A Time Series Classification Approach to Game Bot Detection*

Mario Luca Bernardi
Giustino Fortunato University
Benevento, Italy
m.bernardi@unifortunato.eu

Marta Cimitile
Unitelma Sapienza
Roma, Italy
marta.cimitile@unitelma.it

Fabio Martinelli
Institute for Informatics and Telematics
National Research Council of Italy (CNR)
Pisa, Italy
fabio.martinelli@iit.cnr.it

Francesco Mercaldo
Institute for Informatics and Telematics
National Research Council of Italy (CNR)
Pisa, Italy
francesco.mercaldo@iit.cnr.it

ABSTRACT

Online games consumers are strongly grown in the last years attracted by the always higher quality of the games and the more effective gaming infrastructures. The increasing of on line games market is also concurrent to the diffusion of game bots that allow to automatize malicious tasks obtaining some rewards with respect to the other game players (the game bots user increases personal benefits and popularity with low effort). Given the interest of game developers to preserve game equity and player satisfaction, the topic of game bots detection is becoming very critical and consists to distinguish between game bots and human players behaviour. This paper describes an approach to the online role player games bot detection based on time series classification used to discriminate between human and game bots behavioral features. In this paper an application of the proposed approach in a real role player game is reported.

CCS CONCEPTS

• **Security and privacy → Software security engineering**;

KEYWORDS

time series, game bot detection, machine learning

ACM Reference format:
Mario Luca Bernardi, Marta Cimitile, Fabio Martinelli, and Francesco Mercaldo. 2017. A Time Series Classification Approach to Game Bot Detection. In *Proceedings of WIMS '17, Amantea, Italy, June 19-22, 2017,* 11 pages.
https://doi.org/10.1145/3102254.3102263

1 INTRODUCTION

The game market has been interested in a fleeting evolution in the last years. Looking to the computer age, firstly the game player is confined in a solitary environment playing alone or with a limited number of friends that are in some physical place. Secondly, in the 1990s, the Internet diffusion drives the birth of a plethora of online games consisting of a video game that is either partially or primarily played using an Internet network or another computer network [1]. The online games diffusion allows to create an online player community overrunning the physical distances [30, 34] and extending the existing real-life player communities [34]. Moreover, several platforms for online gaming have been developed to support the worldwide games market that is characterised by always more sophisticated online games. Indeed, online games became ubiquitous on the gaming platforms like PCs, consoles and mobile devices. Moreover, they span many genres enjoying first-person shooters, computer role play game, strategy games, as well as virtual environments for socialising with other players. In particular, in the late 1990s, a new genre of online game called Massively Multiplayer Online Role-Playing Games (MMORPGs) became very popular in the game industry. It consists of putting together both role-playing video games and massively multiplayer online games and supporting the interaction of a large number of players in a virtual world [26]. Given their capability to attract a widespread audience, having different nationalities, occupation and ages and interested to all the online genres listed above [11, 31, 36], MMORPGs are seen as a profitable business by the game bot developers that continually work to improve MMORPGs quality and guarantee customer satisfaction. According to this, they observe that one of the reasons for a possible MMORPGs customer disappointment is the presence of cheater players that use illegal methods to obtain game advantages. In particular, some players can use game bots to perform illegal activities and consequently honest players have been discouraged to play and can decide to retire from the game. Attackers perpetrate illegal activities in the online game [6] with the aim to obtain economical advantages (cyber money can be easily changed into real currency [25]), capture reserved information about the other game players or acquire popularity in the game community with few effort. These illicit behaviours are performed using some game bots that allow to automatically repeat a malicious task in the online game. Game bots are defined as some artificial intelligent system software that can simulate human behaviour in an online game [35]. They can be used in the MMORPG to perform several activities like automating a repetitive and tedious task using a program that simulates the human player's behaviour in the way to earn much more game money and items than human users (a game botnet can

*Produces the permission block, and copyright information

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WIMS '17, June 19-22, 2017, Amantea, Italy
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5225-3/17/06…$15.00
https://doi.org/10.1145/3102254.3102263

play without requiring a break). Basing on the above considerations, we can understand that game bots are damaging for the game reputation, cause inflation in a game's economy and shorten the game's lifecycle [16]. For this reason, MMORPG producers are interested to find new approaches to the game bots detection. According to this, some strategies for game bot detection have been exploited. They consist mainly in repeated Turing test [14], network traffic analysis [21] or bot scanning. The limits of these approaches are that i) they usually interfere with game playing and ii) they can be easily evaded by game bots.

In this paper, we introduce an approach to the game bot detection based on a behaviour analysis of the game player. The main idea is that human behaviour is different from the game bot one and the game bot can be detected analysing this difference on a suitable set of behavioural features. To this aim, we use time series machine learning techniques to build several classifiers able to discriminate human players and game bots basing on a set of behavioural features. We propose and tested the following features: Player Information (PI), Player Actions (PA), Group Activities (GA), Social Interaction Diversity (SID) and Network Measures (NM). For the proposed features, we evaluate their effectiveness of discriminating human and game bot behaviours in a real MMORPG game. This evaluation has been performed using a real-world dataset obtained from the operation of the game called Aion: The Tower of Eternity [1]. It is a very popular MMORPG fantasy free-to-play game. The game company obtained this dataset by collecting, identifying and labelling operations performed by both real players and game bots.

The paper is organized as follows: Section 2 contains some basic notions about time series. Section 3 introduces the proposed features and the detection technique while Section 4 presents the results of the evaluation. Section 5 contains an overview of related work. Finally, paper conclusion and future works are reported in Section 6.

2 BACKGROUND

In this Section, we provide some preliminaries about time series algorithms to deeply understand our game bot identification technique. Time Series is meant to classify different observations of a phenomenon concerning a qualitative character [8]. If this character is represented by the time variation, the series is called historical or time. As a matter of fact, a time series is a sequence taken at successive equally spaced points in time (it is a sequence of discrete-time data). The time series analysis can be used: (i) to briefly describe the time course of a phenomenon (for instance the chart of a series can be able to reveal both regularity and abnormal values); (ii) To explain the phenomenon, identifying its generating mechanism and the possible relationships with other phenomena; (iii) to filter the series (this means the breakdown of the series itself into its unobservable components); (iv) to predict the future trend of the phenomenon. Time Series Analysis is used for many applications such as: Economic Forecasting, Sales Forecasting, Budgetary Analysis, Stock Market Analysis and Yield Projections. There are many methods used to model a time series, for example, the time series forecasting is represented by the use of a model to predict future values based on previously observed ones. Starting from the consideration that time series data have a natural temporal ordering, a stochastic model for a time series will in general reflect the fact that observations close together in time will be more closely related than observations further apart. Furthermore, the time series models will often make use of the natural one-way ordering of time so that values for a given period will be expressed as deriving in some way from past values, rather than from future values.

