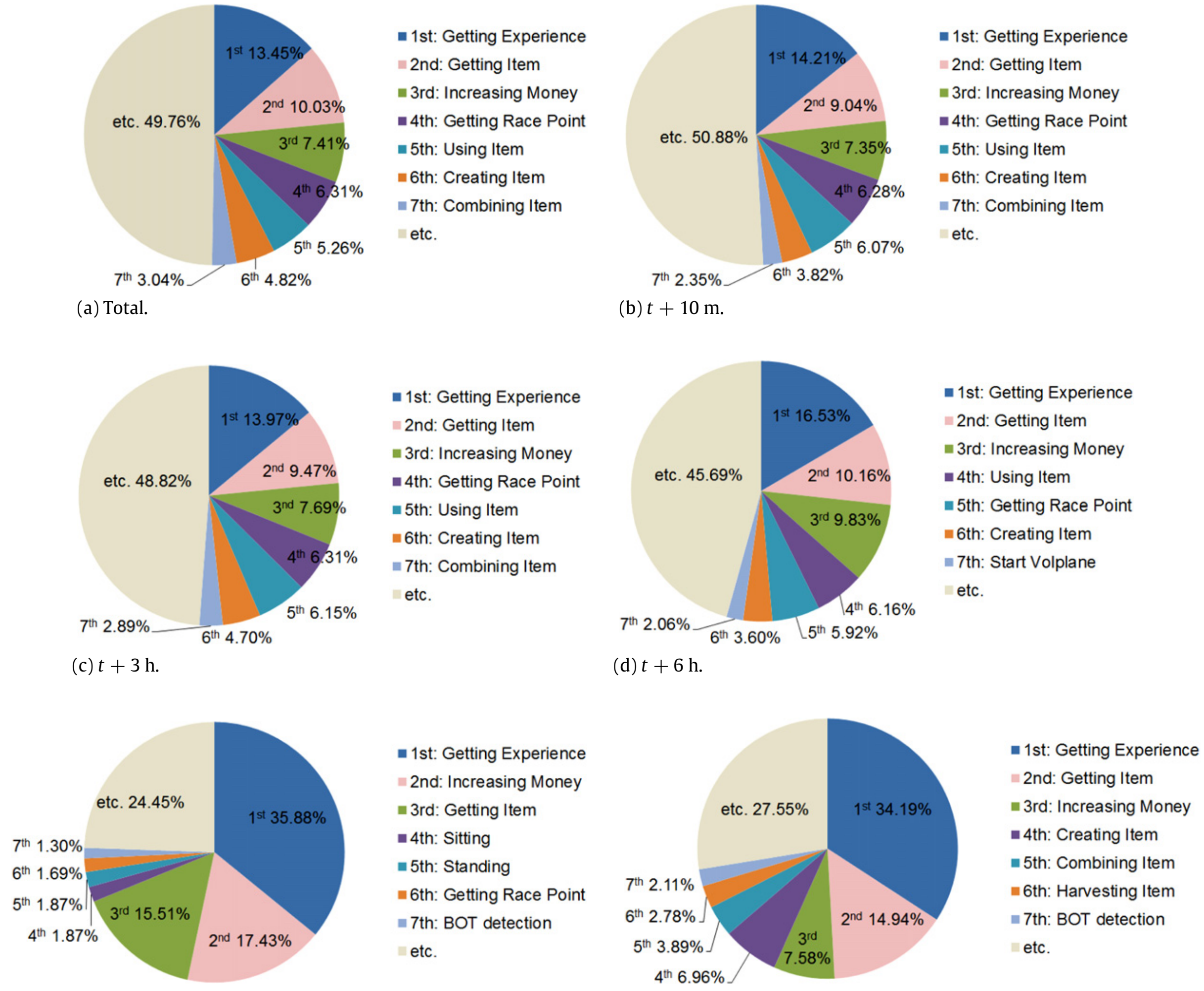


Fig. 4. The main purpose action logs.

