Cheating and Detection Method in Massively Multiplayer Online Role-Playing Game: Systematic Literature Review

MEE LAN HAN1, BYUNG IL KWAK2, AND HUY KANG KIM.3

1Department of AI Cyber Security, Korea University, Republic of Korea (e-mail: blosst@korea.ac.kr)
2School of Software, Hallym University, Republic of Korea (e-mail: kwacka12@hallym.ac.kr)
3School of Cybersecurity, Korea University, Republic of Korea (e-mail: cenda@korea.ac.kr)

Corresponding author: Huy Kang Kim (e-mail: cenda@korea.ac.kr).

ABSTRACT Recently, despite massively multiplayer online role-playing game (MMORPG) based on the PC implementation environment in mobile games, related fraudulent and illegal activities are still prevalent in response to the extension and diversity of the online gaming market. For game users who enjoy the game as leisure or cultural content, these issues inhibit the game’s immersion or cause seceding from the game, thus negatively affecting the overall online gaming industry. This systematic review provides an overview of substantial studies on the occurrence of cheating behavior, countermeasures, and detection methods. We aimed to scrutinize the scope of generable cheating behavior due to the structural characteristics of a stateful game server. Furthermore, we focused on providing the basis for the key mechanisms of online games based on the conceptual attributes of the MMORPG. A comprehensive literature research was conducted, and the classification of countermeasures and detection methods were systematized. The results of the systematic literature review contribute to the current understanding of how to correctly manage detection techniques and methods for cheating behavior in the MMORPG.

INDEX TERMS MMORPG, cheating, anomaly detection, countermeasure, detection method

I. INTRODUCTION

The cross-platform, that is, web/mobile application that enables users in massively multiplayer online role-playing game (MMORPG) based on the mobile and PC implementation environment to play a game in one place, has further expanded and diversified the online game market. Online games, particularly MMORPGs, provide all gamers with identical amounts of time and spatial resources [1]. However, these changes have gained momentum for some MMORPGs to cheat and spread on both mobile and PC game environments. Cheating deprives game users of the opportunity to gain fulfillment through a fair play. Thus, the impartiality and equitability of opportunities for game users has been lost. The prevalence of cheating confuses the principle of fair game play, which was originally sought in MMORPGs, and ultimately results in game users becoming indifferent to the game or leaving it [2]. Furthermore, cheating shortens the game’s shelf life, causing users to abandon the game, thus inflicting economic losses on developers and publishing companies [3]. Similar to the theory that broken windows cause the diffusion of crime [4], whether intentionally or not, when normal game users are frequently exposed to cheaters, cheating can quickly spread to the surroundings, and instigate each other to use them [5]. A game user who cheats may be an individual or a specific illegal game group. If these game users can purchase in-game resources with real money or game money, there may be a constant purchase demand for cheating. After the game user executes the cheating tool, the tool can withdraw the control of the client program.

Cheating includes the use of game bots, illegal private servers (unauthorized copying or production of source code for specific games), account theft, gold-farming groups, and real money trading. Figure 1 explains how the frequently occurring cheating in MMORPGs is organically structured. Game bots are common and wicked cheating tools in MMORPGs. Game users who use a game bot allow their game character to automatically work on gaming moves and techniques depending on the already established cheating matrix. A game bot enables a game user to play a game while awake for more than 48 h, which is physically impossible for
a normal person. Or a game bot enables a low-level game user (that is, high-level monsters that a low-level user is impossible to hunt) to acquire game money and items by defeating a high-level monster [6]. An MMORPG is designed to allow a game user to level up slowly and acquire various skills, equipment, and status depending on the level they have risen. Therefore, if a game user prefers to enter a map that only game users above a certain high level can visit, join a guild, or hunt a higher-level monster, the game user must put considerable time and training into it [7]. These game security solutions prevent hacking issues by detecting malware or cheating that performs memory manipulation, and then disabling it. However, it is not easy to thoroughly defend and purify the MMORPG online gaming environment with only one security solution. Depending on the structural and conceptual framework of each game, it should be possible to detect cheating patterns based on the game users’ play behavior. Table 1 reviews several commercial solutions for MMORPGs.

The remainder of this paper is organized as follows. Section II provides a detailed review of the structural characteristics and key mechanisms. In Section III, we describe various types of cheating behaviors considering the structural attributes of MMORPGs. Section IV presents countermeasures and detection methods for addressing this problem. In Section V, we recommend methods for combining preventive measures and external factors to improve the security and privacy of MMORPGs. Our limitations and conclusions are presented in Section VI.

II. STRUCTURAL FRAMEWORK

A. STATEFUL GAME SERVER

An MMORPG refers to a game in which many game users connect to the same server, move quickly in real time, and interact according to their roles. The most significant factor distinguishing real-time online games is whether other users are visible on the playscreen and if immediate interaction (actions, such as movement and battle) is possible between other users. Multiplayer online (MO)-type games that are not played on a large scale can be real-time online games. Because real-time online games are played by numerous users on each map, all clients continue (i.e., game users) playing while maintaining a connection to the server. Therefore, all the logic of the game, such as monster hunting, item sales, and quest progress, is directed by the game server. This type of game is called a real-time online game, and the game server for this purpose is called a stateful game server. Figure 2 describes systemically a structure and logic of the stateful game server mainly used in MMORPGs. A stateful game server has several characteristics. First, to transfer data related to the actions of all users in real time, all clients maintain a connection with the server. The connection concept is essential because the server should transfer game-related data to the client program quickly [15]. Second, the game content of the connected game users is stored in the game

| Solutions | Company / OS | Description |
|-----------|-------------|-------------|
| DexGuard  | Guard Square / Android | DexGuard operates code obfuscation to protect the critical binary data in an Android application. It prevents access to the source code and modification via reverse engineering and decompilation. Moreover, it supports the integrity of the mobile application. |
| Hackshield | AhnLabs / Window, Linux, Android | HackShield is a hacking and cheating prevention software suite and service for MMOs. (currently, Hackshield is no longer maintained.) |
| FraudForce | iOvation / Window, iOS, Android | FraudForce protects new account fraud, payment fraud, application fraud, and account theft to prevent fraud detection. |
| Kount ACCESS | Window / Window | Kount Access enables companies to identify fraudulent behavior in real-time, such as account theft, fraudulent account creation, account sharing detection, etc. |
| nProtect Game Guard | INCA Internet / Window | nProtect GameGuard generates a temporary module of attack interception when it detects an attack pattern. The solution can apply to a web control system. It uses an encoding function that encrypts execution files, such as EXE and DLL files. It has the detection function of processing modifications, and the prevention function of client hacking attacks. |
| XIGN CODE3 | Wellbia / Window, Linux, Android | XIGNCODE detects non-client game bots based on their execution code. It prevents hacking tools by utilizing the WIN32 API call patterns and frequencies. |
| X-Trap | Wiselogic / Window, Linux | X-Trap neutralizes attacks, and automatically blocks the unauthorized user who modified information and logged into the service server. |
server memory, and the logic is processed in real time. Status maintenance information about the game user and those surroundings, such as the game user’s character information, map information where the game user is currently located, and monster and NPC information currently on the map, is stored in the server in real time. Third, all the logic of the stateful game server is processed based on the data in the server’s memory. The game server reads all user data from the database and stores them in the memory of the game server. Subsequently, content logic is executed based on the memory of the game server. The modified data obtained by changing the logic owing to game play are saved back to the database. If the logic processing is completed in the game server memory, but the game server crashes before being saved to the database, the play data of the game users may be lost. Finally, a stateful game server distributes and expands the game server by adding a separate game world (a type of server concept) to accommodate more game users [16]. Most MMORPGs have similar structures. With one central server as the center, several game users play the game by communicating with the server in real time through their respective client programs connected to the Internet. The game client program can be downloaded from the server when the game is launched. Depending on the updated version, download is performed automatically or manually for each game patch. In an MMORPG, thousands of game users interact simultaneously and play by moving to a game map tailored to the level of game users. Similar to any computer program, the game client has state values that are the current values of all devices in the system, such as memory locations and registry values. In MMORPGs, although the server handles the most critical logic, some game-related state values are designed to be stored and processed in the client program [17]. However, even in some cases, storing game-related state values on the client side can pose a significant threat to security. In particular, an advanced-level attack targeting the game can be performed by manipulating the communication time state value between the client and server (i.e., game server, shop server).