In this paper we consider the multilayer perceptron (MLP) neural network model [12] to evaluate the effectiveness of the behavioural features in the discrimination between a human player and a game bot using te series. The MLP neural network consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one [37]. MLP employs a supervised learning technique, i.e., the backpropagation for training the network [27, 28]. MPL networks are very popular in research fields as image recognition [20] and speech recognition [15].

3 APPROACH

This section describes the adopted behavioural features and the classification approach based upon time series classification. The approach is based on a Multi-Layer Perceptron network for learning as proposed in [23] integrated with the trend and value based analysis proposed by [9]. The underlying assumption, behind the method, is that the playing behaviour distributions of a game bot and human user are not overlapping allowing the MLP-based classifier to correctly classify the players (as humans or bots). The Figure 1 depicts the overall classification process. The first step consists in the dataset pre-processing (performing cleaning up of incomplete and wrong sampled data sessions and normalising signals). The cleaned data is then used to generate two sets: a labelled training set used to train the classifier and an unlabeled test set to perform the performance assessment of the classifier. The remaining of the section discusses each step in more details.

3.1 The Features Model

The features considered for our study were extracted and computed from a real case and are further described in [16].

As reported in Section 1, the selected behavioural features fall into the following four categories: Player Information (PI), Player Actions (PA), Group Activities (GA), Social Interaction Diversity (SID) and Network Measures (NM). PIs are captured to identify the possible gaps between the game bots and the human user's personal values and scores. We take into account the following features: login frequency (PI_1), play time (PI_2), game money (PI_3) and a number of IP addresses (PI_4). PAs are all the features, listed in the following, that are related to the player actions within the game:

- Sitting (PA_1): an action taken by players to recover their health.
- Earning experience points (PA_2): to quantify a player character's progression through the game. Experience points are awarded for the completion of quests, overcoming obstacles and opponents, and for successful role-playing;
- obtaining items (PA_3): an action taken by user to recovery game items;

[1] https://en.aion.gameforge.com/website/

Table 1: The features involved in the study with the correspondent category.

Category	Features
Player Information	PI_1, PI_2, PI_3, PI_4
Player Actions	PA_1, PA_2, PA_3, PA_4,PA_5, PA_6, PA_7, PA_8, PA_9, PA_{10}
Group Activities	GA_1, GA_2
Social Interaction Diversity	SID_1
Network measures	NM_1, NM_2, NM_3, NM_4, NM_5, NM_6, NM_7, NM_8, NM_9

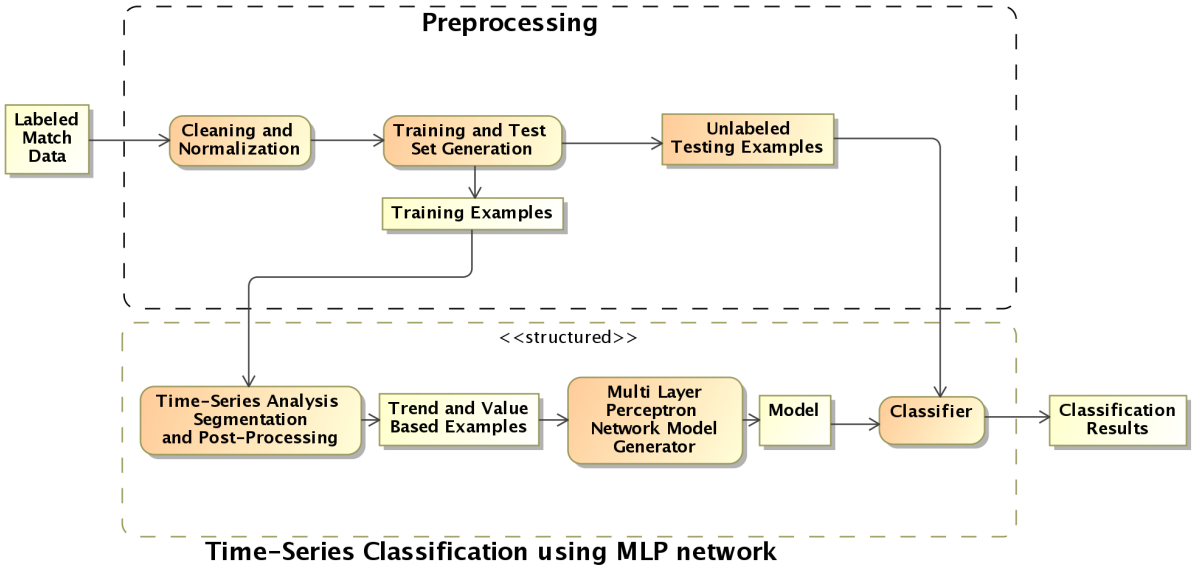

Figure 1: The Overall Classification Process.

- earning game money (PA_4): an action taken by user to earning game money;
- earning player kills points (PA_5): a player can acquire kill points by defeating players of opposing factions. Player kill points can be used to purchase various game items. They are also used to determine the player's rank within the game classification;
- harvesting items (PA_6): the action of harvesting game items;
- resurrecting (PA_7): any player may resurrect themselves by performing one or more activities;
- restoring experience points (PA_8): an action taken by user to recovery experience points;
- being killed by a non-player and player character (PA_9): a non-player character is a fictional character whose actions are automatized while a player character is a fictional character whose actions are directly controlled by a player of the game;
- using portals (PA_{10}): portals allow player to move to the next game step.

Looking to the above-described PA features, we can suppose that they assume different values for a human user or game bot. For example, it may be possible that game bots sit more frequently than human users to continually recover health and achieve many points. Similarly, game bots can perform a higher number of actions to earning experience points or obtaining items on the human users. More interestingly, game bots can connect to the game and play for long periods of consecutive hours, differently from human users, that typically are not able to play during several time-windows (for instance during work hours and sleep). GA features are also based on the idea that there is a gap between the values of the social features of game bots on the human users. In effect, game bots do not attempt to perform social activities as humans.

The GA features include the average duration of party play (GA_1) and some guild activities (GA_2). Looking to the first, party play is a group of two or more players that decide to undertake quests or missions together to increase their capability to complete difficult quests by collaboration and enjoy socialisation. Sometimes, also game bots take part to the party play, but differently, from the human users, they aim to acquire game money and items faster and more efficiently. Similarly, group activities are a set of activities that need to be performed with the collaboration of other players and exhibit different values between humans and bots.

