Received: 2 March 2022 | Revised: 8 July 2022 | Accepted: 15 August 2022

DOI: 10.4218/etrij.2022-0089

ORIGINAL ARTICLE

ETRI Journal WILEY

Quick and easy game bot detection based on action time interval estimation

Yong Goo Kang | **Huy Kang Kim**

School of Cybersecurity, Korea University, Seoul, Republic of Korea

Correspondence
Huy Kang Kim, School of Cybersecurity, Korea University, Seoul, Republic of Korea.
Email: cenda@korea.ac.kr

Funding information
Korea University Grant

Abstract

Game bots are illegal programs that facilitate account growth and goods acquisition through continuous and automatic play. Early detection is required to minimize the damage caused by evolving game bots. In this study, we propose a game bot detection method based on action time intervals (ATIs). We observe the actions of the bots in a game and identify the most frequently occurring actions. We extract the frequency, ATI average, and ATI standard deviation for each identified action, which is to used as machine learning features. Furthermore, we measure the performance using actual logs of the Aion game to verify the validity of the proposed method. The accuracy and precision of the proposed method are 97% and 100%, respectively. Results show that the game bots can be detected early because the proposed method performs well using only data from a single day, which shows similar performance with those proposed in a previous study using the same dataset. The detection performance of the model is maintained even after 2 months of training without any revision process.

KEYWORDS

consumer behavior, detection algorithms, machine learning, online game bot, predictive models

1 | INTRODUCTION

Online games have grown rapidly along with the development of Internet services. The spread of the coronavirus (COVID-19) pandemic has resulted in home quarantines, which have further driven users to online gaming [1, 2]. According to a report from global game analyst Newzoo, it is estimated that approximately 3.0 billion users worldwide play online games and the global game market was valued at approximately $175.8 billion in 2021 [3]. Statista, a global statistics company, predicted that the global traffic of online games would reach approximately 15 EB per month by 2022 [4].

Recent online game trends allow users to play games for free by offering more services and convenience through paid in-game items rather than paying for the games. This is called a free-to-play model. In most cases, players who use paid services are more likely to acquire higher-level items, and their in-game account levels increase quickly. However, those who do not use paid items find it difficult to catch up with the in-game account level of the paid users; thus, leveling up would require considerable time and effort to achieve. Particularly, accounts in a massively multiplayer online role-playing game (MMORPG) are designed with the ability to advance to a high level and acquire virtual goods after

several repetitive planned processes. Players who engage in fraudulent activities with illegal programs generate more advancements in their account level and obtain virtual items in a short time without making time-consuming efforts.

A game bot is an illegal program that automatically performs human actions in a game. With these programs, users can play games without breaks, acquire more items and experience than normal users, and quickly advance their accounts. These cyber-attacks have been further increased by the COVID-19 pandemic [5].

Furthermore, game bots ignore the rules of a game and can unfairly damage the gaming experience of other users. This behavior of game bots can cause players to feel disadvantaged and leave the game [6, 7]. The rampant use of game bots also causes financial damage to game companies. Players who use game bots negatively impact the expected revenue of the paid model because they do not have to pay for items. When the time designed by the developers to reach the highest level is shorter than the time expected, a user who reaches the maximum level leaves the game when there is no additional new content to enjoy. Additionally, increasing the monitoring of game bots increases the costs of game development. Gaming companies analyze the behavior and decision-making preferences of users to create individual user profiles. These profiles are used as a monetizing metric. Gaming bots are a major cause of digital disruption that distorts user profiles [8].

A method that can detect game bots quickly and accurately is crucial. The contributions of this study to game bot detection are as follows:

- We propose a method to detect game bots earlier using only data from a single day by demonstrating results similar to a previous study, which uses longer-term data.
- Our proposed model uses simple features based on ATIs and exhibits performance similar to models that use various user behavioral characteristics.
- We use long-term game logs and accurate data provided by actual game companies to identify game bots.
- We show that the proposed model is valid after two months of training, and the results of the detection model are reliable without any update for the model.

The remainder of this study is organized as follows: Section 2 presents basic terminologies for this study and some examples of existing research on game bot detection. Section 3 presents the proposed detection method for game bots based on ATI. Section 4 shows the experimental results of the proposed method with actual game data and performance evaluation. Section 5 presents the discussion, limitations, and future research directions. Finally, Section 6 presents the conclusions.

2 | BACKGROUND

2.1 | Terminology

In this section, we present the basic terminologies required to understand the problem of bot detection in online game services.

- **Non-player character (NPC):** An NPC is a character in a game that cannot be directly controlled by a player. NPCs provide players with various types of content, such as in-game quests.
- **Game log:** Game logs record all user actions that occur in a game. A log entry consists of a timestamp, action identifier (action ID), user ID, and additional relevant information. The timestamp is recorded in milliseconds. The action ID depends on the action type and has a unique value. This information is used to provide detailed contextual information about the action. For example, in the case of a log entry where a user gives money to another user, the additional information includes the ID of the receiver and the amount of money sent.
- **Game master:** Game master is someone who detects and penalizes game bot users based on their own rules.
- **ATI:** ATI is the time interval at which a particular action is repeated. Figure 1 shows the log recorded based on a user's action and an example of the ATI for each action. The unit of ATI in this study is seconds.
- **Normal user:** A normal user is a legitimate player of the game. Each user has various purposes and plays the game directly using the features and options provided by the game.
- **Game bot:** A game bot is a program that automatically performs the actions of a human playing a game.