B. CONCEPTUAL ATTRIBUTES OF MMORPG

a: Level-up

Most online games follow a structural and conceptual framework, which becomes increasingly difficult as one plays the game. At the initial game login, as the game designers provide a tutorial play guide to make gameplay easy for beginners to access, they encourage beginners to adapt to the online game without churning from the game. The purpose of game planning is to ensure users feel that they are improving on their own. Level-up plays an important role in these game narratives [18]. In MMORPGs, the only alternative to level up is to hunt monsters. Since the characters’ abilities are clearly different for each level, users have no choice but to focus on hunting monsters to level up. The level-up of the game character is an opportunity to hunt stronger monsters and to obtain more robust equipment as a kind of reward through hunting. Moreover, the reward items obtained by winning when the character level is high are more valuable than when the character level is low. A significant amount of game money enters the game user’s inventory when the item is converted to game money. Reaching above a certain character level requires a considerable amount of experience [19]. As the level increases, the game user will feel that the speed of leveling up significantly slows. Consequently, the
Illegal Private Server

A. GAME BOTS

A game bot is a program that can automatically play a gaming character, depending on pre-set functions before playing, without requiring direct control of the game user [6]. Game bots are classified into hardware and software classes depending on their form. A hardware-type game bot executes via a physical device embedded with malicious code, and is connected to the computer via USB or other methods. A software-type game bot is executed by installing a new program on a game user computer. Moreover, game bots are divided into in-game (IG) client bots and out-of-game (OOG) client bots depending on the operating mode [22]. The IG bot performs only a few automated game client functions. It does not change the code of the client program, but reads the graphics on the monitor via screen recognition. In addition, it controls the keyboard and mouse directly by reading the memory values of the game client. The OOG bot directly changes the code in RAM by analyzing used packets from communications between a game server and a client or circumventing the source code. An OOG typically uses long-term gaming services or servers that are neglected by security protocols.

B. ILLEGAL PRIVATE SERVER

An illegal private server refers to a server or game service that is used to provide the same or similar service after arbitrarily manufacturing a game server without obtaining a license or consent from a publisher or developer, tampering with, or distributing a game client. Illegal private servers have various names, such as Rogue Server, Emulation Server, and Illegal Server. Initially, people created a free server to enjoy games for free. However, they have now been transformed into systematic money making, such as game money exchange and in-game gambling content advertisements [23]. First, those who want to create an illegal private server intercept the information communicated between the server and client while using the online games normally. In many cases, information is checked through packet sniffing. The packet sniffing entails intercepting packets, which are the unit of messages delivered through the Internet, and then examining the content. When the information necessary to configure the illegal private server is collected to a certain extent, the server is created based on this information. The developers of the illegal private server developed a server and modulated the client simultaneously. A normal game client has the server’s address provided by the official service; however, the server’s address is changed to connect to an illegal private server. Illegal private server developers occasionally remove the device encrypted by the game developer to prevent tampering with the client’s content. Figure 3 depicts the Illegal private servers mechanism in MMORPGs. The illegal private server causes massive user churn and direct economic damage to game developers and publishing companies. Why do game users connect to an illegal private server rather than to an official service to play games? What are the benefits of illegal services for private servers? Game users who access illegal private servers can use items that can only be worn at high levels in the official service, regardless of their character level. Game users can freely use items after paying real money to the official service. Illegal private server operators require game users to donate server hosting or operating costs instead of allowing game users to play games freely. Game

III. CHEATING BEHAVIOR

A. GAME BOTS

A game bot is a program that can automatically play a gaming character, depending on pre-set functions before playing, without requiring direct control of the game user [6]. Game bots are classified into hardware and software classes depending on their form. A hardware-type game bot executes via a physical device embedded with malicious code, and is connected to the computer via USB or other methods. A software-type game bot is executed by installing a new program on a game user computer. Moreover, game bots are divided into in-game (IG) client bots and out-of-game (OOG) client bots depending on the operating mode [22]. The IG bot performs only a few automated game client functions. It does not change the code of the client program, but reads the graphics on the monitor via screen recognition. In addition, it controls the keyboard and mouse directly by reading the memory values of the game client. The OOG bot directly changes the code in RAM by analyzing used packets from communications between a game server and a client or circumventing the source code. An OOG typically uses long-term gaming services or servers that are neglected by security protocols.

Figure 3. Illegal private servers mechanism in MMORPGs.
users donate money to the operator in various ways (e.g., PayPal, cultural vouchers, etc.) [24], [25].

C. ACCOUNT THEFT
Account information stored in cache memory on a publicly accessible computer (e.g., public personal computer (PC) or PC with no password set) is easily exposed, and can be stolen. Their accounts can be exposed via several channels, including rogue APs, phishing websites, and malware [26], [27]. Account theft causes financial damage by applying a stolen ID and PW to several sites. Furthermore, it changes the PW to make the original account user inaccessible. Figure 4 shows the mechanism by which a game user’s MMORPGs and external service site login accounts are sequentially hijacked. In the online game, account theft implies that they have robbed a user’s account of its game money and items. They can obtain significant amounts of an unfair advantage by conducting cash transactions with a game user’s online information through account theft [28], [29]. In this case, the game company can intervene only if the victim reports the crime. Therefore, it can be time consuming to detect account theft, and it is even more difficult to identify reports. Both the game company and the victims have no choice but to spend considerable time and effort clarifying and repairing such damages.

D. GOLD FARMING GROUPS AND REAL MONEY TRADING
In popular MMORPGs, such as Aion [30], and World of Warcraft (WoW) [31], many game users manage game items at a significant high price with millions of users worldwide. If game users possess many resources or tools for game play, they can retrench the playing time to nurture game characters. Gold farming involves cheating to acquire game items by mobilizing inexpensive labor. Unfair game users typically use game bots to engage in gold farming. Many game bots were installed on each PC. A handful of managers continuously operate the game bots. It is relatively easy to detect large arrays of game bots because of their collective behavior. However, the behavior of small numbers of game bots is not easily detectable. Before MMORPGs were expanded to mobile games, China was well known for its gold-farming business in online games. It is commonly regarded as more profitable in the real world to earn money from playing games than to work in Chinese factories. Many companies have hired full-time employees to play games for more than 12 hours a day. This was a natural phenomenon that occurred in China, and was a severe social problem, especially in an environment where prison officers task prisoners to play games, such as slave labor [32], [33].