The SID features describe the entropy of the match. We expect that

game bots perform only particular actions, whereas human users execute a larger set of socialising tasks.
The SID_1 metric measure the variety of tasks on all possible interactions, as executed by the player. Finally, NM features characterise the player's behaviour in the players' interaction social network. This social network can be represented as a graph having characters as the nodes and their interactions as the edges. For example, an edge between two nodes in the social network graph may highlight the transfer of an item between the two characters. NM include the following metrics as features:

- Degree centrality (NM_1): this feature represents the centrality focused on the degree. The more edges an actor has, the more important it is;
- Betweenness centrality (NM_2): it counts the number of shortest paths between two nodes on which a given actor resides;
- Closeness centrality (NM_3): an actor is considered important if it is relatively close to all other actors. Closeness is based on the inverse of the distance of each actor to every other actor in the network;
- Eigenvector centrality (NM_4): Indicates that a given node has a relationship with other valuable nodes. A high eigenvector value for an actor means that a node has several neighbours with high eigenvector values;
- Eccentricity (NM_5): the eccentricity of node v is calculated by computing the shortest path between node v and all other nodes in the graph; then the longest shortest path is chosen;
- Authority (NM_6): exhibits a node pointed to by many good hubs;
- Hub (NM_7): exhibits a node that points to many good authorities;
- PageRank (NM_8): assigns a numerical weight to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set;
- Clustering coefficient (NM_9): it quantifies how close neighbours are to being a clique. A clique is a subset of all of the edges connecting pairs of vertexes of an undirected graph.

3.2 The Feature Selection

An accurate features selection when trying to reduce the dimensionality of a particular dataset is necessary and require to analyse and understand feature's impact on a model. It is essential to identify which of those features are most relevant to the model and to understand how they correlate with each other, as working with a subset of an orthogonal and independent set of features allows to discard a lot of irrelevant and redundant information. In this work, we opted for a correlation-based feature selection approach (CFS) that exploits correlation matrix filters as discussed in [29]. CFS is a filter algorithm that rates feature subsets according to a correlation based heuristic evaluation function. The evaluation function is biased towards those subsets containing features that are highly correlated with the class and uncorrelated with each other. Hence unnecessary features should be ignored because they will have low correlation with the class. Moreover, redundant features should be eliminated as they will be highly correlated with one or more of the remaining features. The approval of a feature will depend on the extent to which it allows efficient classification in regions of the examples space not already covered by other features. CFS's feature subset evaluation function is:

$$M_S = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}} \quad (1)$$

where M_S is the heuristic goal function of a feature subset S containing k features, r_{cf} is the mean feature-class correlation and r_{ff} is the average feature-feature inter-correlation. Intuitively the numerator of the equation provides an indication of how good a set of features is to classify an example whereas the denominator gives us an indication of how much redundancy there is among the features set.

3.3 Time-Series Classification

To perform an effective classification of behaviours, our approach is based on the extraction of several statistical trend-based and value-based features from time series. The nature and the number of features needed depend on their discriminatory power. There are several mandatory characteristics of such properties: ease of evaluation, not sensitive to transformations. In this work, we used a combination of properties and metrics as proposed in [23] and [9]. The statistical features were selected mixing simple and easy to implement features with the adoption of more complex features (like moments of several orders) needed to improve the efficiency and performances of the method. Feature are then evaluated and the features vectors of each time series are presented to an MLP neural network for classification.

4 THE EVALUATION

We designed an experiment to evaluate the effectiveness of the proposed approach based on the MLP neural network.

More specifically, the experiment is aimed at verifying whether the behavioural features can discriminate the game bot attacks by the human user behaviour. The classification is carried out by using several state-of-the-art machine learning classifiers built with the behavioural feature categories we considered.

The dataset involved in the study was obtained from the operation of Aion, a popular game and it is freely available. The dataset contains all in-game action logs for 88 days, between April 9th and July 5th of 2010. During this period, there were 49,739 players that played more than three h. Among these players, 7702 characters were game bots, identified by the game company. The banned list was provided by the game company to serve as the ground truth, and each banned user has been vetted and verified by human labour and active monitoring. For the log released by the game company, we aggregated the values of the features related to the same user in our dataset, and we marked the feature vectors as "human" or "game bot" according to the game company indications. To assure the privacy of users the dataset is anonymized from user private and personal information. Indeed, the consent of users is taken into account by ensuring that data analysis is within the scope of the end user license agreement: as a matter of fact, when users joined

the Aion game, users granted to NCSOFT, Inc. the permission to use and share user data for analysis purpose.

The evaluation consists in the execution of the steps reported in the process of Figure 1. As an initial validation of the labelled dataset, an analysis of the descriptive statistics of the populations of behavioural features was performed. This to ensure that distributions of metrics are well separated between humans and bots. The next step was the feature selection. The classification analysis aimed at assessing whether the considered feature categories can correctly classify bot and human behaviour. For what concerns the descriptive statistics, we report the box plot of the distribution of game bot and human behaviour for the accepted features to show that the distributions are not overlapping.

The classification analysis was performed using the R statistics environment[2].

4.1 Descriptive statistics

Figures 2, 3, 4, 5, 6, 7, 8, 9 and 10 show the box plots for a subset of features belonging to each category considered in the study. For space reasons, we do not show the box plots related to all features, but they lead to the same considerations.

Figure 2 shows the box plots related to the PI_1 between game bot and human users. The box plots are very similar; we conclude that for the feature point of view game bot and human user login with similar frequency. In our opinion, the explanation of the obtained result is that the feature does not consider the login time but only the login frequency.

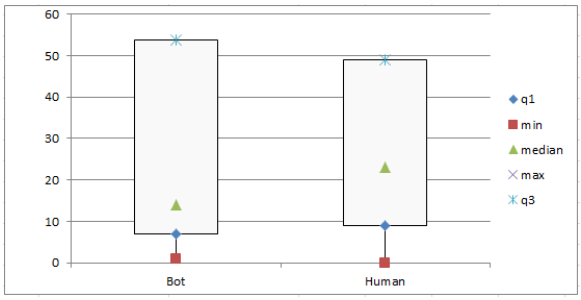

Figure 2: The box plot relating to the game bot and human distributions for PI_1.

The box plot in Figure 3 is related to the PI_2. In this box plot the distributions between game bots and human users are very different: as matter of fact, while human user presents a small distribution, the game bot one exhibits wider values if compared with the human user one. The reason is that game bot can play the game without the need to break, differently from human users.