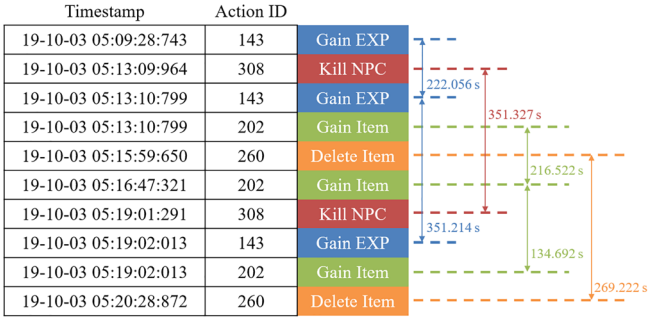


FIGURE 1 Illustration of user's action log and example of the action time interval

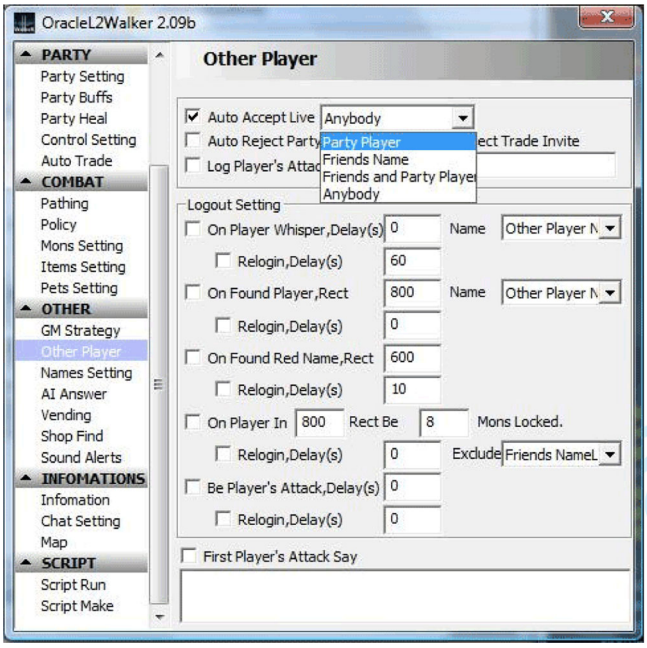


FIGURE 2 Example of the L2walker (bot for NCSOFT's Lineage 2 game) game bot program

Figure 2 shows an example of a game bot program, such as L2walker (bot for NCSOFT's Lineage 2 game). Bot programs provide functionality that users can use to set up behavioral routines for diverse purposes. Generally, players use bot programs to increase their level within a game more quickly or to gain in-game money and experience points (EXP).

2.2 | Literature review

A game company logs all player activities. We use a sample of these logs data to distinguish between normal and game bot users. The log-based online game bot detection methods studied so far can be divided into four categories: individual behavior, interaction behavior, social network, and network diffusion perspectives.

Thawonmas and others [9] detected game bots using a feature where bots repeat the same behavior compared with normal users. Chen and others [10], Mitterhofer and others [11] and van Kesteren and others [12] explored the fact that the movement path of game bots is more repetitive than that of normal players. Several mobile games have recently provided automatic play functions. Therefore, using the methods proposed in the aforementioned studies, there are cases where it is difficult to distinguish between game bots and normal users.

Chen and Hong [13] classified user activities in a 200-min period into idle and action, which is based on 20-min segments, to detect game bots. This method has limitations because it can misidentify users without motionless game bots or those who play for four hours or more. Lee and others [14] extracted users' action sequences to distinguish between game bots and normal users. Lee and others [15] detected game bots by parameterizing the log repeatability as self-similarity.

Ahmad and others [16] classified game bots into gatherers, bankers, dealers, and marketers. They analyzed the characteristics of each bot type and applied the detection method to EverQuest II. Kang and others [17] analyzed the party play logs to detect game bots. Chung and others [18] first classified individual and group play styles using clustering algorithms and then detected game bots for each cluster. This technique has a limitation, in which it must be preceded by learning using a sufficient amount of data for each cluster.

Keegan and others [19], Kwon and others [20], and Woo and others [21] used social network features to detect multiple game bots. This method can be applied to a new game with existing learning data, but it is difficult to apply it to a game without a collaborative play feature between users (i.e., a single-player online game)

Kang and others [22] detected game bots by analyzing chat messages among users, and Lee and others [23] proposed a method using the psychological characteristics of normal and game bot users.

Kang and others [24] detected game bots using a combination of player information, actions, group activities, social interaction information, and network information.

Tao and others [25] proposed a generalized game bot detection framework for MMORPGs called NetEase Games' Guard (NGUARD). NGUARD uses the methods of both clustering and classification from AI. Xu and others [26] proposed a generalized game bot detection framework for MMORPGs called NGUARD+. This framework employs a combination of supervised and unsupervised methods to detect game bots based on user behavior sequences. They proposed an auto-iteration mechanism to automatically adapt to mutated and new game bots, which is suitable for a rapidly changing online environment.

Li and others [28] introduced a transformer-style detection model called FingFormer. They focused on situations where there were no cheating data and performed a graph analysis of finger movements on a sensor. As this study uses finger input, it is uncertain whether auto-action, which plays automatically, can be applied to the most commonly used MMORPGs.