Real money trading (RMT) refers to the exchange of game items with real-world goods i.e. cash transactions. RMT is a rational economic activity owing to the differences in opportunity costs, and it serves a mediating role between the virtual world and real-world economies. However, RMT can also be used to mediate illegal transactions owing to the enhancement and diversification of cheating in online games [34], [35]. RMT has become a powerful motivation for gold farming and accounting theft. RMT is closely related to the gold farming industry. Gold farming groups operated by large-scale, organized, and segmented groups employ game bots for virtual mining. For RMT, the roles of stolen game accounts are classified according to their assigned roles in MMORPG [36], [37]. First, the gold farmer uses game bots to produce game items and resources on a large scale through economies of scale. Because all items mined on a
large scale are assigned an ID, it is easy to track whether an item has been used illegally in the game. However, game money does not follow the same criteria. Therefore, Merchant plays the role of exchanging a large-scale of mined game items with game money through the NPC shop. Finally, the banker serves as a safe store when orders are received from game users or when there is a system to maximize profits through cash transactions [38]. RMT causes a problem that aggravates the imbalance of the virtual economy in the game. Furthermore, it directly damages game users because it preempts a specific location to secure game money and items, and obstructs passage. The most critical problem with RMTs is that unauthorized cheating, such as game bots and account theft, is widespread [39]. Figure 5 presents in detail a structural role relationship between gold farming and RMT.

IV. DETECTION METHODOLOGY

A. COUNTERMEASURES

a: Countermeasures based on the structural framework

Most MMORPGs follow a similar structure, in which several game users connect around one server and communicate with the server in real time through each client’s program downloaded from the server. As shown in Table 2, the countermeasures of MMORPGs for detecting cheating can be explained by dividing the structural framework into three parts: client-side, network-side, and server-side.

First, on the client side, when a game user’s personal computer with the client program installed is infected with malicious code and exposed to threats, an algorithm to detect cheating may be performed. The window event log, including the game user’s keyboard and mouse input sequences, is used as a feature for some types of abnormality countermeasures. It is possible to create attributes that effectively classify game users who perform cheating and those who do not by extracting specified event sequences. The average interval between event occurrences, event occurrence rate, and event pattern converted into 26-dimensional vectors were also used as features to detect game bots by using the decision tree algorithm in MMORPGs [44]. Gianvecchio et al. studied attributes that effectively classify game users as either game bots or not through numerous event logs, such as keystrokes, mouse pointer arrow movement, mouse pointer clicking, and drag-and-drop motion from a game user. Game bots have shorter keystroke durations and more periodic patterns than normal game users do. The mouse pointer movement speed of the game bot increases linearly with the distance of the pointer. Game bots have shorter drag-and-drop durations than normal game users [41].

Thereafter, anomaly detection on the network side is accomplished by analyzing the network communication traffic occurring between the client program and server during game play. The most representative game bots among cheating can identify differences from normal game users by analyzing periodic game play patterns or rapid increases in transmission traffic. The decision tree algorithm is more effective in identifying the difference between a game bot and normal game users by considering the size and interval time of packets transmitted between client programs and a game server [46].

Finally, the stateful game server has a structural framework that records all the actions of game users from the moment they log in to the MMORPG until they log off. The log information stored in a stateful game server can only be

Figure 5. Structural role relationship between gold farming and RMT.
accessed by the person in charge of the game developer or publishing company. The log records login information that include when a game user accesses the login server, game play information, chat information, and party play information. Moreover, when the game user accesses the shop server and purchases a paid item with game money, the purchase information remains. As described above, the difference between normal game users and cheating, such as game bots, gold farming, and RMT, is based on the behavior of game users via the logs left in various forms [50]. By measuring the distribution of playing time, distribution of idle time, and entropy of a game play in the game, it is possible to identify whether the game user cheats [52], [53].

b: **Countermeasures based on cheating behavior**

The game user’s movement trajectory in the map, action sequence according to the trajectory, relationship setting with other game users, attributes of the game user’s item or game money transaction with NPCs or other users, and network traffic information sent to the server while playing the game can be considered behavior-based countermeasures. The trajectory of the game user on the map traces the path the game user moves while playing. In the case of normal game users, their trajectory on the map is irregular and complex, but in the case of an abnormal game user who only plays in a specific area of the map (i.e., a game user using a game bot), their trajectory on the map is slightly regular and simple. In the case of MMORPG games, quest-oriented games are pursued because of their structural and conceptual framework characteristics. Consequently, most game users have similar behavior sequences depending on the trajectory in the MMORPG map [54]. After logging in to the game until logging out, that is, after logging in and entering the initial map, an action sequence appears, such as talking to an NPC and receiving a quest, hunting monsters according to the quest, or visiting a shop. However, in the case of abnormal game users who cheat, they pursue game quest-oriented games, as well as their action sequences appear quite different from normal game users. In particular, the aspects of experience points, items, game money, and party playtime obtained from game users’ party play using game bots are extracted differently from those of normal game users. Therefore, these features are essential countermeasures for detect cheating [58].

As game users in MMORPG games advance to beginner, intermediate, and advanced levels, they form a highly interactive and complex social network with other game users. Chatting with game users currently logged into the same game server or the requesting party plays the game together. Additionally, it forms a social network in which game users can acquire game money and items through battles between game users using the players vs. players (PVP) system. However, as can be observed from the RMT cheating properties, it is possible to confirm simple and one-way transactions of items and game money between game users [41], [43]. When a game user plays a game, the packet information transmitted between the server and the client includes data, such as packet size, packet type, frequency of transmitted packets, and transmission timing. It is possible to detect game bots at the network level by collecting features related to network packets that occur depending on the game user’s behavior at the time, that is, when a game user moves from the current location to another location on the map, or when a game user attacks monsters, and then acquires an item or game money from them [48].

### B. Detection Techniques

The methodologies that can detect Cheating in MMORPGs can be divided into five categories: statistics analysis, data mining, similarity analysis, and network-based analysis, as shown in Table 3.

Cheating behavior detection based on statistical inference was conducted a lot in early research of MMORPG. The statistical technique is generally based on the measurement of variance, mean, standard deviation, distribution, p-value, and z-value from separately classified log data for the features applicable to all MMORPGs. Moreover, statistical techniques are an effective cheat detection method to detect suspicious patterns depending on the threshold values of some statistical features defined as a feature. The activity patterns and the idle time distribution of a game user better represent characteristics of game users when playing (e.g., hunting or mining) [53]. RMT players process a much more significant amount of game money than general users. Their activities are typically silent without participation, such as chatting, party play, guild, or shop trading in the community [34]. An advantage of the statistical technique is to detect cheating game users regardless of the game user’s playing period. This approach can adjust the unit of a dataset for analysis. Statistical methods enable analysis while setting a time window in the unit of a dataset and moving according to