Figure 4 shows the box plot related to game bot and human distributions for PA_1. The PA_1 is related to the number of rounds that game bots and human users can play. This result is consistent with the one we obtained by analysing the previous box plot: as matter of fact, game bots distribution seems to be wider if compared with the human users one.

[2]http://www.project-r.org

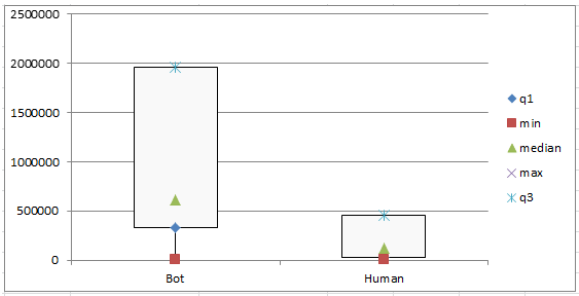

Figure 3: The box plot relating to the game bot and human distributions for PI_2.

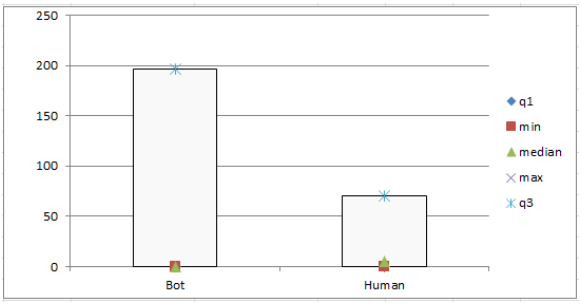

Figure 4: The box plot relating to the game bot and human distributions for PA_1.

Figure 5 shows the box plot related to PA_2. This feature is related to the ability to earn points to buy power for the character. Confirming what we expected, game bots can gather more points if compared with human users: this result is reflected in the wider dimension of game bots distribution with respect with the human users one.

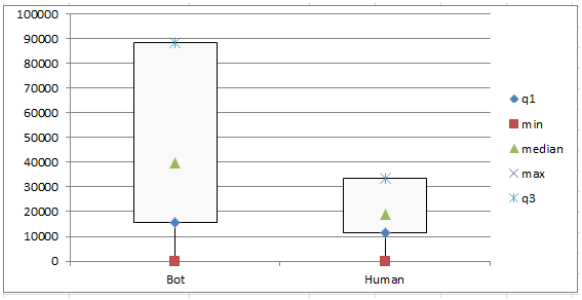

Figure 5: The box plot relating to the game bot and human distributions for PA_2.

Figure 6 shows the box plot related to the GA_1. In this case, the human user box plots are wider if compared with the game bot one. This happens because usually, game bots have not interest to make matches to play with other users: the focus of game bots is only to gather points to have character reinforcements.

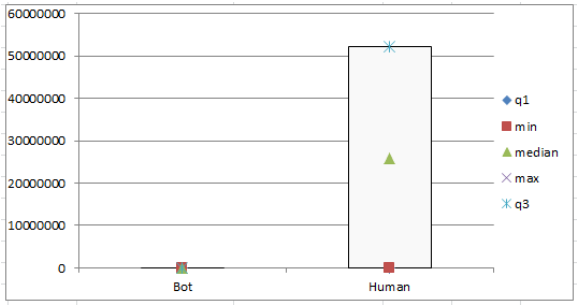

Figure 6: The box plot relating to the game bot and human distributions for GA_1.

Figure 7 shows the box plots related to the GA_2. The feature is related to the capacity of player to perform role missions. Confirming the previous box plot, game bot has no interest to cooperate to play, while human user usually needs to play together to complete successfully difficult missions.

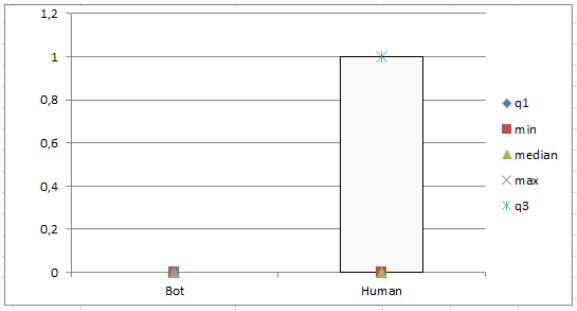

Figure 7: The box plot relating to the game bot and human distributions for GA_2.

Figure 8 shows the distributions for the SID_1 feature. The two distributions appear to be very similar: this is symptomatic of the fact the game bots like human users interact between them, for this reason, the considered feature is not discriminative between the two distributions.

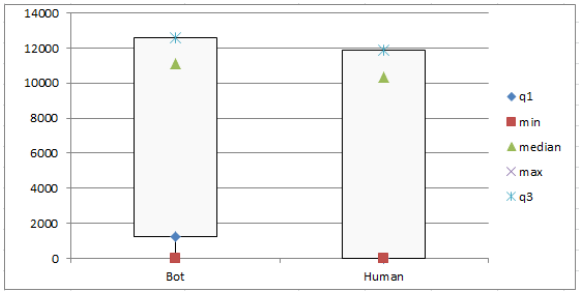

Figure 8: The box plot relating to the game bot and human distributions for the SID_1.

Figure 9 shows the box plot related to the NM_1 feature. These box plots exhibit that human users present a wider degree centrality if compared with game bots. We conclude from these box plots that human users hare more friends if compared with game bots, for this reason, the human user's distribution is wider if compared with game bots one.

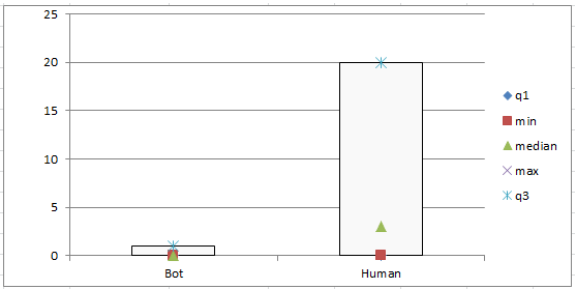

Figure 9: The box plot relating to the game bot and human distributions for NM_1.

Figure 10 shows the NM_2 feature box plots. As in the previous box plots, the human user's distribution is wider than the game bot one. It is happening because, in order to reach an objective or another player, the human player uses the shortest path. Differently game bots do not consider this: this is reflected by the different distributions of the box plots.