In this study, we demonstrate that the proposed method detects game bots with high performance using long-term data for 66 days. Furthermore, we show that the model trained using data from a single day maintains

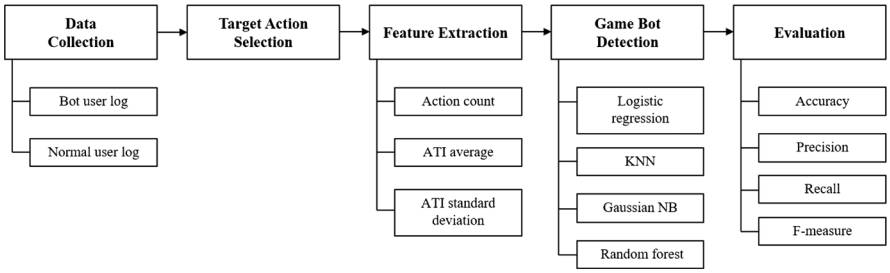


FIGURE 3 Overall process of the proposed game bot detection method

effective performance over time and that the results of game bot detection are reliable.

3 | PROPOSED METHOD

Figure 3 shows that our game bot detection method comprises five processes: data collection, target action selection, feature extraction, game bot detection, and evaluation.

3.1 | Data collection

All user actions in online games are stored in a log file or database on the server. Game masters use this log data to analyze user behavior and patterns. In this study, we present experiments using actual log data from Aion, which is one of the most common MMORPGs in the world (produced and serviced by NCSOFT [29]), to evaluate the proposed game bot detection method.

3.2 | Target action selection

The game bot repeatedly performs specific actions necessary to achieve its goal (e.g., making money, item production, or character level-up). We minimize other unnecessary actions to achieve more efficiency than a normal user playing fairly. The actions of the game bots are identified from the log and are selected as the target actions for analysis. Because the characteristics of game bots are different for each game and there are various types of game bots in one game, different game bots can be detected by selecting the appropriate target actions through the log analysis. In this study, the five actions that the bot performs the most are selected as target actions.

3.3 | Feature extraction

The time interval of each target action is extracted, and the average and standard deviations are calculated after selecting the target actions. The proposed method extracts 15 features. The data of the game bots and normal users comprise vectors.

3.3.1 | Calculating the ATI average

The ATI for an action with ActionID is defined as follows:

$$\mathrm{ATI}_i^{\mathrm{ActionID}} = T_i^{\mathrm{ActionID}} - T_{i-1}^{\mathrm{ActionID}}. \tag{1}$$

In (1), T_i^{ActionID} signifies the timestamp for ith action with ActionID. The ATI average for action with ActionID is defined as follows:

$$m^{\mathrm{ActionID}} = \frac{\sum_{i=1}^{N} \mathrm{ATI}_i^{\mathrm{ActionID}}}{N^{\mathrm{ActionID}}}, \tag{2}$$

where N^{ActionID} is the count of the ATI for action with ActionID.

3.3.2 | Calculating the ATI standard deviation

The ATI standard deviation for action with ActionID is defined as follows:

$$\sigma^{\mathrm{ActionID}} = \sqrt{\frac{\sum_{i=1}^{N} \left(\mathrm{ATI}_i^{\mathrm{ActionID}} - m^{\mathrm{ActionID}}\right)^2}{N^{\mathrm{ActionID}}}}. \tag{3}$$

3.3.3 | Calculating the action frequency

The frequency of action with ActionID is defined as $N^{\mathrm{ActionID}} + 1$.

3.4 | Game bot detection

The features discussed earlier are extracted from the game logs, and the classification technique of machine

learning is used to detect game bots. In this study, classification techniques, such as logistic regression, K-nearest neighbor (KNN), Gaussian Naïve Bayes (NB), and random forest, are applied.

3.5 | Evaluation

In this study, we evaluate the performance using tenfold cross-validation. We randomly mixed and divided the dataset in a ratio of 9:1 for training and testing, respectively. Furthermore, we randomly mix the dataset and partition it into 10 equal-sized subsets. Nine of the 10 subsets are used as training data, and the remaining are used as validation data for testing the model. Then, the cross-validation process is repeated 10 times, with each of the 10 subsets used exactly once as the validation data. The 10 results are averaged to produce a single measure.

Table 1 shows the number of cases that can be created using the actual answers and the results predicted by the machine learning model.

The result is a true positive (TP) if the correct answer is a game bot and the model predicts a game bot. A false negative (FN) occurs when the correct answer is a game bot and the model predicts a normal user. False positive (FP) and true negative (TN) occur when the correct answer is a normal user, and the model predicts a game bot and a normal user, respectively. TP and TN are correct predictions, whereas FN and FP are incorrect predictions. The accuracy of the model indicates the number of correct predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{4}$$

Generally, the performance of the model cannot be evaluated using only accuracy. When the number of accounts of normal users is overwhelmingly higher than the number of game bots, the accuracy of the model cannot be correctly evaluated. For example, when the data used for verification represents 970 normal users and 30 game bots, the accuracy is estimated to be as high as 97%, and the model unconditionally predicts a normal user. Therefore, precision (P), recall (R), and F-measure (F) must be used in addition to accuracy. P is the percentage of users predicted by the model to be game bots, who are actual game bots:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{5}$$

R is the percentage of correct predictions by the model that are actual game bots:

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

A higher P means that the model reliably predicts game bots. The higher the R value, the better the detection of the game bots. F is the harmonic mean value of P and R. It provides a measure of both indicators and can be calculated as follows:

$$F = \frac{P \times R}{P + R} \times 2. \tag{7}$$

In online games, minimizing the false positives, which mistake normal users for game bots rather than detect all existing game bots, is more important. Therefore, we evaluated the performance of the proposed model using the weighted F-measure (F_α) to assign weights and applied the F-measure to a weighted value rather than the recall itself. In this study, we assume that $\alpha = 0.9$.

$$F_\alpha = \frac{P \times R}{(1-\alpha) \times P + \alpha \times R}. \tag{8}$$

$$F_{0.9} = \frac{P \times R}{0.1 \times P + 0.9 \times R}. \tag{9}$$

For the proposed model, we measure the accuracy, precision, recall, and weighted F-measure of the prediction results.

