

온라인 게임 로그 데이터 클러스터링 기반 일일 단위 게임봇 판별

김 주 환,^{1*} 최 진 영^{2†}
^{1,2}고려대학교 정보보호대학원 (대학원생, 교수)

Detecting Daily-Driven Game-Bot Based on Online Game Play Log Clustering

Joo Hwan Kim,^{1*} Jin-Young Choi^{2†}
^{1,2}Graduate School of Information Security, Korea University
(Graduate student, Professor)

요 약

온라인 게임 봇은 이미 수 많은 방식을 통해 사람들에게 알려져 왔으며, 사용자의 게임 흥미 저하, 게임 내 경제 순환 파괴, 게임 콘텐츠 및 수명 단축 등 많은 문제점을 야기한다. 정상적이지 않은 게임 봇 운영을 방지하는 것은 장기적으로 게임 제작사와 게임 플레이어에게 모두 악영향을 미치게 되므로 이에 대한 탐지 및 제재는 필수가 되었다. 하지만 제재 단계에서 생기는 오인 제재의 딜레마를 피하기 쉽지 않다. 게임사 측에서 유저를 제재하기 위해서는 객관적인 분석 지표인 로그를 가지고 제재 여부를 판단해야 하며, 로그에서 추출한 정보를 근거로 확보해야 한다.

본 연구에서는 탐지 대상 기간의 로그에 대하여 이를 일일 단위로 나누어서 게임봇 유저 판별을 수행할 것이다. 일일 단위 탐지를 위해 탐지 기간을 하루 단위로 나누어 해당 일자에 대한 게임봇 여부를 우선 판별하고, 이후 최종 결과를 판단하였다. 제안한 방법론을 통해 일반 유저 스타일과 게임봇 유저 스타일이 섞여 있는 경우를 쉽게 탐지해 낼 수 있을 것이다. 본 논문에서 제안한 방법론으로 테스트한 결과, 분류 정확도를 확인할 수 있는 지표 중 하나인 F1-score가 0.898에서 0.945로 향상되었다.

ABSTRACT

Online game-bots are already known for a lot of persons by various ways. It leads to problems such as declining game player's interest, in-game financial crisis, etc. Detecting and restricting of game-bot is now essential. Because both publishers and players get disadvantages from their long term abnormal working. But it is not easy to restrict, because of false restriction risks. Game publishers need to distinguish game-bot from server-side game logs. At last, it should can make reasons for game-bot restriction.

In this paper, we classified game-bot users by using daily separated game logs for testing data. For daily-driven detection, we separated total dataset into one day logs. Preliminary detects game-bots with one day logs, and determines total results by using these data. Daily driven detection advantages on detection which contains combined game playing style. Which shows like normal user and game-bot. These methodology shows better F1-score, which one of indicator which demonstrate classification accuracy. It increases from 0.898 to 0.945 by using Random Forest classifier.

Keywords: Online-game security, Bot detection, Data clustering, Categorizing

I. 서론

게임봇의 경우 사람이 직접 플레이 하지 않는다는 점을 악용하여 단순 노동 등의 반복 작업을 직접 했을 때에 비해 아주 낮은 비용으로 진행할 수 있어, 직접 플레이하는 유저들에 비해 생산성이 비정상적으로 늘어나게 된다. 이는 정상 플레이 유저들의 박탈감을 일으킬 수 있는 요소로, 게임봇 규모가 커지게 될 경우에는 이를 탐지하여 제재를 할 필요성이 있다. 특히 게임봇의 대부분을 차지하는 GFG(Gold Farming Group)의 주요 목적은 게임 내 재화를 수집해 이를 현금화 하는 것이 목적으로 게임의 본 목적인 재미의 영역을 넘어서게 된다.

게임봇 탐지가 고도화되어감에 따라, 게임봇 사용자들 역시 이러한 변화에 적응하여 탐지되지 않는 방향으로 발전해 나가고 있다. 단순 반복 입력이 아닌 랜덤 좌표 입력, 랜덤 시간 휴식, 랜덤 입력 패턴 등 무작위 요소를 포함 시킴으로서 단순 시퀀스 분석이나 인터벌 분석 등을 이용하여 탐지되지 않도록 게임봇을 발전시키고 있다. 정상 유저로 위장하기 위해 게임봇 목적에 필요 없는 행동들을 포함시키기도 하며, 이는 단순한 패턴 분석 및 정량 분석이 아닌 좀더 정밀하고 세밀한 차이를 감지할 수 있는 분석을 요구하게 만들고 있다.