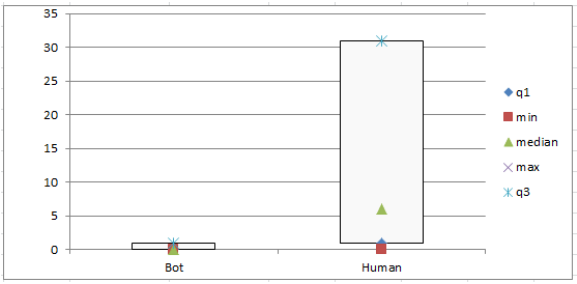

Figure 10: The box plot relating to the game bot and human distributions for NM_2.

4.2 Classification analysis

Four metrics were used to evaluate the classification results: Precision, Recall, F-Measure and ROC Area.

The precision has been computed as the proportion of the examples that truly belong to class X among all those which were assigned to the class. It is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved:

$$Precision = \frac{tp}{tp+fp}$$

where tp indicates the number of true positives and fp indicates the number of false positives.

The recall has been computed as the proportion of examples that were assigned to class X, among all the examples that truly belong to the class, i.e., how much part of the class was captured. It

is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$Recall = \frac{tp}{tp+fn}$$

where *tp* indicates the number of true positives and *fn* indicates the number of false negatives.

The F-Measure is a measure of a test's accuracy. This score can be interpreted as a weighted average of the precision and recall:

$$F\text{-}Measure = 2 * \frac{Precision*Recall}{Precision+Recall}$$

The Roc Area is defined as the probability that a positive instance randomly was chosen is classified above a negative randomly chosen.

To train the classifier, we defined T as a set of labeled traces *(M, l)*, where each time series M is associated to a label $l \in \{B, H\}$ (where B represents the game bot, while H the human user).

For the learning phase, we use a k-fold cross-validation: the data set is randomly partitioned into k subsets. A single subset is retained as the validation dataset for testing the model, while the remaining $k-1$ subsets of the original dataset are used as training data. We repeated the process for $k = 10$ times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the procedure depicted in Figure 1 and summarized below:

(1) build a training set $T \subset D$;
(2) build a testing set $T' = D \div T$;
(3) run the training phase on T;
(4) apply the learned classifier to each element of T'.

The results that we obtained with this procedure are shown in Table 2.

The following consideration can be made:

- best results are obtained for the threshold based segmentation on fixed windowing (the rows labelled with TB): the F-measure ranges from 0,88 to 0,98 in this class depending on the feature category (the best category is PA);
- using a single MLP hidden layer provides best times and good performances, but best F-measure was always below 0,89;
- using a greater MLP network with more nodes, we obtain the best results on classification (both the F-measure and the ROC area are equal to 0,98). However for this class times becomes higher making features selection mandatory (this is particularly true for the usage in running platforms to identify and disable suspect accounts at run-time);
- the worst feature category resulted from NM in particular for what concerns false negatives.

4.3 Feature Selection

As highlighted in Section 3, features selection was performed using a CFS-based approach. Figure 11 shows the Correlation Matrix used as a filter (removing from the set all the features with a correlation greater than 0.8) for the PA group.

Results of the feature selection are shown in Table 3.

We obtained that the most discriminating feature in the PI category is just the play time one, while in PA category are four features: sitting, earning experience points, obtaining items and earning player kill points.

To evaluate whether the two new feature set belonging to PI and PA categories can overcome the previous classifiers we learned, we build different classifiers with the features resulting from the features selection step: Table 4 shows the results we obtained.

The overall values of precision and recall after features selection remains quite good on a much lower number of features set. This allows to build the model and train the MLP classifier much more efficiently obtaining almost the same performances.

- relating to the network with one hidden layer, the best precision value lowers from the 0.89 to 0.88 whereas the recall one lowers from 0.89 to 0.87; the model construction time falls to 389s to 280s with a reduction of 30%.
- relating to the network with two hidden layers, the best precision remains equal whereas the recall one lowers from 0.98 to 0.97; but in this case, the model construction time is even lower (the model is constructed in 336s instead of 536s) with a reduction of more than 40%.

The time analysis confirms the effectiveness of the feature selection step in order to quickly identify the game bot: as matter of fact with the exception of the RandomForest algorithm, the remaining ones can learn the classifiers in less than 1 second: for this reason the developed classifiers can discriminate the game bot from the human user in less than one second in the worst case (i.e., when the classifier is built with the new data).

After the feature step selection, the best feature set is represented by following feature belonging to PA Category: sitting, earning experience points, obtaining items and earning player kill points. Classifying with this feature set we obtain, using the RandomForest classification algorithm, a precision equal to 0.96 and a recall equal to 0.986.

5 RELATED WORK

The increasing interest to detect game bots in MMORPG drove several studies pointed to the adoption of machine learning techniques. In [16] a deepen analysis of existing approaches is proposed. Here, authors define as server-side botnet detection approaches, the set of methods based on data mining techniques to analyse log data from game servers. The advantage deriving by the adoption of these approaches is that online game service providers can detect and block the game bot users without deploying additional programs on the client-side. Server-side detection methods can be classified into the following categories: social activity, action frequency, sequence, moving path, similarity, and gold farming group. In this work we focus on the social activity approaches. Social activity detection methods are based on users social interactions analysis [33]. For instance, in [24] authors assume that humans and game bots tend to form their social network in different ways and use some features to capture this social behaviour for detecting game bots. Also in [17], a statistical analysis of user behaviours in game activity logs is performed. Authors establish some threshold levels for the