Entrance' log (recorded when a player goes to a special place to complete difficult quests) and 'PvP' logs (recorded when a player attacks other players) are not found. It shows that game bots rarely if ever interact with other player groups, and practically are interested only in harvesting items and trading them. By contrast, the purposes of normal parties are diverse: getting race points, quest completion and instance dungeon entrance are all commonly found. The purposes of outliers are limited to hunting, harvesting, gathering materials and crafting. Our analysis thus strongly suggests that the purposes of normal parties and outliers are indeed different. We also cross-checked against the 'Bot Detection' log recorded internally at NCsoft. As shown in Fig. 4, the agreement is high: parties lasting over 12 h, designated as bots by our method, are highly likely to have been determined to be bots by the company as well. Next, we studied distinctive features in the main purpose action logs of normal parties and outliers, and discovered the main purpose of bot parties.

To compare the diversity of actions of bot parties and normal parties, we investigated the top seven action logs according to party duration and derived the entropy index that expresses the diversity of action logs. Fig. 5 shows the top seven behavior logs of each group according to party duration as pie charts. As shown in the charts, there is no significant difference between chart (a) and the other charts, except for charts (e) and (f) in the ratio of the top seven logs. We observed that the distribution of action logs is biased to the top 2 logs of 'Getting Experience' log and 'Getting Item' log in the outlier parties. While the sum of the percentages of the 'Getting Experience' log and 'Getting Item' log in the normal parties are up to 26%, they are 51% and 49% in charts (e) and (f), respectively.



(a) Total.

(b) $t + 10$ m.

(c) $t + 3$ h.

(d) $t + 6$ h.

(e) $t + 12$ h.

(f) $t + 24$ h.

Fig. 5. The top seven behavior logs according to party duration.

Table 5
Entropy according to party duration.

	Total	$t + 10$ m	$t+3$ h	$t+6$ h	$t + 12$ h	$t + 24$ h
Entropy	4.779	4.835	4.731	4.679	3.338	3.322

We calculated the entropy of action log distribution using Formula (1):

$$-\sum_{i=1}^{n} Pi \log_2 Pi \tag{1}$$

n: the number of logs (for log $i = 1, 2, 3, \dots, n$).

Pi: the probability of the occurrence of a log i.

As listed in Table 5, the entropy value decreases as the party duration increases. Outliers parties take less diverse actions than normal parties.

Fig. 6 presents the entropy value according to party duration. The entropy decreases sharply at 12 h.

For further action log analysis, we analyzed the rankings of the most frequent logs by normal users. Fig. 7 shows the rankings of the most frequent logs by normal user in a line graph.

Since normal parties execute multiple tasks as needed to thrive in MMORPGs, their rankings on major logs tend to be similar regardless of party duration. As we expected, 'Getting Experience' log, 'Getting Item' log and 'Increasing Money' log rank high

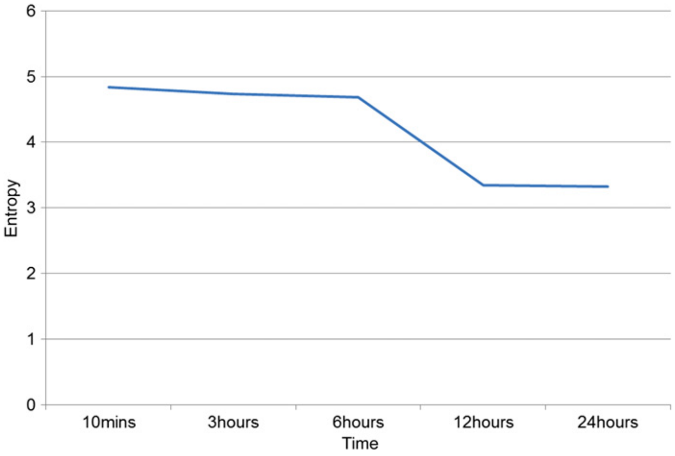


Fig. 6. Entropy according to party duration.