| Detection Location | Related research | Merits and Demerits |
|--------------------|------------------|---------------------|
| Client-side        | [40]–[45]        | Merit: quick response for detection  |
|                    |                  | Demerit: collision with anti-virus mechanism |
| Network-side       | [46]–[49]        | Merit: independent scalable Resource-to-game characteristics |
|                    |                  | Demerit: low accuracy rate and high network traffic cost |
| Server-side        | [34], [36], [50]–[88] | Merit: high accuracy rate to detect the fraud and illegal activity; readily available log |
|                    |                  | Demerit: low universal validity because of the unique feature of game type |
Next, data mining in MMORPGs was used as a decision-making technique to correlate data and extract meaningful information by analyzing game behavior factors, such as game playing patterns, party play, chatting, and whisper logging during a game session MMORPG. Cheating such as a game bot has a different action sequence pattern than normal game users. Since the game bot among cheating behavior’s primary purpose is to more efficiently gain items (i.e., since it is related to the monopoly of hunting space and the problem of time allocation), its behavior pattern is repetitive and simple. By sequencing the behavior of a game user, it is possible to distinguish between the behavior of a game bot and that of a non-game bot [56]. The conclusion that, as compared to normal users, the game bot users more frequently pursue efficiency when hunting monsters, which is also reflected in the diversity of interaction with other game users and the frequency of chatting activities and volume. Briefly summarized indexes are required to deal with large-scale data in real-time in text-mining techniques. When other factors related to gameplay are combined, various kinds of cheating game users can be detected and identified with a higher accuracy [64]. In addition, the Latent Dirichlet Allocation (LDA) algorithm, which is frequently mentioned in recent natural language processing methodologies, detects cheating game users in MMORPGs. This algorithm is a type of generative probabilistic model. It can transform articles into a topic model. By converting the behavioral patterns of game users into feature vectors and applying them to machine learning algorithms, it is possible to classify them into several groups [82].

The similarity analysis technique in MMORPGs checks the presence or absence of cheating by analyzing the similarity between the game user’s current behavior and his/her previous behavior. In the case of a game bot, because only a predefined action is repeated, and the similarity pattern between the previous action and the current action of the corresponding game user appears to be almost identical. Put simply, simple behavioral patterns may appear during activities, such as hunting, chatting, trading, parties, and guild. Conversely, normal game users show fairly irregular behavioral patterns in activities, such as hunting, chatting, trading, parties, and guild. Mitterhofer et al. proposed a method to classify game bots by analyzing character movement patterns with similarity pattern-matching techniques on the server side. The combination of extracted waypoint information can classify game bots (i.e., the character’s saved starting point), and the longest common prefix (LCP) algorithm, which is mostly used for string comparisons [55]. Moreover, there is a method to detect a game bot depending on the battle behavior sequences of a game user compared to a similarity pattern-matching technique. The Levenshtein distance algorithm measures the number of changes required to change one string to another when comparing two strings [59]. Self-similarity among the similarity analysis technique includes features to show the similarity of game user actions as a function of time lag. Self-similarity considers statistically significant features in all games, rather than a single action such as a moving pattern. A self-similarity-based cheating detection method is more robust than a simple threshold-based cheating detection system. However, it is difficult to detect cheating game users who play only for a short period [36].

Network-based analysis can be used to produce a relational map of a network comprising of nodes and edges. A node is an endpoint in a network that can communicate with other nodes. The edges are lines connecting the nodes. There are two types of networks: directed and undirected. While a directed network has edges in the direction, an undirected network has unordered edges. These network-based analysis techniques is useful to detect roles and relationships between game users [39]. It is based on social interaction networks and monetary transactions. When normal game users interact within the game community, they show strong social connections and social engagement. In contrast, game users who mainly perform cheating behavior called gold farming by gold farmers and RMT have weak social connections as compared with normal game users [54]. Similarly, normal game users tend to connect with other game players who are more influential than themselves in their community as a guild or party play game form. In contrast, cheating game users tend to make connections without any such reckoning [62]. The network-based analysis technique is based on monetary transactions depending on the role of game users and the direction of the network. In particular, game users called gold farming by gold farmers and RMT consist of some types
of characters such as gold farmers, merchants, and banking characters. When looking at monetary transactions within the entire community, game users are divided into providers, merchants, bankers, and consumer or buyer types. Here, the consumer type is the non-RMT type [38], [78].

C. MACHINE LEARNING MODELS

This section describes how the machine learning algorithms are used in their respective research and what results can be derived from them: Decision Tree, Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbors (KNN), Bayesian network, AdaBoost, and Multilayer Perceptron (MLP). Table 4 details the characteristics, advantages, and disadvantages of machine-learning algorithms used to detect cheating in MMORPGs.

Decision tree algorithm is useful for supervised learning. The goal of this algorithm is to generate output value prediction models based on an input value. It has an overfitting problem, which cannot properly generalize a training dataset. This algorithm is based on a heuristic method that cannot guarantee an optimal decision tree [91]. However, it has a strength, making it easy to intuitively understand the results. Moreover, both numeric and nominal data can be applied to the algorithm. It can handle the sequencing of window event data related to changing character behaviors. Other features, such as mean, standard deviation, variation, and event patterns, are used with the training data for the decision tree algorithm [42]. Decision tree algorithm was used in identifying the difference between a game bot and normal game users by considering the size and interval-time of packets transmitted between client programs and a game server [47].

SVM algorithm is a binary linear classification model that can determine what category the given data falls into. While most machine learning algorithms train a model using all training data, SVM algorithms learn data by picking only support vectors that define decision boundaries. Therefore, it is very fast, as there is no need to learn all the data [92]. The MMORPG has mingled data characteristics that are difficult to classify linearly. This is so because the quests and game performance methods defined for each occupation of the game user are different, and the behavior pattern of each game user within the occupation group is also different [63]. SVM algorithm based on the game activity and chatting features is not sufficiently good to detect cheating game users among many game users. The F-score of SVM is often lower rather than other algorithms [93].

A Naïve Bayes classifier is a statistical classification method based on the Bayes theorem. It is a conditional probability-based classification method that calculates the probability that data belong to each class. It is a simple, fast, and efficient algorithm. When training data, the computation cost is small, because it works well regardless of the data size. Unlike other classification algorithms, Naïve Bayes requires having an independent condition between features of each other. It means that there is no correlation between them [94]. Furthermore, it is not ideal for datasets with many numerical features. Naïve Bayes classifier works well when the data mainly used for training come in. However, when an outlier that was not in the training data comes into the model, the likelihood becomes 0, and classification may not work correctly [68]. Since aggregate features or temporal features based on the particular character of a game user were not independent of each other, the accuracy was lower than in different algorithms for the extracted same features [61], [62].

KNN is an algorithm that classifies data based on the distances among the training data. This algorithm uses the Euclidean Distance and Kullback-Leibler divergence to calculate variables’ distances. In general, the advantage of this algorithm is that it is simple and easy to implement. However, unlike other machine learning algorithms, it does not create a model, so it is limited in understanding the relationship between features and classes. With an increase of the number of data, the number of computations increases, which slows down the class classification process [95]. In particular, since the Euclidean distance is not defined for nominal data, MMORPG, with many numeric data, increases the number of features that need to be converted into a numerical format. After converting features related to the action and chatting of game users to a numerical format, it can obtain the distance values [62], [96]. Although it is computationally expensive to calculate the distances among all data, it guarantees high accuracy [52].