TABLE 1 Number of cases that can be created using the actual answers and the results predicted by the machine learning model

		Prediction of model	
		Game bot	**Normal user**
Answer	Game Bot	True positive (TP)	False negative (FN)
	Normal User	False positive (FP)	True negative (TN)

4 | EXPERIMENTAL RESULTS

4.1 | Dataset and environment

Here, the log used for the analysis is data gathered over 66 days from Aion, which is about 1.2 TB. The total number of action types in this period was 162. Among these, the game company specified what it considered

game bots. During this period, 1466 and 31 643 bot and normal user accounts, respectively, were observed. We performed the experiments using a computer with an i7 2.9 GHz CPU, 16 GB RAM, 2 TB SSD, and Windows 10 platform.

We segmented the dataset daily to train and test the game bot detection model. Table 2 shows the distribution of the number of segments for each dataset period used in this study.

4.2 | Target action selection

Figure 4 shows the distributions for all activities during the data observation period. Table 3 shows the meaning of each action ID. The top five actions of the game bots represented 77% of all their actions, and the top five actions of a normal user constituted only 53% of the total. Particularly, we observed that the game bots mainly perform "gain EXP"(meaning gaining experiment points to achieve a higher level, which is represented as Action ID 143) and item-related activities (represented as 201, 202, 205, and 260). In contrast, normal users performed actions related to NPCs with a frequency similar to those of the item-related actions and "gain EXP." Thus, the game bot intensively performs particular actions for a specific purpose. Therefore, the performance of the game bot detection is expected to improved by tracking those actions that the game bot performs more frequently than the normal user. Furthermore, we selected 143 (gain EXP), 201 (create item), 202 (gain item), 205 (collect item), and 260 (delete item) as target actions.

4.3 | Feature extraction

We expect the ATI average and standard deviation of the game bots to be lower than that of normal users.

Figure 5 shows the boxplot of the ATI average and standard deviation for the game bots and normal user accounts. For actions that gain experience points, the average ATI and standard deviation for game bots were less than 140 s and 400 s, respectively. The average and standard deviation for normal users were greater than those for the game bots, and for the item-related actions, the ATI average and standard deviation of normal users were higher than those of game bots. The frequency of playing the game was low for normal users. Thus, the ATI average values of the game and standard deviation were high. Thus, normal users played intensively for a

TABLE 2 Distribution of the number of the segment for each dataset period

		Game bot	Normal user	Total
66 days	Training	15 968	437 924	453 892
	Test	1775	48 659	50 434
	Total	17 743	486 583	504 326
7 days	Training	2193	46 615	48 808
	Test	244	5180	5424
	Total	2437	51 795	54 232
1 day	Training	301	6448	6749
	Test	34	717	751
	Total	335	7165	7500

TABLE 3 Description of action IDs

Action ID	Description
143	Gain EXP
187	Increase kina (kina is the name of the currency in Aion)
201	Create item
202	Gain item
205	Collect item
260	Delete item
301	Spawn NPC
307	Drop NPC item
308	Kill NPC

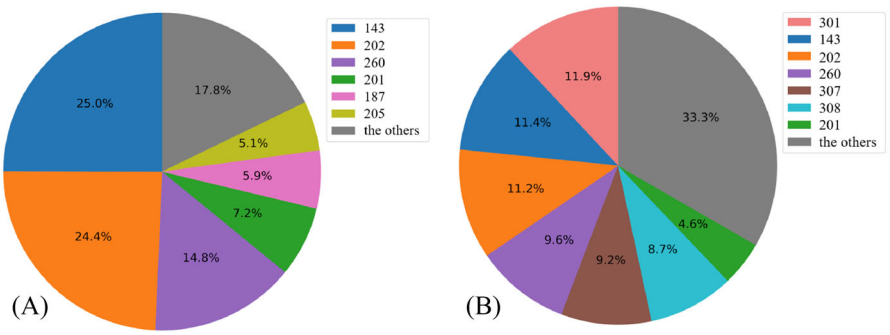


FIGURE 4 (A) Game bot and (B) Normal user's action distribution

certain period; however, their game play was short. This shows that the ATI characteristics of game bots and normal users are different. We created a model based on the frequency, ATI average, and ATI standard deviation of the five aforementioned actions.

4.4 | Game bot detection

Table 4 shows the prediction results for the models based on the period of the dataset used. Figure 6 shows the performance scores from classifying game bots and normal users by applying classification techniques for the following data: (a) over the entire period, (b) for 7 days, and (c) from a single day. In the case of (a), which comprises data for 66 days, we observed that the logistic regression, KNN, and Gaussian NB models showed low values of P, R, and $F_{0.9}$, respectively. We confirmed that the random forest model showed a P, R, and $F_{0.9}$ of 100%, 49.85%, and 90.86%, respectively. For the 7 day data, the accuracies of all four models were more than 90%, and the random forest model showed the highest value, at 97.4%.