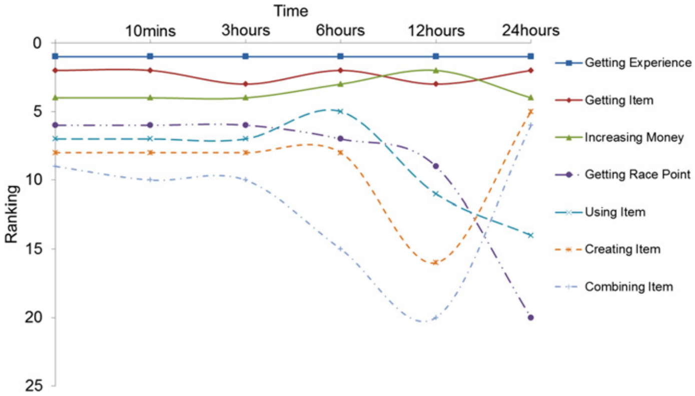


Fig. 7. The most frequent logs by normal user.

overall. These logs are recorded when gamers are hunting, which is the basic action in MMORPGs. In contrast, the rankings of 'Getting Race Point' log and 'Using Item' log are reduced in outliers. This implies that the outliers composed of game bots do not act to gain race points; since users can buy rare items and upgrade their positions with race points, it is natural that game bots, who do not find them useful, rarely work to earn them. In addition, 'Creating Item' and 'Combining Item' logs have different rankings in charts (e) and (f). This means that outlier groups are categorized into two groups based on their jobs. When users create items through harvesting or extracting, 'Creating Item' log is recorded. When users directly combine various materials to produce items, 'Combining Item' log is recorded. Game bots can use diverse methods such as hunting, harvesting, gathering materials, and crafting. The cheater who uses a game bot program can choose the job that determines the main action for achieving game money or items in the bot program. Although they have various options in selecting their jobs, for reasons of efficiency they commonly assign a particular job to the game bots. This results in some outliers harvesting, thus showing a high frequency of 'Creating Item' log, and others combining items, showing a high frequency of 'Combining Item' log. Fig. 8 illustrates the proportion of the most frequent logs by normal users according to party duration. The ratio of 'Getting Experience' log increases from 15% in normal parties to 35% in outlier parties. The ratio of 'Getting Item' log increases from 10% in normal parties up to 15% in outlier parties. This implies that game bots focus on getting game money and items while human players do not.

For the analysis of details of action logs, we included 6 action logs, namely discriminant classifiers. These are not major action logs, but are important in distinguishing between bots and normal parties because their rankings or ratios change drastically in outliers. We check the rankings of the other action logs that are discriminant classifiers according to party duration, as shown in Fig. 9.

The ranking of 'Quest Update' log drops from 10th in normal parties to below 40th. This means that completing quests is not performed frequently in outliers. Through completion of difficult quests or missions, normal parties upgrade their level and are qualified to install stigma, which can add special skills. We recognized that the ranking of 'Start Volplane' log in outliers falls below 30th. On the other hand, the ranking of 'Start Volplane' log in normal parties remains around 10th. When characters glide in a high altitude zone by using acceleration, the 'Start Volplane' log is recorded. While using the

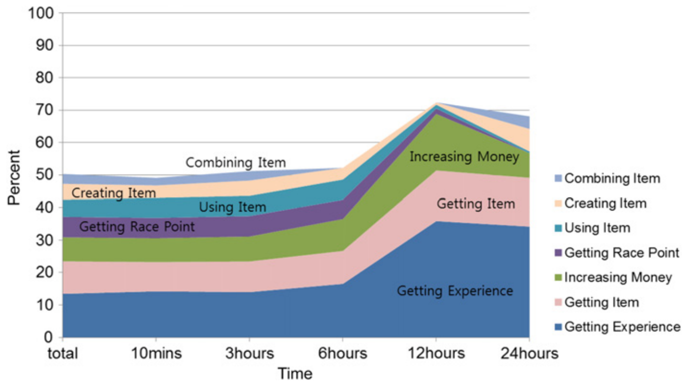


Fig. 8. The most frequent logs by normal user.