Bayesian Network is a directed acyclic graph model known as a belief network. Bayesian networks graph and analyze causal relationships between random variables. In general, a graph can be expressed as a node and an edge. In a Bayesian network, nodes become random variables, and edges become stochastic relationships. It is necessary to reduce the feature data of extraneous variables, making it easy to explain mutual relationships [97], [98]. This algorithm has strengths in terms of interpretability because causal relationships are specified. However, all random variables included in the model and their causal relationships reflect the subjectivity of the modeler who designs it. Therefore, the model’s performance that describes the data can vary depending on the designed modeling [61], [68]. The classification accuracy was most improved when character attributes like race (human, orc, elf, etc.), demographic attributes, and state information of a specific country are mixed into one feature and applied to the Bayesian Network [68].

AdaBoost is a classifier that combines other simple learning algorithms. It uses the weak classifier, and the results from the algorithms become weights. It has a sensitive property in error data, but updates the weights iteratively during converged of specific values [99]. This algorithm has a low error rate and a fast computation speed. Previously, researchers have used sequence features based on behavior characteristics to detect the gold farming groups [68].

A neural network is an algorithm with one or more hidden layers between the input layer and the output layer. This algo-
rithm is also used to classify binary classes or multi classes by learning normal data and abnormal data. This model is a network structure as it fully connects the nodes between lower and higher layers. After the collected data are placed into input layers, the sum of weights is iteratively calculated. In the output layer, the result of the algorithm is classified [100]. Based on their three major cheating scenarios, several features are set at the neural network: the movement speed of a tree, the prediction error increase on the new data.

| Algorithm             | Related research | Formula and Characteristic | Merits and Demerits |
|-----------------------|------------------|----------------------------|---------------------|
| Decision Tree         | [42], [44], [52], [61], [62], [68], [70], [83] | $(x, Y) = (x_1, x_2, x_3, \ldots, x_k, Y)$ | Merit: it generates easy-to-understand rules in an if-then format; it can handle both continuous and categorical variables; it is a nonparametric method that does not require assumptions about the model. Demerit: in a regression model where the response variable is a numeric type, and the output variable is a continuous type, or the tree is too deep-rooted, predictability is poor; when there are many branches of a tree, the prediction error increase on the new data. |
| SVM                   | [51], [52], [60], [63], [70], [71], [87], [89] | $\frac{1}{2} \sum_j w_j^2 + c \sum_i \xi_i$ | Merit: interpreting the result is easy; there is not a lot of computation; low error rate. Demerit: it is only able to cope with the binary classification; sensitive to tuning parameters and kernel selection. |
| Naïve Bayes           | [51], [52], [56], [61], [67], [68], [88] | $p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)}$ | Merit: it is easy and fast to predict in multi-class classification; it is especially effective for categorical data, rather than numeric data; low computational complexity. Demerit: if the assumption of independence is not established or weak, errors may occur in the result; it is difficult to apply because there are not many completely independent situations. |
| KNN                   | [42], [52], [61], [62], [68] | Euclidean Distance Measure | Merit: the algorithm is simple and efficient, implementing the algorithm easy; performance is good in numeric data classification; no need to make estimates for data distribution. Demerit: if the amount of learning data is large, the classification speed is slow; if the size of the dimension is large, it takes a long time to calculate, and a large amount of memory is required. |
| Bayesian Network      | [52], [62], [68], [69], [90] | $p(x) = \prod_{v \in V} p(x_v|x_{pa(v)})$ | Merit: interpreting the result is possible. Demerit: if the number of nodes is large, it takes a long time to calculate; it is not numeric data but categorical data as an input value. |
| Adaboost              | [52], [61], [62], [68] | $f(x) = \sum_{t=1}^T a_t h_t(x)$ | Merit: low error rate and easy code Demerit: sensitive to outliers |
| Neural Network        | [41], [42], [70], [77], [79], [80], [87] | $\sum x_1 w_1 + x_2 w_2 + \ldots + x_n w_n$ | Merit: both qualitative and quantitative variables can be analyzed; a nonlinear combination between input variables is possible, high predictability; easy embodiment. Demerit: if the neural network is complicated, it takes a long time to calculate; it is difficult to interpret the result; the result is not constant because variables input randomly for analyzing. |
playing patterns, and asset [77]. Neural network algorithms have a black-box function. For that reason, although the performance of a detection system of cheating behavior based on a neural network is good, its disadvantage is that it is difficult to interpret and explain the detection rules.

V. RECOMMENDATIONS
As discussed in Section III, cheating affects the fair game play of other game users around the map, where cheating occurs by corrupting spontaneous and favorable participation. Furthermore, cheating cracks the game’s goals, rules, and system restrictions little by little, as defined within the game environment. In the following paragraphs, we summarize the tasks that a game provider should do to prevent cheating in advance and make a game more secure.

A. INSECURE INTERNAL ELEMENT ON THE MMORPG
Software bugs and flaws (e.g., API misuse during game development), problems with input validation (e.g., buffer overflows), SQL injection, and cross-site scripting have become major security threats to online gaming. Distributed systems and their use of multithreading at the game server result in common forms of time synchronization and state value bugs [102]. Hackers may attempt to embed unexpected behaviors in their switching logic, such as moving from one server to another, moving between continents, moving between gaming servers, and moving to a shop or an exchange server. If the game user logs off from the game or disconnects LAN services during their game movement, the status of the character, game item, and game money may behave differently from the defined game rules. This duplex bug is still rampant among the many MMORPGs. Repeated cheating of the server by the same account may unnecessarily increase the processing time of the server. It can act as a fraud platform for cheating to raise game items and money. Moreover, game users can exploit known bugs when planning a game map. The game map has a path (i.e., a convex line) that determines the movement of the NPC opponents. This path is invisible to the game user, but if a person identifies it unexpectedly, it can be used to create a safe zone and to estimate attack ranges against opponents. When a macro program that automatically hunts monsters is combined with cheating play, their various stats, such as health point (HP), magic point (MP), and level, can be raised significantly faster than the stats of other game users who do not use it [103], [104].

B. EXTERNAL FACTORS FOR DOING SECURITY MEASURE
There are three external elements of game security: registration, login, and authentication. First, a customer registers an account on the game portal site before becoming a game user. The game user can then access a game portal via a client program downloaded from the server after passing a secure HTTP, user authentication, email authentication, or mobile phone authentication. Game providers also recommend that users change their passwords frequently while heavily constraining password requirements for security. (i.e., Passwords generally comprise a combination of numbers, letters, and special characters of at least eight characters.) After the account is registered, the game user logs into the game portal and fulfills several security measures [105]. The security measures are as follows: 1) An online anti-key logger checks and protects the PC from key-loggers who leak sensitive information after intercepting the keyboard inputs. 2) A firewall service detects malware, such as hacking tools or viruses. 3) Malware inspection services detect intrusion and hacking. It also detects forbidden network access and illegitimate passwords, helping prevent information leaks and data corruption. Finally, the authentication process should use a combination of one-time passwords (OTP) and CAPTCHA to increase the difficulty of accessing the system. Game users cannot play the game before providing an authorization number sent via the mobile OTP to a mobile phone. A PC OTP can be run only on a PC with an installed game client. Personal information is protected from data leaks because a new authorization number is issued in every session. The game user can designate the PC from which an accessible game will be played, and register it legitimately. If a gaming session spawns from an unauthorized PC, the gaming service can disallow the session and account theft through game blocking. Moreover, a game provider should provide the gamer with a reporting mechanism against cheating.