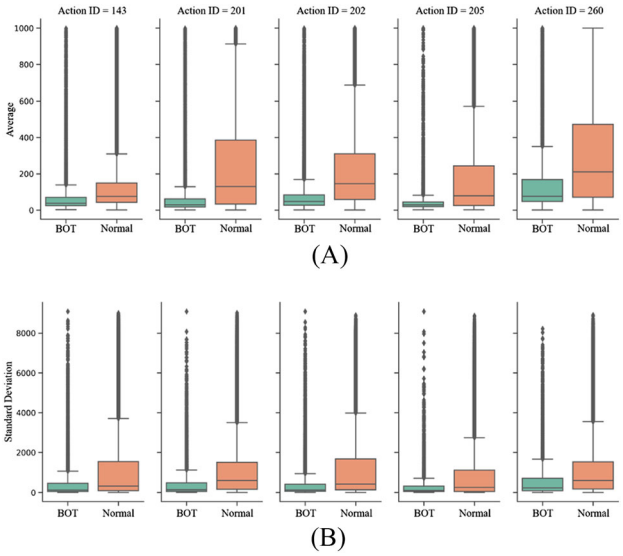


FIGURE 5 (A) Comparison of ATI average and (B) ATI standard deviation between bots and normal users for the selected actions

The logistic regression model showed 72.4% P, 8.6% R, and 41.6% $F_{0.9}$, indicating low R and $F_{0.9}$ values. The KNN model showed values of 74.1%, 43.4%, and 69.2% for P, R, and $F_{0.9}$, respectively. The Gaussian NB model showed low P, R, and $F_{0.9}$. Finally, the random forest model showed 100% P, 42.2% R, and 88% $F_{0.9}$, where R and P were low and high, respectively. Therefore, the random forest model was the most reliable for game bot detection.

The random forest technique is a well-known ensemble learning classification method. It overcomes the overfitting problem of the decision tree by constructing multiple decision trees, and it is also robust when training with an imbalanced dataset. Based on our segmented dataset, which consists of approximately 95% normal users and 5% game bots, the random forest model is expected to perform particularly well in that context, compared with other classification techniques under evaluation.

The random forest model showed the best performance when using data from a single day. The accuracy, P, R, and $F_{0.9}$ were 97.7%, 100%, 50%, and 90.9%, respectively. We observed that the performance of this model using the data from a single day was similar to that using the 7 and 66 day data. This implies that our proposed method can detect game bots using data from only a single day, which has similar performance to those shown using methods that use logs collected over a long period.

To confirm that our proposed method outperforms other existing methods, we compared it with that proposed in a previous study [24], using the same dataset. Figure 7 shows this comparison. The performance of our method is based on data from a single day and the random forest model. We observed that our method has better accuracy and P, lower R, and similar $F_{0.9}$ compared with the previous method. The high P means that our method for detecting game bots is more reliable than the previous study. The existing method, which uses several features, can be replaced with our proposed method as the accuracy and $F_{0.9}$ are similar. However, additional parameters must be used to improve the low R value in our method.

TABLE 4 Results of prediction for the models based on the period of the dataset used

	66 days				7 days				1 day			
Model	**TP**	**FN**	**FP**	**TN**	**TP**	**FN**	**FP**	**TN**	**TP**	**FN**	**FP**	**TN**
Logistic regression	65	1710	76	48 583	26	218	11	5169	7	27	0	717
KNN	755	1020	373	48 286	102	142	58	5122	10	24	4	713
Gaussian NB	738	1037	3376	45 292	110	134	360	4820	19	15	53	664
Random forest	725	1050	0	48 659	107	137	0	5180	17	17	0	717