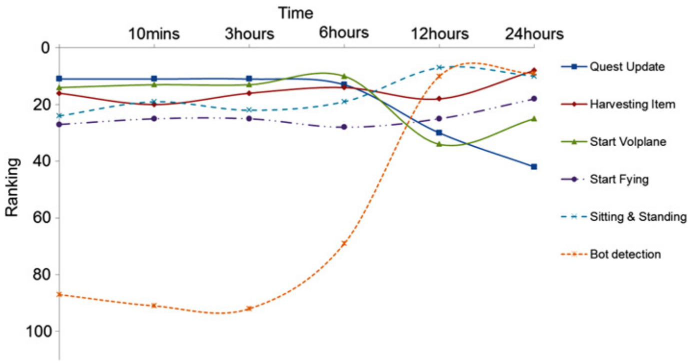


Fig. 9. The distinguishable logs at outlier user.

Table 6
Bot detection rule.

Rule-base
Getting experience log >= 34% and getting race point log <= 1.69% and sitting log <= top 10 and using item log <= 1.19% and quest completion log <= 0.16% and start volplane log >= top 34 and party member = 2 and party duration >= 600 s

volplane technique, normal characters want to move faster from a high altitude zone to a low altitude zone, whereas game bots hunt in flat places, so they use the running technique instead of the volplane technique.

The ranking of ‘Harvesting Item’ log is 10th in outliers that aim to sell materials gained from harvesting and 20th in normal parties. The rankings of ‘Sitting & Standing’ log certainly reflect the features of game bots. Game bots use the rest function of sitting and standing frequently to recover physical health and mental health. Game bots use the rest function periodically even in unnecessary situations. On the other hand, normal users also take a rest when the player’s power is low or there is no monster to hunt.

Through multi-step analysis on party-action logs, we identified significant classifiers that distinguish online game bots from normal users. We also built the rule-base by setting the threshold value of each significant classifier. The rule is shown in Table 6. We can detect hunting bots using the detection rule.

To evaluate the proposed framework, we compared the bot detection results from our rule base with internal monitoring rules and the banned account lists provided by the game company. As listed in Table 7, we identified 49 bots by applying our classifying rules among 52,377 party-play users. The first column indicates the number of users detected as bots from our rule base. The second column indicates the number of users who have detection code among our detected users. The third column indicates the number of users who are banned from the company among our detected users. The detection log is recorded by internal monitoring rules when users perform abnormal actions. Even though the internal rules leave the detection log on every single abnormal action, they fail in detecting game bots detected by our rule base. The accuracy rate of bot detection by the proposed framework is 95.92% (47/49).

In Fig. 10, the comparison results are displayed using a Venn diagram. The users detected by our rule base do not overlap completely with the ones detected by internal monitoring rules. Especially, the red area of the diagram represents the number of the bots that are not detected by the internal monitoring rules but detected by our rule-base and finally are

Table 7
Bot detection rate and accuracy rate comparison.

Proposed detection method	Bot detection log	Banned account list
49	26	47

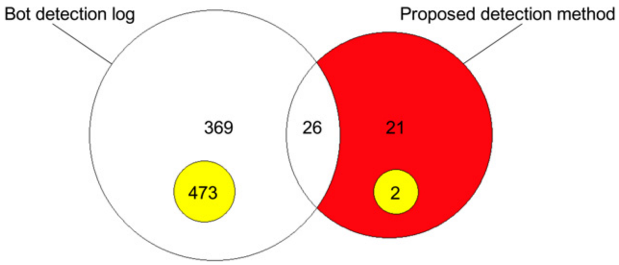


Fig. 10. Bot-classification results of the proposed framework and internal monitoring rules.

Table 8
Accuracy rate of bot detection.

Method	Accuracy rate
Party log (7 days)	95.92% (47/49)
VPN/PPTP log (30 days)	35.83% (929/2593)
Harvest log (1 day)	54.17% (13/24)
Chat log (9 days)	67.56% (25/37)
Trade log (30 days)	38.97% (265/680)

banned from the company. Our rule base increases the detection rate by detecting bots that are not detected by the internal rules, so it reduces the risk of false-negative detection error. The numbers in yellow in the diagram represent the numbers of users that are detected by the internal monitoring rules or our base but are not banned.