게임에서 제재된 게임봇 유저의 경우 일반적으로 해당 게임 제작사에서 제재 사유를 상세히 알려주지 않는다. 해당 정보를 통해 게임봇 제작자가 어떠한 이유로 제재되었는지를 파악하여 악용할 수 있기 때문이다. 게임봇 분석 연구에서도 게임봇이 어떠한 유형 및 어느 시점의 플레이 데이터를 기반으로 제재되었는지를 알 수 없으며, 본 연구에서 사용한 데이터 역시 로그에 기록된 계정에 대해서 게임봇인지 아닌지 로만 표시되어 있다.

본 연구에서는 클러스터링 기법을 기반으로 일일 단위의 게임봇 탐지하는 방법을 소개한다. 게임봇으로 분류된 유저의 경우에도 모든 일자에 게임봇을 사용하지 않았을 가능성이 있으며, 이러한 경우 정상 유저의 로그와 유사하여 분석에 있어 혼동을 유발할 가능성이 높다. 이러한 오탐 영향을 줄이기 위해 클러스터링을 적용함으로써 일반 유저가 많이 속해있는 그룹의 경우 봇 유저의 영향력이 줄고, 봇 유저가 많이 속한 그룹의 경우에는 봇 유저의 영향력이 늘어갈 수 있을 것이다. 본 연구에서 일일 단위의 봇 사용 여부는 라벨로 주어지지 않았지만 클러스터링

기법을 이용함으로써 봇 유저와 가까운 로그 그룹을 추정하여 해당 그룹의 라벨을 일일 단위 라벨로 사용하였다.

본 연구에서는 일일 단위 게임봇 탐지 판단 방식을 제안하였으며, 유저의 특정 일자에 대한 플레이 로그가 게임봇인지 아닌지의 Ground truth는 확박가 불가능하다는 점을 보완하기 위해 클러스터링 기법을 활용한 일일 단위 게임봇 탐지에 대한 효과를 결론으로 도출하였다.

본 연구에서는 2장에 기존 게임봇 탐지 방법에 대하여 작성하였으며, 추가로 게임봇 탐지에 이용된 클러스터링 기법에 대하여 설명하였다. 3장에서는 실험에 사용된 온라인 게임 데이터에 대한 설명과 로그 추출 방식에 대하여 설명하였다. 해당 일자에 대한 게임봇 유사 정도를 구분하기 위해 클러스터링을 수행하며, 분류를 위한 지표 선택 과정을 정의하였으며 해당 결과를 통해 분류 및 최종적으로 게임봇으로 판별하는 방식을 정의하였다. 4장에서는 클러스터링 및 기계학습 알고리즘을 이용하여 3장에서 정의된 방법론을 통하여 분류된 게임봇 탐지율 및 성능을 계산 및 일일 탐지 기법을 적용하지 않은 결과와 비교하였다. 5장에서는 실험 결과에 대한 결론을 기술하였다.

II. 관련연구

게임 로그 중 특정 유저군에 포커스를 맞추어 분류 연구를 진행한 사례가 존재한다[1]. 해당 연구에서는 성장 유형에 따른 게임 봇 탐지 방법론에 대해서 연구가 되었다. 성장 유형을 제한하기 위하여 83일의 기간 중 캐릭터 생성부터 최고레벨을 달성한 유저들을 대상으로 연구를 진행하였으며, 성장 관련 로그들을 중심으로 클러스터링을 진행하였다. 다른 연구로는 플레이 유저 전체를 기준으로 클러스터링을 진행하여 각 그룹 안에서 분류를 진행한 연구가 존재한다[2]. 두 연구 모두 핵심 분류 방식 중 클러스터링 기법을 사용한 연구로, 본 논문에서 제안한 방법론과 유사성이 크기 때문에 비교를 해보면, 그 결과는 Table 1.과 같다.