Category	MLP/Window	Precision	Recall	F-Measure	Roc Area	Time (min)
	MLP 1 HL/100s	0,80	0,76	0,78	0,79	280,00
	MLP 1 HL/150s	0,82	0,77	0,80	0,81	326,00
	MLP 1 HL/200s	0,84	0,79	0,81	0,83	338,00
	MLP 1 HL/250s	0,86	0,81	0,83	0,85	363,00
PI	MLP 1 HL/TB	0,88	0,83	0,85	0,87	402,00
	MLP 2 HL/100s	0,89	0,84	0,87	0,89	416,00
	MLP 2 HL/150s	0,91	0,86	0,89	0,90	450,00
	MLP 2 HL/200s	0,93	0,88	0,90	0,92	464,00
	MLP 2 HL/250s	0,95	0,90	0,92	0,94	469,00
	MLP 2 HL/TB	0,96	0,92	0,94	0,96	470,00
	MLP 1 HL/100s	0,82	0,82	0,82	0,84	247,00
	MLP 1 HL/150s	0,84	0,84	0,84	0,85	293,00
	MLP 1 HL/200s	0,86	0,86	0,86	0,87	341,00
	MLP 1 HL/250s	0,87	0,87	0,87	0,89	387,00
PA	MLP 1 HL/TB	0,89	0,89	0,89	0,91	389,00
	MLP 2 HL/100s	0,91	0,91	0,91	0,93	432,00
	MLP 2 HL/150s	0,93	0,93	0,93	0,95	480,00
	MLP 2 HL/200s	0,95	0,95	0,95	0,96	486,00
	MLP 2 HL/250s	0,96	0,96	0,96	0,98	503,00
	MLP 2 HL/TB	0,98	0,98	0,98	0,98	536,00
	MLP 1 HL/100s	0,78	0,80	0,79	0,81	268,00
	MLP 1 HL/150s	0,80	0,82	0,81	0,83	282,00
	MLP 1 HL/200s	0,82	0,84	0,83	0,85	304,00
	MLP 1 HL/250s	0,84	0,86	0,85	0,86	319,00
GA	MLP 1 HL/TB	0,86	0,88	0,87	0,88	329,00
	MLP 2 HL/100s	0,87	0,89	0,88	0,90	389,00
	MLP 2 HL/150s	0,89	0,91	0,90	0,92	422,00
	MLP 2 HL/200s	0,91	0,93	0,92	0,94	470,00
	MLP 2 HL/250s	0,93	0,95	0,94	0,95	496,00
	MLP 2 HL/TB	0,95	0,97	0,96	0,97	499,00
	MLP 1 HL/100s	0,74	0,79	0,76	0,78	233,00
	MLP 1 HL/150s	0,76	0,80	0,78	0,80	266,00
	MLP 1 HL/200s	0,78	0,82	0,80	0,81	275,00
	MLP 1 HL/250s	0,80	0,84	0,82	0,83	290,00
SID	MLP 1 HL/TB	0,81	0,86	0,83	0,85	292,00
	MLP 2 HL/100s	0,83	0,88	0,85	0,87	387,00
	MLP 2 HL/150s	0,85	0,89	0,87	0,89	429,00
	MLP 2 HL/200s	0,87	0,91	0,89	0,90	444,00
	MLP 2 HL/250s	0,89	0,93	0,91	0,92	467,00
	MLP 2 HL/TB	0,90	0,95	0,92	0,94	478,00
	MLP 1 HL/100s	0,77	0,65	0,70	0,72	233,00
	MLP 1 HL/150s	0,79	0,67	0,72	0,74	262,00
	MLP 1 HL/200s	0,81	0,69	0,74	0,75	307,00
	MLP 1 HL/250s	0,82	0,70	0,76	0,77	343,00
NM	MLP 1 HL/TB	0,84	0,72	0,78	0,79	389,00
	MLP 2 HL/100s	0,86	0,74	0,79	0,81	516,00
	MLP 2 HL/150s	0,88	0,76	0,81	0,83	521,00
	MLP 2 HL/200s	0,90	0,78	0,83	0,84	546,00
	MLP 2 HL/250s	0,91	0,79	0,85	0,86	567,00
	MLP 2 HL/TB	0,93	0,81	0,87	0,88	614,00

Table 2: Classification results: Precision, Recall, F-Measure and RocArea for classifying the feature categories, computed with six different MLP network and windows size for signal segmentation. The Time column represents the time in seconds taken to build the model.

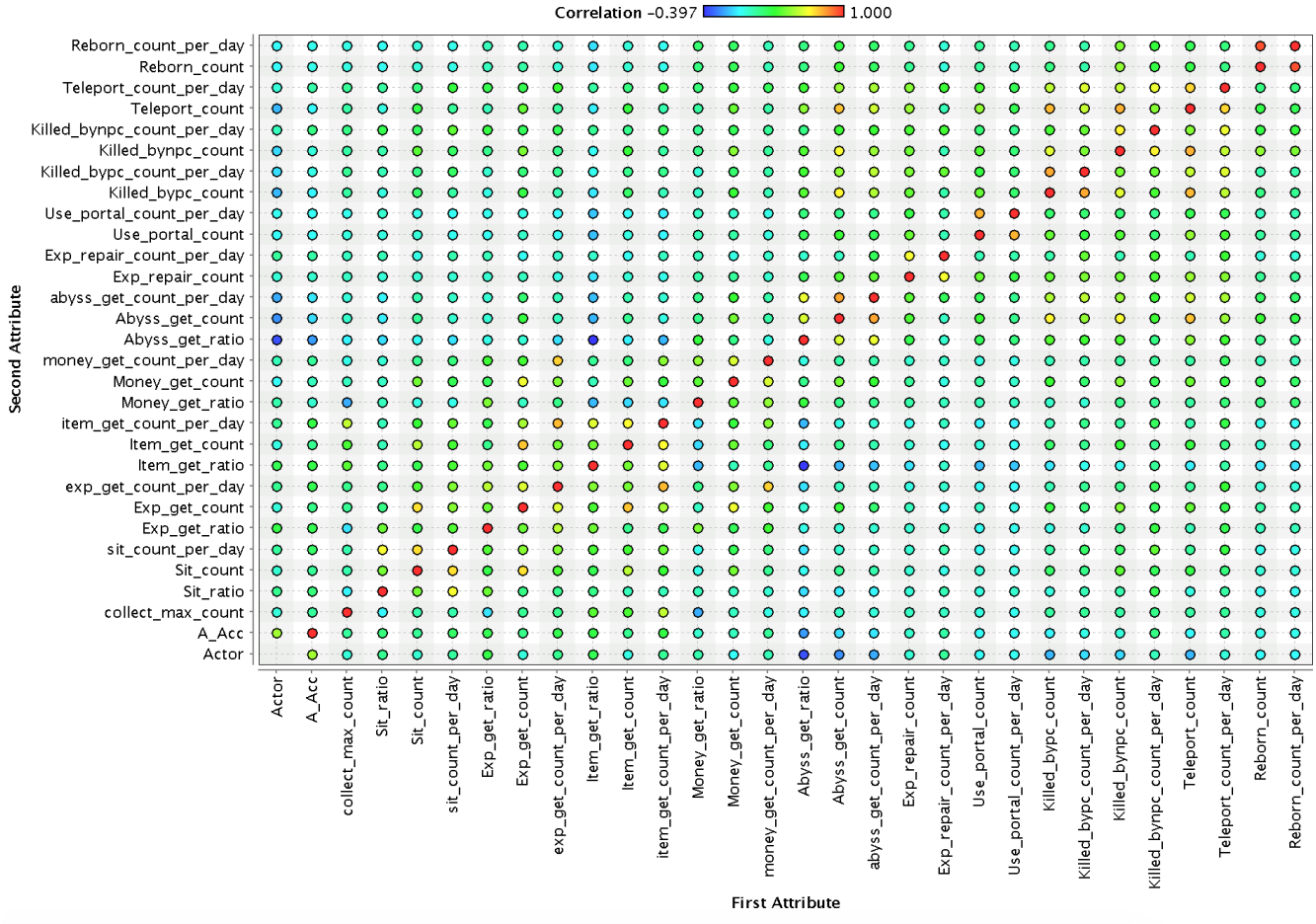

Figure 11: Correlation Matrix for Feature Selection on "Player Actions" features set.