We performed bot detection using other data sources to compare the accuracy rate of each data source. The accuracy rates using the VPN login log, harvest log, chat log and trade log range between 36% and 68%. The party log-based detection method shows the highest accuracy rate. Table 8 lists the bot detection accuracy rates using various data sources.

5. Conclusion

In this paper, we proposed the user behavior analysis framework for online game bot detection. The proposed framework will enable game publishers to overcome the drawbacks of client side and network side detection methods, such as interruption during game play or collision with other software. To the best of our knowledge, our research is the first attempt to adopt party play to analyze user behavior at a group level. Previous studies on online game bot detection attempted to use user behaviors of simple actions such as moving or aiming. Game bots, automated player agents mainly play a game by forming a group with other bots, usually of two players; one helps and protects the other to gain an unfair advantage in a fast and easy way. Because the party plays of game bots have different goals from those of normal parties, their actions have different patterns from normal party players.

We applied the proposed framework to in-game logs of AION in NCsoft. Using a visual chart and statistical analysis, we could build a rule reasoner that contains significant classifiers and their threshold values and detects game bots based on the rule base. We evaluated the bot detection accuracy of users who are monitored by the internal rules of the company and the banned account lists from the company. The accuracy rate was up to 95.92% on the banned account list.

6. Future work

In our future study, we intend to incorporate the features of gamers' social networks formed by their party play into a more refined detection framework. We expect that it would improve the detection accuracy because, unlike humans, bots have almost no incentive to engage in a social network [24]. Moreover, we plan to investigate classification techniques, feature weighting [25] and feature-selection methods, which may be appropriate for bot detection tasks as well. Further, we expect to improve the bot detection accuracy rate through incorporating login IP data and other user behaviors, such as chatting, harvesting and trading.

Table 9
Log and term description.

Name	Description
Sitting	When the character sits down
Standing	When the character stands up
Getting experience	When the character gains experience points
Start flying	When the character flies
Getting race point	When the character gains race points
PvP	When the character combats with other characters
Increasing money	When the character gains game money
Instance dungeon entrance	When the character enters instance dungeon
Start volplane	When the character starts volplane
Creating item	When the character creates an item
Getting item	When the character gains an item
Harvesting item	When the character harvests an item
Using item	When the character uses an item
Combining item	When the character combines an item
Quest completion	When the character completes a quest
Bot detection log by the other security solution	When the character is detected as a bot by the other security solution
Race points	Race points can be earned through normal player versus player combat when defeating other player. Race points can be used to purchase various items from contribution vendors.
Instance dungeon	Instance dungeon is a special area that generates a new copy of the location for each group.
Volplane	The ability to glide is governed by momentum, so the higher the character is, the further the character will be able to go.
Harvesting (gathering)	Gathering such as ores, herbs, fruits, fish and other kinds of gatherable items supports crafting. Gatherable items are materials for crafting.
Combining (crafting)	Through crafting, many kinds of items can be created, many of which are not available for purchase from NPC vendors.

Acknowledgments

This research was supported by the Ministry of Knowledge Economy, Korea, under the "ITRC" support program supervised by the National IT Industry Promotion Agency (NIPA-2011-C1090-1001-0004).

Appendix

See Table 9.

References

[1] Z. Halim, A.R. Baig, H. Mujtaba, Measuring entertainment and automatic generation of entertaining games, International Journal of Information Technology, Communications and Convergence 1 (2010) 92–107.

[2] K.M. Woo, H.M. Kwon, H.C. Kim, C.K. Kim, H.K. Kim, What can free money tell us on the virtual black market, in: ACM SIGCOMM, Canada, 2011.

[3] H.K. Kim, Online Game Security, CODEGATE 2009, Korea, 2009. http://www.hksecurity.net/home/pds/codegate2-1%28huykang.kim%29.pdf.