기타 다른 연구에서는 클라이언트 측에서 수행할 수 있는 게임봇 탐지 수단에 대하여 연구한 방식[3, 4]이 있으나, 이러한 방식은 클라이언트를 통해 사용자들에게 배포가 되어야 하기 때문에 클라이언트 위변조, 개인정보 수집, 분석 기법 노출 등의 문제가

Table 1. Comparison with Past Clustering Methodology

Comparison	Contents		
Past Works	J. Lee[1]	Y. Chung[2]	This Paper
Dataset	AION	Yulgang Online	AION
Classification target	part of users	total users	total users
Key Methodology	Clustering, selection	Clustering	Clustering, daily detection
MAX results	Precision (0.937)	F1-score (0.962)	F1-score (0.945)

있어 잘 사용되지 않는 방식이다. 서버에서 게임봇을 탐지하는 방식은 정확도가 높다는 점과 함께 보안 문제가 발생하지 않는다는 점이 있어 많이 사용된다. 해당 방식은 서버 측으로 수집되는 로그를 이용하여 해당 로그를 토대로 정보를 가공하여 분석하는 방식이다. 이동 패턴의 차이를 연구하는 방식[5, 6], 파티나 거래, 귓속말 등 소셜 네트워크를 기본으로 분석하는 방식[7, 8, 9], 게임머니의 이동을 분석하는 방식[10], 자기 유사도를 통한 분석 등의 방식[11] 등이 존재한다.

본 연구에서는 클러스터링 기법을 기반으로 하여

일일 단위의 게임봇 탐지를 통하여 F1-score를 상승시키는 결과를 보여 줄 것이다. 클러스터링 기법을 이용함으로써 봇 유저와 가까운 로그 그룹을 추정하여 분류할 수 있게 될 것이며, 일일 단위로 판단한 게임봇 여부를 통해 최종적으로 해당 유저가 게임봇인지 여부를 판별해 낼 것이다.

III. 게임봇 분류 방법론

3.1 사용 데이터셋

본 연구를 진행하기 위해, 대상 기간인 2010년 5월 8일 ~ 2010년 5월 24일의 17일 간의 유저 행동 로그 분석을 진행한다. 대상 게임은 NCSoft사의 AION이며, 해당하는 기간의 플레이어 로그에 대하여 각 로그별 발생 횟수를 종합하여 지표로 사용하게 된다. 일일 단위 구분은 0시를 기준으로 진행하였다.

3.2 일일 단위 로그 클러스터링 및 분류 프레임워크

본 연구에서 사용하게 된 프레임워크는 Fig.1. 과 같다.

먼저, 일일 로그에 대하여 게임봇에 대한 K-means 클러스터링을 진행한다. K-means 알고리즘의 특징으로는 많은 변량에 대하여 선형 시간복잡도로 알고리즘을 수행한다는 장점이 있고, 분류 결