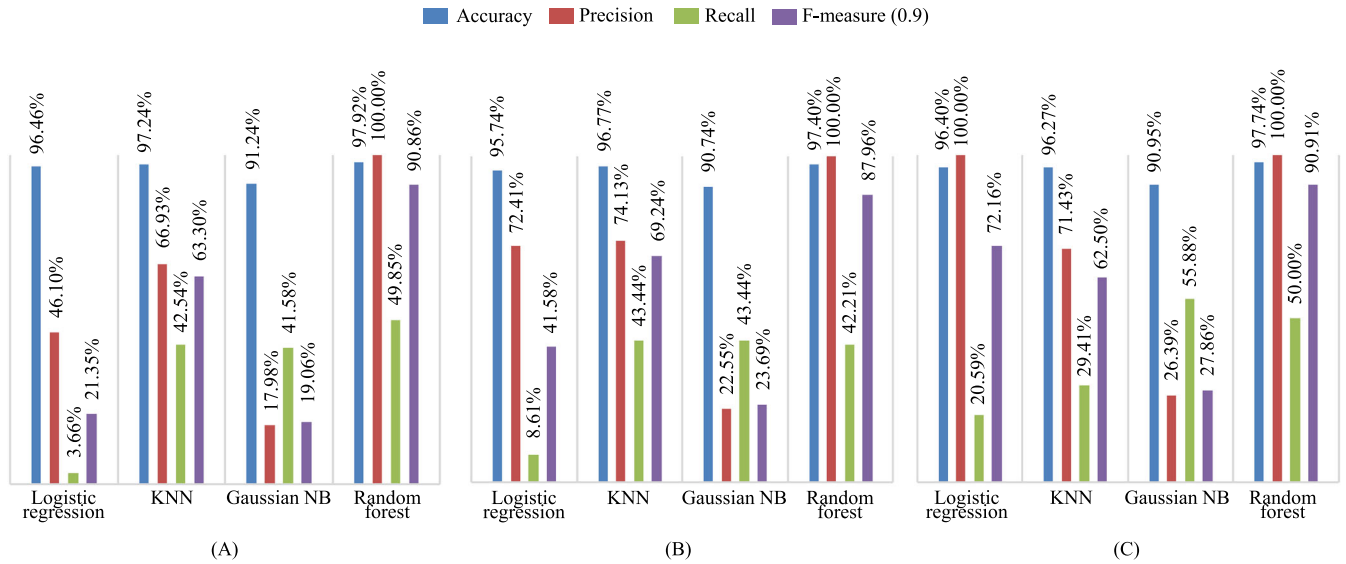


FIGURE 6 Performances according to the dataset period

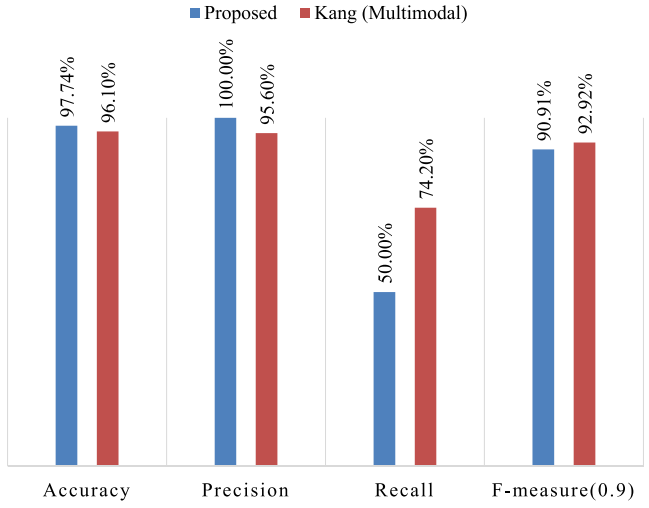


FIGURE 7 Performance comparison between the proposed method and a previous study

5 | DISCUSSION

5.1 | Detection time

We considered the time required for bot detection. It is difficult to apply the bot detection solution despite the performance of the proposed model when the time required for model prediction is long. We assumed that the feature values required for the model had already been calculated and saved to a file. Then, we measured the time starting with loading the feature files. Table 5 shows the training and testing time for the proposed models. We observed the training and testing time of the random forest model using the data from a single day at 0.0316 s and 0.0047 s, respectively. It was slower than the training time of the KNN and Gaussian NB, and it was also slower than the test time of logistic regression and Gaussian NB. However, it is sufficiently applicable to a real bot detection solution. When using the 66 day data, the training time of the logistic regression model increased by approximately 246 times (48.2519 s). The training and testing time of the KNN model increased by approximately 5675 (106.1266 s) and 413 times (16.1593 s), respectively. However, the training and testing time of the random forest model increased by approximately 104 (3.2925 s) and 26 times (0.1239 s), respectively. We observed that the random forest model is robust for games that log more data than the data from a single day used in our experiments.

5.2 | Model robustness

Lee and others [15] showed that the performance of the model deteriorated over time, thus showing that the model with the newest ground truth data must be retrained and updated. The detection model can become invalid because the playing pattern of game bots changes for various reasons. This implies that performance can decrease over time after the detection model has been trained. We used two random forest models to analyze how long our model was valid after training. One model was trained with the dataset for the first 7 days, and we measured the daily performance for the remaining 59 days of data. The other model was trained with the data from the first day, and we measured the daily performance for the remaining 65 days of data. Figure 8 shows

TABLE 5 Training and testing time for the proposed models

	66 days		7 days		1 day	
Model	**Training**	**Test**	**Training**	**Test**	**Training**	**Test**
Logistic regression	48.2519 s	0.1094 s	3.5210 s	0.0125 s	0.1959 s	0.0031 s
KNN	106.1266 s	16.1593 s	0.9076 s	0.6241 s	0.0187 s	0.0391 s
Gaussian NB	1.1438 s	0.1234 s	0.1343 s	0.0156 s	0.0156 s	0.0031 s
Random forest	3.2925 s	0.1239 s	0.3058 s	0.0194 s	0.0316 s	0.0047 s

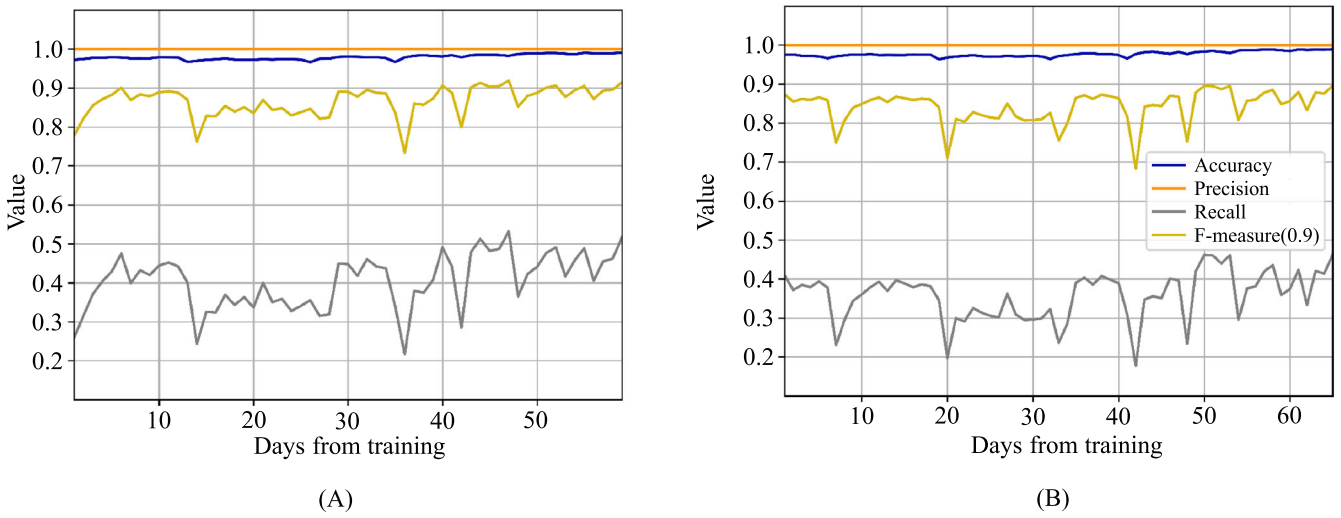


FIGURE 8 Variances of performance by the elapsed days from training: (A) trained model with 7 days data and (B) trained model with 1 day data

the variances of the performance by the elapsed days from the training of the two models. The results show that the precision of both models is maintained at 100% for the entire period. We observed that the models are valid after about two months of training, and the results of the detection model are reliable without further revision processes.