analysed activities to discriminate game bots from human being behaviour. Similarly, in [19], the bot detection is performed analysing the window event sequences produced by the game players. Some learning algorithms have been used to distinguish human from auto player event sequences. Moreover, in [5] a manifold learning approach for game bots detection is proposed. This approach is based on the analysis of avatars movement trajectories assuming that human players trajectories differ from the game bot ones. The difference between human players and game botnet trajectories are also verified in [4]. Here, the proposed trajectories analysis method has been validated on a real game called Quake 2 and the obtained results are of interest (detection accuracy is more than 95% for a trace of 200 seconds or longer). Similarly, authors in [3] propose an approach based on comparing the traffic generated by human players versus the game bots. Other approaches that are mainly based on behavioral analysis of the game playing to discriminate between human and botnet are proposed in [13, 16, 18, 22, 32]. The common factor for all this approaches consists in the assumption that the different game style of the human player can be used to discriminate between game bots and human players. Looking to the behavioural approaches reported above, we can underline that their main drawback consists in the limited number of considered features (they usually are based on one or two behavioural features) that are selected basing on the game domain more than on the player game style. Indeed, the limited number of features makes the approach dependent on the single game domain. This issue is discussed in [7]. Here, authors propose an alternative method based on a larger set of features for the game players behaviour analysis. This approach is closer to our proposed approach considering both game-related and player-related features. On [7], in our approach, we show a higher precision in the game bot detection.

To the best of authors knowledge, this is the first approach to game bot detection exploiting time series classification using an MLP network. The approach has the advantage that it takes into account the history of the player's actions and behaviour on cited ones and this makes it more suitable to identify bot misbehaviours with greater accuracy. Moreover, the well-known drawback of the method related to the sensitivity of the noise in the classified time series has no impact in this case since collected data is not affected by noise. The efficiency of the approach is comparable with the cited ones, in particular, performing filter-based features selection

Table 3: Feature Selection Results.

PI	*play time*
PA	*A_Acc*
PA	*collect_max_count*
PA	*Sit_ratio*
PA	*Sit_count*
PA	*Exp_get_ratio*
PA	*Exp_get_count*
PA	*exp_get_count_per_day*
PA	*collect_max_count*
PA	*Item_get_ratio*
PA	*Money_get_ratio*
PA	*Money_get_count*
PA	*Abyss_get_ratio*
PA	*Abyss_get_count*
PA	*Exp_repair_count*
PA	*Exp_repair_count_per_day*
PA	*Use_portal_count*
PA	*Killed_bypc_count*
PA	*Killed_bynpc_count*
PA	*Killed_bynpc_count_day*
PA	*Teleport_count_per_day*
PA	*Reborn_count*

Table 4: Feature Selection Classification results: Precision, Recall, F-Measure and RocArea for the selected feature.

Category	Algorithm	Precision	Recall	F-Measure	Roc Area	Time
Selected Features (see Table 3)	MLP 1 HL/100s	0,82	0,82	0,82	0,84	134,00
	MLP 1 HL/150s	0,84	0,83	0,84	0,85	193,00
	MLP 1 HL/200s	0,85	0,84	0,85	0,87	221,00
	MLP 1 HL/250s	0,86	0,87	0,87	0,89	237,00
	MLP 1 HL/TB	0,88	0,87	0,89	0,91	280,00
	MLP 2 HL/100s	0,91	0,90	0,91	0,93	232,00
	MLP 2 HL/150s	0,92	0,92	0,93	0,95	280,00
	MLP 2 HL/200s	0,94	0,95	0,95	0,96	286,00
	MLP 2 HL/250s	0,95	0,97	0,96	0,98	303,00
	MLP 2 HL/TB	0,98	0,97	0,98	0,98	336,00

(obtaining a reduction of time up to 40%). It's also interesting however to perform more experimentation comparing features selection filter and wrapped approaches to discover if there are chances to further reducing times preserving the classifier effectiveness.

6 CONCLUSIONS

The online game is becoming a widespread market. Users can connect to the Internet to play with other members of the game community. They can perform a set of activities aiming to win upgrades for their characters, socialise with other players, enhance game currency and selling characters boosted. Unfortunately, there are also some unfair players that are interested in acquiring more popularity and currency in the game with less effort (in particular game currency can be easily changed in real word money). These users can be interested in adopting the game bot to automatize game activities. Indeed, a game bot can simulate the human player behaviour performing the playing activities without all the interruption that are necessary to the human players (i.e., sleeping time). It became very important for game producers to detect game bot and expel from the game all the swindler players. In this paper, an approach to discriminate human users from game bots is proposed. It is based on a set of discriminating behavioural features: some features are related to the player and other are related to the game. The effectiveness of the behavioural features in the discrimination between human player and game bot using a time series technique is evaluated using a MLP neural network model [12]. The obtained results show an F-Measure equal to 0.95. As future work, we will extend our investigation whether the feature vector we considered in this work can detect bot in the social network. Furthermore, we will also consider the adoption of Process Mining techniques [2]

and Formal Methods [10] to extract the game bots patterns and verify if there are differences with respect the human user's ones.

ACKNOWLEDGMENTS

This work has been partially supported by H2020 EU-funded projects NeCS and C3ISP and EIT-Digital Project HII.