[4] J. Hu, F. Zambetta, Security issues in massive online games, Security and Communication Networks 1 (2008) 83–92.

[5] K. Park, H. Lee, A taxonomy of online game security, Encyclopedia of Internet Technologies and Applications (2007) 606–611.

[6] R.V. Yampolskiy, V. Govindaraju, Embedded noninteractive continuous bot detection, Computers in Entertainment 5 (2008) 1–11.

[7] P. Golle, N. Ducheneaut, Preventing bots from playing online games, Computers in Entertainment 3 (2005) 3.

[8] K.T. Chen, J.W. Jiang, P. Huang, H.H. Chu, C.L. Lei, W.C. Chen, Identifying MMORPG bots: a traffic analysis approach, EURASIP Journal on Advances in Signal Processing, Spain (2009).

[9] S. Hilaire, H. Kim, C. Kim, How to deal with bot scum in MMORPGs? Communications Quality and Reliability, Canada (2010) 1–6.

[10] R. Thawonmas, Y. Kashifuji, K.T. Chen, Detection of MMORPG Bots Based on Behavior Analysis, ACM, USA, 2008, pp. 91–94.

[11] K.T. Chen, L.W. Hong, User Identification Based on Game-Play Activity Patterns, ACM, Australia, 2007, pp. 7–12.

[12] S. Yeung, J.C.S. Lui, J. Liu, J. Yan, Detecting cheaters for multiplayer games: theory, design and implementation, in: Networking Issues in Multimedia Entertainment, USA, 2006, pp. 1178–1182.

[13] M. Varvello, G.M. Voelker, Second life: a social network of humans and bots, in: Network and Operating Systems Support for Digital Audio and Video, The Netherlands, 2010, pp. 9–14.

[14] J. Yan, Bot, cyborg and automated turing test, Computer Science, England 5087 (2009) 190–197.

[15] M.A. Ahmad, B. Keegan, J. Srivastava, D. Williams, N. Contractor, Mining for gold farmers: Automatic detection of deviant players in MMOGs, Computational Science and Engineering, Canada 4 (2009) 340–345.

[16] M. van Kesteren, J. Langevoort, F. Grootjen, A step in the right direction: bot detection in MMORPGs using movement analysis, in: Artificial Intelligence, The Netherlands, 2009.

[17] S. Mitterhofer, C. Platzer, C. Kruegel, E. Kirda, Server-side bot detection in massively multiplayer online games, IEEE Security & Privacy 7 (2009) 29–36.

[18] R. Thawonmas, M. Kurashige, K.T. Chen, Detection of landmarks for clustering of online-game players, International Journal of Virtual Reality 6 (2007) 11–16.

[19] H. Kim, S. Hong, J. Kim, Detection of auto programs for MMORPGs, Computer Science 3809 (2005) 1281–1284.

[20] S. Gianvecchio, Z. Wu, M. Xie, H. Wang, Battle of botcraft: Fighting bots in online games with human observational proofs, in: Computer and Communications Security, USA, 2009, pp. 256–268.

[21] J.N. Kelly, Play time: an overview of the MMORPG genre, 2004. http://www.anthemion.org.

[22] Y. Ponomarchuk, D.W. Seo, Intrusion detection based on traffic analysis and fuzzy inference system in wireless sensor networks, Journal of Convergence 1 (2010) 35–42.
[23] S.S. Son, A.R. Kang, H.C. Kim, T.K. Kwon, J.Y. Park, H.K. Kim, Multi-Relational Social Networks in a Large-Scale MMORPG, ACM SIGCOMM, 2011.
[24] B. Buter, N. Dijkshoorn, D. Modolo, Q. Nguyen, S. van Noort, B. van de Poel, A.A. Salah, A.A.A. Salah, Explorative visualization and analysis of a social network for arts: the case of deviantART, Journal of Convergence 22 (2010) 87–94.
[25] Y. Ye, X. Li, B. Wu, Y. Li, A comparative study of feature weighting methods for document co-clustering, International Journal of Information Technology, Communications and Convergence 1 (2011) 206–220.