5.3 | Limitations and future work

Nevertheless, this study has several limitations. First, the generalizations of the model should be considered. Thus, we must ensure that models using only data from a single day can be applied to other online games. Different games have different designs of how users are satisfied and the actions they must take to achieve satisfaction. Therefore, there is no guarantee that the proposed method can be used in other games. However, the main purpose of using game bots is to take advantage of in-game gains with higher efficiency, rather than playing them directly. Therefore, we can expect that games will distinguish the ATIs between game bots and normal users. We can select the actions of the game bots used in a game during the target-action selection phase. Particularly, MMORPGs characteristics similar to those of Aion should be able to use our proposed models. Second, the recall rate of our proposed model is low, implying that the detection missed numerous positives. Methods to improve the recall rate should be studied in the future. Third, game bots can attempt to circumvent this detection once their developers understand the detection mechanism. The most intuitive attempt is to mimic the normal user's time interval as much as possible by generating a random delay between actions. The results are expected as follows: (a) The ATI has changed but still has a higher play efficiency than normal users, and (b) play efficiency is similar to or lower than normal users. In the former case, the proposed model with a low recall rate cannot effectively detect game bots. Accordingly, we must research the model maintenance framework to relearn and update the model using recent data. In the latter case, it is hard to determine whether game bots harm normal users; thus, we must first examine the need to detect game bots. Finally, direct performance analysis using existing studies is required. The online

game domain has limitations in releasing the dataset or acquiring a public dataset. We plan to compare performance by implementing existing game bot detection models, such as NGUARD [25, 26], and applying our dataset.

6 | CONCLUSIONS

We proposed a novel game bot detection method that has a high detection performance. The proposed model mainly considers how to select target actions and extract simple features. First, we selected five target actions based on the actions performed by users. Next, we extracted 15 features based on the time interval for the actions. Finally, we adapted four ML models to detect the game bot. We confirmed that the random forest model has the best performance with 97%, 100%, and 91% on the accuracy, precision, and F-measure, respectively. These results were similar to the previous study [24], which used the same dataset and showed similar performance even when training with data from a single day. The model maintained its performance after approximately 60 days. The findings of this prospective study will contribute meaningful insights to detect abnormal behavioral patterns and estimate their primary purpose and intent. Future research should focus on improving the recall rate, generalizing the model, and applying it to other games.

ACKNOWLEDGMENT

This study is supported by Korea University Grant.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ORCID

Yong Goo Kang https://orcid.org/0000-0002-2614-1915

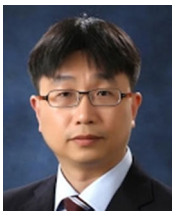
REFERENCES

1. D. L. King, P. H. Delfabbro, J. Billieux, and M. N. Potenza, *Problematic online gaming and the covid-19 pandemic*, J. Behav. Addict. **9** (2020), 184–186.
2. M. Á. López-Cabarcos, D. Ribeiro-Soriano, and J. Piñeiro-Chousa, *All that glitters is not gold. The rise of gaming in the covid-19 pandemic*, J. Innov. Knowl. **5** (2020), no. 4, 289–296. https://doi.org/10.1016/j.jik.2020.10.004
3. Newzoo, *Newzoo global games market report 2021*, Tech. report. newzoo, 2021. https://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2021-free-version. [Accessed 6 July 2022].
4. J. Clement: *Data volume of global consumer internet traffic from 2017 to 2022, by subsegment (in exabytes per month)*. Tech. report. Statista, 2018. [Accessed 6 July 2022]. https://www.statista.com/statistics/267194/forecast-of-internet-traffic-by-subsegment/
5. S. Venkatesha, K. R. Reddy, and B. R. Chandavarkar, *Social engineering attacks during the COVI-19 pandemic*, SN Comput. Sci. **2** (2021), no. 2, 1–9.
6. C. D'Anastasio, *Bot mafias have wreaked havoc in world of warcraft classic*, 2020. https://www.wired.com/story/world-of-warcraft-classic-russian-bots/. [Accessed 6 July 2022].
7. A. Fujita, H. Itsuki, and H. Matsubara, *Detecting real money traders in mmorpg by using trading network*, (Seventh Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford, CA, USA), 2011, pp. 26–31.
8. D. Bunker, *Who do you trust? The digital destruction of shared situational awareness and the covid-19 infodemic*, Int. J. Inform. Manag. **55** (2020), 102201. https://doi.org/10.1016/j.ijinfomgt.2020.102201
9. R. Thawonmas, Y. Kashifuji, and K.-T. Chen, *Detection of MMORPG bots based on behavior analysis*, (Proceedings of the International Conference on Advances in Computer Entertainment Technology, Yokohama, Japan), 2008, pp. 91–94.
10. K. T. Chen, A. Liao, H. K. K. Pao, and H. H. Chu, *Game bot detection based on avatar trajectory*, (International Conference on Entertainment Computing, Pittsburgh, PA, USA), 2008, pp. 94–105.
11. S. Mitterhofer, C. Kruegel, E. Kirda, and C. Platzer, *Server-side bot detection in massively multiplayer online games*, IEEE Secur. Privacy **7** (2009), no. 3, 29–36.
12. M. van Kesteren, J. Langevoort, and F. Grootjen, *A step in the right direction: Botdetection in mmorpgs using movement analysis*, (Proc. of the 21st Belgian-Dutch Conference on Artificial Intelligence, Eindhoven, Netherlands), 2009, pp. 129–136.
13. K.-T. Chen and L.-W. Hong, *User identification based on gameplay activity patterns*, (Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games, Melbourne, Australia), 2007, pp. 7–12.
14. J. Lee, J. Lim, W. Cho, and H. K. Kim, *I know what the BOTs did yesterday: Full action sequence analysis using naive Bayesian algorithm*, (12th Annual Workshop on Network and Systems Support for Games, Denver, CO, USA), 2013, pp. 1–2.
15. E. Lee, J. Woo, H. Kim, A. Mohaisen, and H. K. Kim, *You are a game bot!: Uncovering game bots in mmorpgs via self-similarity in the wild*, (NDSS, San Diego, CA, USA), 2016, pp. 1–15.
16. M. A. Ahmad, B. Keegan, J. Srivastava, D. Williams, and N. Contractor, *Mining for gold farmers: Automatic detection of deviant players in MMOGs*, (International Conference on Computational Science and Engineering, Vancouver), 2009, pp. 340–345.
17. A. R. Kang, J. Woo, J. Park, and H. K. Kim, *Online game bot detection based on party-play log analysis*, Comput. Math. Appl. **65** (2013), no. 9, 1384–1395.
18. Y. Chung, C. Y. Park, N. R. Kim, H. Cho, T. Yoon, H. Lee, and J. H. Lee, *Game bot detection approach based on behavior analysis and consideration of various play styles*, ETRI J. **35** (2013), no. 6, 1058–1067.
19. B. Keegan, M. A. Ahmed, D. Williams, J. Srivastava, and N. Contractor, *Dark gold: Statistical properties of clandestine networks in massively multiplayer online games*, (IEEE Second International Conference on Social Computing, Minneapolis, MN, USA) 2010, pp. 201–208.