REFERENCES

[1] Ernest Adams. 2014. *Fundamentals of game design.* Pearson Education.
[2] Mario Luca Bernardi, Marta Cimitile, Chiara Di Francescomarino, and Fabrizio Maria Maggi. 2016. Do Activity Lifecycles Affect the Validity of a Business Rule in a Business Process? *Inf. Syst.* 62, C (Dec. 2016), 42–59. https://doi.org/10.1016/j.is.2016.06.002
[3] Kuan-Ta Chen, Jhih-Wei Jiang, Polly Huang, Hao-Hua Chu, Chin-Laung Lei, and Wen-Chin Chen. 2006. Identifying MMORPG Bots: A Traffic Analysis Approach. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE '06)*. ACM, New York, NY, USA, Article 4. https://doi.org/10.1145/1178823.1178829
[4] Kuan-Ta Chen, Andrew Liao, Hsing-Kuo Pao, and Hao-Hua Chu. 2009. Game bot detection based on avatar trajectory. *Entertainment Computing-ICEC 2008* (2009), 94–105.
[5] Kuan-Ta Chen, Hsing-Kuo Kenneth Pao, and Hong-Chung Chang. 2008. Game bot identification based on manifold learning. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*. ACM, 21–26.
[6] Ying-Chieh Chen, Patrick S Chen, Ronggong Song, and Larry Korba. 2004. Online Gaming Crime and Security Issue-Cases and Countermeasures from Taiwan.. In *PST*. 131–136.
[7] Yeounoh Chung, Chang Yong Park, Noo-Ri Kim, Hana Cho, Tae Bok Yoon, Hunjoo Lee, and Jee-Hyong Lee. 2015. A Behavior Analysis-Based Game Bot Detection Approach Considering Various Play Styles. *CoRR* abs/1509.02458 (2015).
[8] Philippe Esling and Carlos Agon. 2012. Time-series Data Mining. *ACM Comput. Surv.* 45, 1, Article 12 (Dec. 2012), 34 pages. https://doi.org/10.1145/2379776.2379788
[9] Bilal Esmael, Arghad Arnaout, Rudolf K. Fruhwirth, and Gerhard Thonhauser. 2012. *Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations.* Springer Berlin Heidelberg, Berlin, Heidelberg, 392–403. https://doi.org/10.1007/978-3-642-31128-4_29
[10] Nicoletta De Francesco, Giuseppe Lettieri, Antonella Santone, and Gigliola Vaglini. 2016. Heuristic search for equivalence checking. *Software and System Modeling* 15, 2 (2016), 513–530. https://doi.org/10.1007/s10270-014-0416-2
[11] Mark D Griffiths, Mark NO Davies, and Darren Chappell. 2004. Online computer gaming: a comparison of adolescent and adult gamers. *Journal of adolescence* 27, 1 (2004), 87–96.
[12] Simon Haykin and Neural Network. 2004. A comprehensive foundation. *Neural Networks* 2, 2004 (2004), 41.
[13] Sylvain Hilaire, Hyun-chul Kim, and Chong-kwon Kim. 2010. How to deal with bot scum in MMORPGs?. In *Communications Quality and Reliability (CQR), 2010 IEEE International Workshop Technical Committee on*. IEEE, 1–6.
[14] P. Hingston. 2009. A Turing Test for Computer Game Bots. *IEEE Transactions on Computational Intelligence and AI in Games* 1, 3 (Sept 2009), 169–186. https://doi.org/10.1109/TCIAIG.2009.2032534
[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and others. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.
[16] Ah Reum Kang, Seong Hoon Jeong, Aziz Mohaisen, and Huy Kang Kim. 2016. Multimodal game bot detection using user behavioral characteristics. *SpringerPlus* 5, 1 (2016), 523.
[17] Ah Reum Kang, Jiyoung Woo, Juyong Park, and Huy Kang Kim. 2013. Online game bot detection based on party-play log analysis. *Computers & Mathematics with Applications* 65, 9 (2013), 1384–1395.
[18] Yoshitaka Kashifuji. 2008. Detection of MMORPG Bots Based on Behavior Analysis. *ACE* 2008 (2008), 4.
[19] Hyungil Kim, Sungwoo Hong, and Juntae Kim. 2005. Detection of auto programs for MMORPGs. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 1281–1284.
[20] Yann LeCun, Yoshua Bengio, and others. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361, 10 (1995), 1995.
[21] N. W. Lo and Shiou-Hung Chen. 2008. A study of anti-robot agent mechanisms and process on online games. In *2008 IEEE International Conference on Intelligence and Security Informatics*. 203–205. https://doi.org/10.1109/ISI.2008.4565057
[22] Yutaro Mishima, Kensuke Fukuda, and Hiroshi Esaki. 2013. An analysis of players and bots behaviors in MMORPG. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*. IEEE, 870–876.
[23] Alex Nanopoulos, Rob Alcock, and Yannis Manolopoulos. 2001. Feature-based Classification of Time-series Data. *International Journal of Computer Research* 10 (2001), 49–61.
[24] Jehwan Oh, Zoheb Hassan Borbora, Dhruv Sharma, and Jaideep Srivastava. 2013. Bot detection based on social interactions in MMORPGs. In *Social Computing (SocialCom), 2013 International Conference On*. IEEE, 536–543.
[25] Richard A Paulson and James E Weber. 2006. Cyberextortion: an overview of distributed denial of service attacks against online gaming companies. *Issues in Information Systems* 7, 2 (2006), 52–56.
[26] Thorsten Quandt and Sonja Kröger. 2013. *Multiplayer: The social aspects of digital gaming*. Vol. 3. Routledge.
[27] Frank Rosenblatt. 1961. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms.* Technical Report. DTIC Document.
[28] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation.* Technical Report. DTIC Document.
[29] Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán. 2007. *Filter Methods for Feature Selection – A Comparative Study.* Springer Berlin Heidelberg, Berlin, Heidelberg, 178–187. https://doi.org/10.1007/978-3-540-77226-2_19
[30] A Fleming Seay, William J Jerome, Kevin Sang Lee, and Robert E Kraut. 2004. Project massive: a study of online gaming communities. In *CHI'04 extended abstracts on Human factors in computing systems*. ACM, 1421–1424.
[31] Tina L Taylor. 2009. *Play between worlds: Exploring online game culture.* Mit Press.
[32] Ruck Thawonmas, Yoshitaka Kashifuji, and Kuan-Ta Chen. 2008. Detection of MMORPG bots based on behavior analysis. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACM, 91–94.
[33] Matteo Varvello and Geoffrey M. Voelker. 2010. 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 (NOSSDAV '10)*. ACM, New York, NY, USA, 9–14. https://doi.org/10.1145/1806565.1806570
[34] Barry Wellman and Milena Gulia. 1999. Virtual communities as communities. *Communities in cyberspace* (1999), 167–194.
[35] Roman V Yampolskiy and Venu Govindaraju. 2008. Embedded noninteractive continuous bot detection. *Computers in Entertainment (CIE)* 5, 4 (2008), 7.
[36] Nick Yee. 2008. Maps of digital desires: Exploring the topography of gender and play in online games. *Beyond Barbie and Mortal Kombat: New perspectives on gender and gaming* (2008), 83–96.
[37] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. 2014. Time series classification using multi-channels deep convolutional neural networks. In *International Conference on Web-Age Information Management*. Springer, 298–310.