VI. LIMITATIONS AND CONCLUSION
The current study has several limitations. From a structural framework point of view, research on game server structures other than the stateful game server structure has not been conducted. There are two main types of network configurations for online games: client/server (CS) and Peer to Peer (P2P). The network configuration method is determined by how the network is connected between players or who handles the game logic. The P2P method connects the game between players in progress. It is used in MO-type games, where a limited people play the game together (i.e., mostly about ten people). Because the player’s client processes the logic directly, there is minimal burden on the server, and fast action processing and accurate collision handling are possible. Next, this study considered the conceptual attributes of MMORPGs. As described above, MMORPG follows the CS method, but game genres other than MMORPGs, that is, FPS games, action games, racing, and sports games, follow the P2P method. Therefore, this systematic review may be complete and does not address other genres of games that follow a peer-to-peer approach.

This study provides extensive research on various occurrence cheating methods as well as methods for detecting MMORPGs. From a structural framework point of view, considering the stateful game server and conceptual attributes of MMORPGs, it explains in detail whether MMORPGs are structurally prone to cheating. Functional categories provide a structured overview of cheating behavior and detection.
methods based on stateful games. In addition, we have discussed the challenges and limitations of both the theoretical and practical viewpoints. We anticipate that the literature review will contribute to the current understanding of how to correctly manage their detection techniques and methods for cheating behavior in MMORPGs.

References

[1] T. Mancini, C. Imperato, and F. Sibilla, “Does avatar’s character and emotional bond expose to gaming addiction? two studies on virtual self-discrepancy, avatar identification and gaming addiction in massively multiplayer online role-playing game players,” Computers in Human Behavior, vol. 92, pp. 297–305, 2019.

[2] J. Tao, J. Xu, L. Gong, Y. Li, C. Fan, and Z. Zhao, “Nguard: A game bot detection framework for netease mmorpg,” in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 811–820.

[3] R. Sifa, A. Drachen, F. Block, S. Moon, A. Dubhashi, H. Xiao, Z. Li, D. Klabjan, and S. Demetriadis, “Archetypal analysis based anomaly detection for improved storytelling in multiplayer online battle arena games,” in 2021 Australasian Computer Science Week Multiconference, 2021, pp. 1–8.

[4] J. Q. Wilson and G. L. Kelling, “Broken windows,” Atlantic monthly, vol. 249, no. 3, pp. 29–38, 1982.

[5] J. Woo, A. R. Kang, and H. K. Kim, “The contagion of malicious behaviors in online games,” in ACM SIGCOMM Computer Communication Review, vol. 43, no. 4. ACM, 2013, pp. 543–544.

[6] J. Woo, S. W. Kang, H. K. Kim, and J. Park, “Contagion of cheating behaviors in online social networks,” IEEE Access, vol. 6, pp. 2908–29108, 2018.

[7] K. H. Kim and H. K. Kim, “Oldie is goodie: effective user retention by in-game promotion event analysis,” in Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, 2019, pp. 171–180.

[8] GuardSquare, “Dexguard,” available at https://www.guardsquare.com/en/dexguard.

[9] HackShield, “Online game security service hackshield,” available at https://company.ahlab.com/company/site/en/business/onlinesgame.jsp.

[10] Iovation, “Stop online fraud in real time,” whitepaper available at https://io.iamonow.com/content.iovation.com/resources/PDF/Iovation. fraud-prevention-solutions.pdf.

[11] Kount, “Online games and gaming fraud solution,” available at https://www.kount.com/industry-solutions/online-games-fraud-detection.

[12] INCAnet, “nprotect gameguard,” available at http://gameguard.nprotect.com/en/index.html.

[13] Wellthia, “Xigncode,” available at http://www.wellthia.com/home/en/pages/xigncode3/

[14] Wiselogic, “X-trap,” available at http://www.wiselogic.co.kr/index.htm.

[15] M. Suznjevic and M. Matijasevic, “Player behavior and traffic characterization for mmorpgs: a survey,” Multimedia systems, vol. 19, no. 3, pp. 199–220, 2013.

[16] M. Merabti and A. El Rhalibi, “Peer-to-peer architecture and protocol for a massively multiplayer online game,” in IEEE Global Telecommunications Conference Workshops, 2004. Globecom Workshops 2004. IEEE, 2004, pp. 519–528.

[17] A. Yahyavi and B. Kemne, “Peer-to-peer architectures for massively multiplayer online games: A survey,” ACM Computing Surveys (CSUR), vol. 46, no. 1, pp. 1–51, 2013.

[18] Y.-R. Shi and J.-L. Shih, “Game factors and game-based learning design model,” International Journal of Computer Technology, vol. 2015, 2015.

[19] Z. W. Lee, C. M. Cheung, and T. K. Chan, “Understanding massively multiplayer online role-playing game addiction: A hedonic management perspective,” Information Systems Journal, vol. 31, no. 1, pp. 33–61, 2021.

[20] D. Wemys, F. Cellina, E. Lobisger-Kägi, V. De Luca, and R. Castri, “Does it last! long-term impacts of an app-based behavior change intervention on household electricity savings in Switzerland,” Energy Research & Social Science, vol. 47, pp. 16–27, 2019.

[21] C. Phillips, D. Johnson, M. Klarkowski, M. J. White, and L. Hides, “The impact of rewards and trait reward responsiveness on player motivation,” in Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play, 2018, pp. 393–404.

[22] J. Wio and H. K. Kim, “Survey and research direction on online game security,” in Proceedings of the Workshop at SIGGRAPH Asia. ACM, 2012, pp. 19–25.

[23] E. Lee, B. Kim, S. Kang, B. Kang, Y. Jang, and H. K. Kim, “Profit optimizing churn prediction for long-term loyal customers in online games,” IEEE Transactions on Games, vol. 12, no. 1, pp. 41–53, 2018.

[24] D. Thomas and N. Bonnie, “A qualitative study of ragnarok online private servers in-game sociological issues,” The 5th International Conference on the Foundations of Digital Games, pp. 86–95 2010.

[25] R. Nick, “The closed world of private game servers,” available at http://news.bbc.co.uk/2/hi/technology/8397770.stm, 2009.

[26] B. Pal, T. Daniel, R. Chatterjee, and T. Ristenpart, “Beyond credential stuffing: Password similarity models using neural networks,” in 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019, pp. 417–434.

[27] K. C. Wang and M. K. Reiter, “Detecting stuffing of a {User’s} credentials at her own accounts,” in 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 2201–2218.

[28] “Use the same password for everything? you’re fuelling a surge in current account fraud,” available at http://www.telegraph.co.uk/finance/personalfinance/bank-accounts/12149022/Use-the-same-password-for-everything-Youre-fuelling-a-surge-in-current-account-fraud.html, 2016.

[29] “This is what can happen if you use the same passwords over and over,” available at http://www.telegraph.co.uk/money/consumer-affairs/can-happen-use-passwords/, 2017.

[30] NCSOFT, “Aion online: The official fantasy mmospg website,” available at http://na.aionline.com/en/.

[31] BlizzardEntertainment, “World of warcraft,” available at https://worldofwarcraft.com/en-us/.

[32] J. Dibbell, “The life of the chinese gold farmer,” The New York Times, vol. 17, 2007.