20. H. Kwon, A. Mohaisen, J. Woo, Y. Kim, E. Lee, and H. K. Kim, *Crime scene reconstruction: Online gold farming network analysis*, IEEE Trans. Inform. Forensics Secur. **12** (2016), no. 3, 544–556.
21. J. Woo, A. R. Kang, and H. K. Kim, *The contagion of malicious behaviors in online games*, SIGACM SIGCOMM Comput. Commun. Rev. **43** (2013), no. 4, 543–544.
22. A. R. Kang, H. K. Kim, and J. Woo, *Chatting pattern based game bot detection: Do they talk like us?* KSII Trans. Int. Inform. Syst. **6** (2012), no. 11, 2866–2879.
23. J.-H. Lee, S. W. Kang, and H. K. Kim, *Detecting malicious behaviors in MMORPG by applying motivation theory*, J. Korea Game Soc. **15** (2015), no. 4, 69–78.
24. A. R. Kang, S. H. Jeong, A. Mohaisen, and H. K. Kim, *Multimodal game bot detection using user behavioral characteristics*, SpringerPlus **5** (2016), no. 1, 523.
25. J. Tao, J. Xu, L. Gong, Y. Li, C. Fan, and Z. Zhao, *NGUARD: A game bot detection framework for NetEase MMORPGs*, (Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK), 2018, pp. 811–820.
26. J. Xu, Y. Luo, J. Tao, C. Fan, Z. Zhao, and J. Lu, *Nguard+: An attention-based game bot detection framework via player behavior sequences*, ACM Trans. Knowl. Discov. Data **14** (2020), no. 6, 1–24.
27. W. Li, X. Chu, Y. Su, D. Yao, S. Zhao, R. Wu, S. Zhang, J. Tao, H. Deng, and J. Bi, *FingFormer: Contrastive graph-based finger operation transformer for unsupervised mobile game bot detection*, (Proceedings of the ACM Web Conference, Lyon, France), 2022, pp. 3367–3375. https://doi.org/10.1145/3485447.3512272
28. NCSOFT, *AION: The Tower of Eternity*, NCSOFT, Seoul, Republic of Korea, 2008. Game [PC].

AUTHOR BIOGRAPHIES

Yong Goo Kang received his BS and MS degrees in Computer Science and Engineering from Hanyang University, Republic of Korea, in 2009 and 2011, respectively. He is currently pursuing the PhD degree in Graduate School of Information Security from Korea University, Republic of Korea. He has been working at Samsung Electronics since 2011 and is currently a member of Samsung Research's Security Team. His current research interests are in data-driven security using machine learning.



Huy Kang Kim received his PhD in industrial and systems engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2009. He received an MS degree in industrial engineering from KAIST in 2000. He received a BS degree in industrial management from KAIST in 1998. He founded A3 Security Consulting, the first information security consulting company in South Korea in 1999. Also, he was a member and the last leader of KUS (KAIST UNIX Society), the legendary hacking group in South Korea. Currently, he is a professor in School of Cybersecurity, Korea University. His recent research is focused on detecting intrusions or abnormal activities in online games, automobile, and various payment systems. Before joining Korea University, he was a technical director (TD) and a head of information security department of NCSOFT (2004-2010), one of the most famous MMORPG companies in the world.

How to cite this article: Y. G. Kang and H. K. Kim, *Quick and easy game bot detection based on action time interval estimation*, ETRI Journal **45** (2023), 713–723. https://doi.org/10.4218/etrij.2022-0089