[33] V. Lehdonvirta, “Geographies of gold farming: New research on the third-party gaming services industry,” The Connectivity, Inclusion, and Inequality Group, http://cii. oii. ox. ac. uk/2014/10/29/geographies-of-gold-farming-new-research-on-the-third-party-gaming-services-industry/consulde le 7 juin 2016, 2014.

[34] H. Isuku, A. Takeuchi, A. Fujita, and H. Matsuura, “Exploiting mmospg log data toward efficient rmt player detection,” in Proceedings of the 7th international conference on advances in computer entertainment technology. ACM, 2010, pp. 118–119.

[35] B. Park and D. H. Lee, “The interplay between real money trade and narrative structure in massively multiplayer online role-playing games,” International Journal of Computer Games Technology, vol. 2017, 2017.

[36] 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.” in NDS, 2016.

[37] J. Tao, J. Lin, S. Zhang, S. Zhao, R. Wu, C. Fan, and P. Cui, “Mvan: Multi-view attention networks for real money trading detection in online games,” in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2536–2546.

[38] Y. Noh, S. Jeong, and H. K. Kim, “Trading behind-the-scene: Analysis of online gold farming network in auction house system,” IEEE Transactions on Games, 2021.

[39] E. Lee, J. Woo, H. Kim, and H. K. Kim, “No silk road for online gamers! using social network analysis to unveil black markets in online games,” in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1825–1834.

[40] R. McDaniel and R. V. Yampolskiy, “Development of embedded captacha elements for bot prevention in fisher random chess,” in International Journal of Computer Games Technology, vol. 2012, p. 2, 2012.

[41] S. Gianvecchio, Z. Wu, M. Xie, and H. Wang, “Battle of botcraft: fighting bots in online games with human observational proofs,” in Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009, pp. 256–268.

[42] H. Kim, S. Hong, and J. Kim, “Detection of auto programs for mmorpgs,” in Australasian Joint Conference on Artificial Intelligence. Springer, 2008, pp. 1281–1282.

[43] R. V. Yampolskiy and V. Govindaraju, “Embedded noninteractive continuous bot detection,” Computers in Entertainment (CIE), vol. 5, no. 4, p. 7, 2008.
VOLUME 4, 2016

J. Lee, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen, “Identifying mmosp bots: A traffic analysis approach,” EURASIP Journal on Advances in Signal Processing, vol. 2009, no. 1, p. 797159, 2008.

S. Hilaire, H.-c. Kim, and C.-k. Kim, “How to deal with bot scum in mmosp?” in Communications Quality and Reliability (CQR), 2010 IEEE International Workshop Technical Committee on. IEEE, 2010, pp. 1–6.

A. Cornelissen and F. Grootjen, “A modern turing test: Bot detection in mmosps.”

J. Christensen, M. Cusick, A. Villanes, O. Veryovka, B. Watson, and A. Cornelissen and F. Grootjen, “A modern turing test: Bot detection in massively multiplayer online games,” IEEE Security & Privacy, vol. 7, no. 5, pp. 38–44, 2009.

M. A. Ahmad, B. Keegan, A. Roy, D. Williams, J. Srivastava, and N. Contractor, “Win, lose or cheat: The analytics of player behaviors in online games,” North Carolina State University. Dept of Computer Science, 2017.

Y. Mishima, K. Fukuda, and H. Esaki, “An analysis of players and bots behaviors in mmosp,” in Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on. IEEE, 2013, pp. 870–876.

A. Fujita, H. Itsuki, and H. Matsubara, “Detecting real money traders in mmosp by using trading network.” in AIIDE, 2011.

M. A. Ahmad, B. Keegan, S. Sullivan, D. William, J. Srivastava, and N. Contractor, “Illicit bits: Detecting and analyzing contraband networks in massively multiplayer online games,” in Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 2011, pp. 127–134.

M. Varvello and G. M. Voelker, “Second life: a social network of humans and bots,” in Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video. ACM, 2010, pp. 9–14.

S. Mitterhofer, C. Kruegel, E. Kirda, and C. Platzer, “Server-side bot detection in massively multiplayer online games,” IEEE Security & Privacy, vol. 7, no. 5, pp. 38–44, 2009.

J. Lee, J. Lim, W. Cho, and H. K. Kim, “In-game action sequence analysis for game bot detection on the big data analysis platform,” in Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems-Volume 2. Springer, 2015, pp. 403–414.

J. Oh, Z. H. Borbor and J. Srivastava, “Automatic detection of compromised accounts in mmosps,” in Social Informatics (SocialInformatics), 2012 International Conference on. IEEE, 2012, pp. 222–227.

A. R. Kang, J. Woo, J. Park, and H. K. Kim, “Online game bot detection based on party-play log analysis,” Computers & Mathematics with Applications, vol. 65, no. 9, pp. 1384–1395, 2013.

C. Platzer, “Sequence-based bot detection in massive multiplayer online games,” in Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on. IEEE, 2011, pp. 1–5.

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 Journal, vol. 35, no. 6, pp. 1058–1067, 2013.

M. A. Ahmad, B. Keegan, A. Roy, D. Williams, J. Srivastava, and N. Contractor, “Guilt by association? network based propagation approaches for gold farmer detection,” in Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on. IEEE, 2013, pp. 121–126.

J. Oh, Z. H. Borbor, D. Sharma, and J. Srivastava, “Bot detection based on social interactions in mmosps,” in Social Computing (SocialCom), 2013 International Conference On. IEEE, 2013, pp. 536–543.

R. Thawonmas, Y. Kashifjui, and K.-T. Chen, “Detection of mmosp bots based on behavior analysis,” in Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology. ACM, 2008, pp. 91–94.

A. R. Kang, H. K. Kim, and J. Woo, “Chatting pattern based game bot detection: do they talk like us?” KSI Transactions on Internet & Information Systems, vol. 6, no. 11, 2012.

H. Kwon, K. Woo, H.-c. Kim, C.-k. Kim, and H. K. Kim, “Surgical strike: A novel approach to minimize collateral damage to game bot detection,” in Proceedings of annual workshop on network and systems support for games. IEEE Press, 2013, pp. 3–13.

S. H. Park, J.-H. Lee, H.-W. Jung, and S.-W. Bang, “Game behavior pattern modeling for game bots detection in mmosp,” in Proceedings of the 4th international conference on ubiquitous information management and communication. ACM, 2010, p. 33.

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,” in Proceedings of annual workshop on network and systems support for games. IEEE Press, 2013, pp. 1–2.

M. A. Ahmad, B. Keegan, J. Srivastava, D. Williams, and N. Contractor, “Mining for gold farmers: Automatic detection of deviant players in mmosp,” in Computational Science and Engineering, 2009. CSE’09. International Conference on, vol. 4. IEEE, 2009, pp. 340–345.

S. Yeung, J. C. Lui, J. Liu, and J. Yan, “Detecting cheaters for multiplayer games: theory, design and implementation,” in Proc IEEE CCON, vol. 6, 2006, pp. 1178–1182.

J. Woo, H. J. Choi, and H. K. Kim, “An automatic and proactive identity theft detection model in mmosps,” Applied Mathematics & Information Sciences, vol. 6, no. 3, p. 291, 2012.

H.-K. Pao, H.-Y. Lin, K.-T. Chen, and J. Faadil, “Trajectory based behavior analysis for user verification,” Intelligent Data Engineering and Automated Learning-IDEAL 2010, pp. 316–323, 2010.

K. Woo, H. Kwon, H. K. Kim, and H. K. Kim, “What can free money tell us on the virtual black market?” ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, pp. 392–393, 2011.

H. Kwon and H. K. Kim, “Self-similarity based bot detection system in mmosp,” in Proceedings of the 3rd international conference on internet, 2011, pp. 477–481.

H. J. Choi, J. Y. Woo, and H. K. Kim, “Detecting account thefts on the server-side by analyzing game log in mmosp,” in Proceedings of the 9th International Conference on Internet, 2011, pp. 501–506.

D. Seo and H. K. Kim, “Detecting gold-farmers’ groups in mmosp by connection information,” in Proceedings of the 3th international conference, 2011, pp. 583–588.

M. van Kesteren, J. Langevoort, and F. Grootjen, “A step in the right direction: Bot detection in mmosps using movement analysis,” in Proc. of the 21st Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2009), 2009, pp. 129–136.

J. Lee, S. W. Kang, and H. K. Kim, “A study on hard-core users and bots detection using classification of game character’s growth type in online games,” Journal of the Korea Institute of Information Security and Cryptology, vol. 25, no. 5, pp. 1077–1084, 2015.

H. Kwon, A. Mohaisen, J. Woo, Y. Kim, E. Lee, and H. K. Kim, “Crime scene reconstruction: Online gold farming network analysis,” IEEE transactions on Information Forensics and Security, vol. 12, no. 3, pp. 544–556, 2017.

H. Kwon, S. Yang, and H. K. Kim, “Crime scene re-investigation: A postmortem analysis of game account stealers’ behaviors,” arXiv preprint arXiv:1705.00242, 2017.

M. L. Bernardi, M. Cimitile, F. Martinelli, and F. Merecaldo, “A time series classification approach to game bot detection,” in Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. ACM, 2017, p. 6.

D. Liu, X. Gao, M. Zhang, H. Wang, and A. Stavrou, “Detecting passive cheats in online games via performance-skilfulness inconsistency,” in Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on. IEEE, 2017, pp. 615–626.

Z. Zhang, H. Anada, J. Kawamoto, and K. Sakurai, “Detection of illegal players in massively multiplayer online role playing game by classification algorithms,” in Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference On. IEEE, 2015, pp. 406–413.

A. R. Kang, S. H. Jeong, A. Mohaisen, and H. K. Kim, “Multimodal game bot detection using user behavioral characteristics,” SpringerPlus, vol. 5, no. 1, p. 523, 2016.

S. W. Kang, J. Lee, J. Yoon, and H. K. Kim, “A study of rmt buyer detection for the collapse ofigg in mmosp,” Journal of the Korea Institute of Information Security and Cryptology, vol. 25, no. 4, pp. 849–861, 2015.

S. H. Jeong, A. R. Kang, and H. K. Kim, “Analysis of game bots’ behavioral characteristics in social interaction networks of mmosp,”
in ACM SIGCOMM Computer Communication Review, vol. 45, no. 4. ACM, 2015, pp. 99–100.

[86] J.-h. Lee, S. W. Kang, and H. K. Kim, “Detecting malicious behaviors in mmorpg by applying motivation theory,” Journal of Korea Game Society, vol. 15, no. 4, pp. 69–78, 2015.

[87] H. M. Song and H. K. Kim, “Game-bot detection based on clustering of asset-varied location coordinates,” Journal of the Korean Institute of Information Security and Cryptology, vol. 25, no. 5, pp. 1131–1141, 2015.

[88] S.-H. Park, H.-W. Jung, T.-B. Yoon, and J.-H. Lee, “Behavior pattern modeling based game bot detection,” Journal of Korean Institute of Intelligent Systems, vol. 20, no. 3, pp. 422–427, 2010.

[89] H. Kim and H. K. Kim, “Research on online game bot guild detection method,” Journal of the Korean Institute of Information Security and Cryptology, vol. 25, no. 5, pp. 1115–1122, 2015.

[90] A. Roy, M. A. Ahmad, C. Sarkar, B. Keegan, and J. Srivastava, “The ones that got away: False negative estimation based approaches for gold farmer detection,” in Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom). IEEE, 2012, pp. 328–337.

[91] J. Gareth, W. Daniela, H. Trevor, and T. Robert, “An introduction to statistical learning with applications in r.” Springer.

[92] E. Scholkopf, A. J. Smola, F. Bach et al., Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[93] P. V. Dinh, T. N. Nguyen, and Q. U. Nguyen, “An empirical study of anomaly detection in online games,” in 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS). IEEE, 2016, pp. 171–176.

[94] H. Zhang, “The optimality of naive bayes,” 2004.

[95] N. S. Altman, “An introduction to kernel and nearest-neighbor non-parametric regression,” The American Statistician, vol. 46, pp. 175–185, 1992.

[96] K.-T. Chen, A. Liao, H.-K. K. Pao, and H.-H. Chu, “Game bot detection based on avatar trajectory,” in International Conference on Entertainment Computing. Springer, 2008, pp. 94–105.

[97] X. San, J. Dai, P. Liu, A. Singhal, and J. Yen, “Using bayesian networks for probabilistic identification of zero-day attack paths,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 10, pp. 2506–2521, 2018.

[98] W. Alhakami, A. Alharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, “Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection,” IEEE Access, vol. 7, pp. 52 181–52 190, 2019.

[99] R. E. Schapire, “Explaining adaboost,” Empirical inference, pp. 37–52, 2013.

[100] C. M. Bishop, “Neural networks for pattern recognition,” OXFORD University Press.

[101] K. Prasetya et al., “Artificial neural network for bot detection system in mmorpg,” in 2010 9th Annual Workshop on Network and Systems Support for Games. IEEE, 2010, pp. 1–2.

[102] G. McGraw, Software security: building security in. Addison-Wesley Professional, 2006, vol. 1.

[103] G. McGraw and C. CTO, Exploiting online games: cheating massively distributed systems. Addison-Wesley, 2008.

[104] N. Cano, Game Hacking: Developing Autonomous Bots for Online Games. No Starch Press, 2016.

[105] A. B. Jeng and C. L. Lee, “A study on online game cheating and the effective defense,” in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, 2013, pp. 518–527.

MEE LAN HAN received her Ph.D. degree in Information Security from School of Cybersecurity, Korea University, Seoul, South Korea, in 2021. She received her B.S. degree in Computer and Science from Sejong University, in 2012. Currently, she is an assistant professor in School of Software, Hallym University. Prior to joining Hallym University, she has worked as a research professor in School of Cybersecurity, Korea University, in 2021. Her research interests include vehicle security, network security, machine learning, and deep learning.

BYUNG IL KWAK received his Ph.D. in Information Security from School of Cybersecurity, Korea University, Seoul, South Korea, in 2021. He received a B.S. degree in Computer and Science from Sejong University, in 2012. Currently, he is an assistant professor in School of Software, Hallym University. Prior to joining Hallym University, he has worked as a research professor in School of Cybersecurity, Korea University, in 2021. His research interests include vehicle security, network security, machine learning, and deep learning.

HUY KANG KIM received his Ph.D. in industrial and systems engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2009. He received an M.S. degree in industrial engineering from KAIST in 2000. He received a B.S. 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 K1IS (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 solving many security problems in online games based on the user behavior analysis. 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.

***