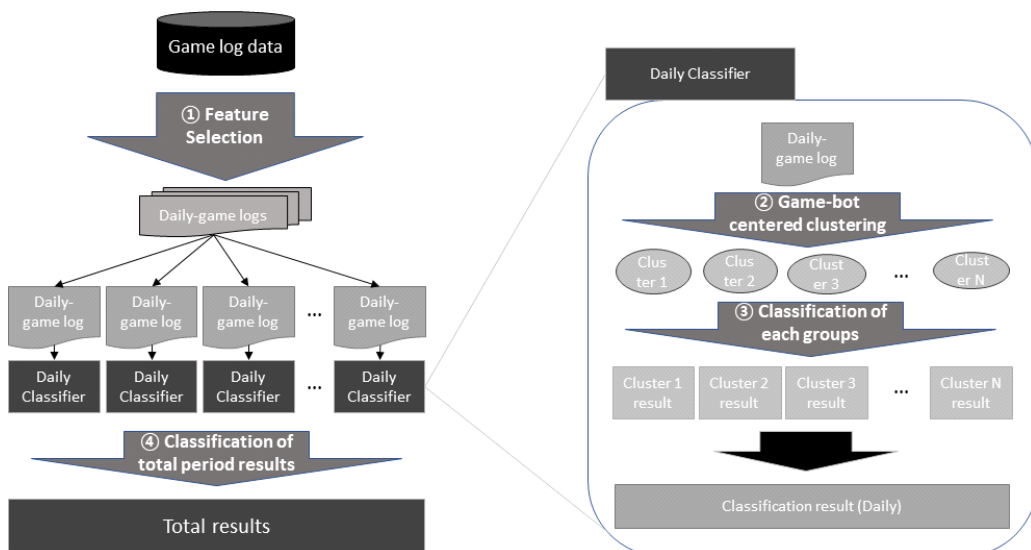


Fig. 1. Classification Framework

과도 선형 시간복잡도 알고리즘에 비해 좋다는 장점이 있다. 선형 연구와는 차별점을 두어 게임봇의 특징을 중심으로 그룹을 묶을 수 있도록 전체 유저가 아닌 게임봇 유저의 로그를 중심으로 클러스터링을 진행한다. 이 때 로그는 일반 유저와 게임봇 유저를 잘 구분할 수 있는, 게임봇 그룹에서 특별하게 많이 발생하는 로그를 지표를 선정하였다. 게임봇 라벨에 대한 클러스터링이 완료되면, 일반 유저에 대한 클러스터 매칭을 진행하여 일반 유저를 게임봇 유저로 클러스터 그룹에 포함시킨다.

일일 단위 로그가 위의 과정을 통하여 K 개의 군집으로 군집화 되었다면, 각각의 클러스터 그룹에 대하여 분류 알고리즘인 SVM(Support Vector Machine) 혹은 RF(Random Forest)를 적용하여 분류기를 학습시킨다. 학습된 K개의 분류기와 클러스터링 센터를 이용하여 일일 단위 샘플이 입력되었을 때 이를 분류할 수 있는 일일 단위 분류기를 제작한다.

마지막으로, 제작된 일일 단위 분류기를 17일의 기간동안 1일 단위로 적용하여 분류 결과를 도출해낸다. 도출해낸 분류 결과를 이용하여, 해당 유저를 최종적으로 게임봇 유저로 판단하는 기준을 설정한다. 설정된 기준으로 테스트를 진행하여, 클러스터링 및 일일 단위 게임봇 유저 판단을 진행하지 않은 비교군과 발생한 성능의 차이를 확인하였으며, F1-score를 통해 결과를 비교 및 분석하였다.

3.3 특징값 추출 및 카운트 표준화

먼저, 주어진 온라인 게임 이용 데이터에 대하여 정상 유저와 게임봇의 로그를 분리한다. 최초 클러스터링에 사용될 로그는 게임봇 유저의 데이터이며, 로그 중에서 Weka를 이용한 feature selection 과정 및 전체 로그 종류를 대상으로 클러스터링을 진행했을 때, 봇 유저 그룹에서 특이점을 보인 로그 종류를 사용하며, 해당 로그 종류는 Table 2. 와 같다.

13가지 로그를 지표로 사용하여 게임봇 유저에 대한 클러스터링을 진행한다. 로그가 발생한 카운트를 이용하여 클러스터링을 진행하지만, 로그 카운트의 분포는 로그 종류에 따라 큰 차이를 가지고 있다. 하루에 1회 발생하는 로그가 있는 반면, 하루 최대 1만회까지 발생하는 로그도 존재한다. 이를 표준화하여 클러스터링 시 격차를 줄임과 동시에 Z-score 등의 표준화 방식에서 없어지는 로그 종류 별 발생횟

Table 2. Clustering Features

Feature	Description
Enter World	Character log-in
Get Experience	Get exp. from various action
Save Char Info	Saves character information periodically(5min.)
Save Item Info	Saves item information periodically(30min.)
Increase Money	Game money increasing(kina)
Create Item	Make item from creation
Get Item	Get item from reward, looting, etc.
Harvest Item	Harvest item from gathering
Equip Item	Equip some equipment
Trade Give	Giving some items to other playing character
Item Get DBID	Creating own distinct ID for items
Use BATTERY	Toggle(On/Off) item which is consumable and assist battle
Update Quest	Updates qusets' current state

수 편차를 보완하기 위해 식(1)을 통해 로그 카운트를 표준화하였다.

$$C_{standard(A,x)} = \frac{\sqrt{C_{A,x}}}{\sqrt{\text{mean}_{x=1..n}(\sqrt{C_{A,x}})}} \quad (1)$$

A : log category x : user
 $C_{K,x}$: Count A of x

또한 선형 연구[1,12]에서 사용되었던 지표를 이용하여 과거 연구와의 비교를 진행하였다. 동일한 지표를 사용함으로써 본 방법론에 사용된 지표가 유효하다는 것을 확인해 볼 것이다.

3.4 게임봇 유형 클러스터링

1일간의 데이터를 기준으로 13가지 로그 지표를 이용하여 게임봇 유저 클러스터링을 K-means 알고리즘을 이용하여 진행한다. K의 값을 변동해 가면서 테스트를 진행하며 K=10인 경우와 K=15인 경우

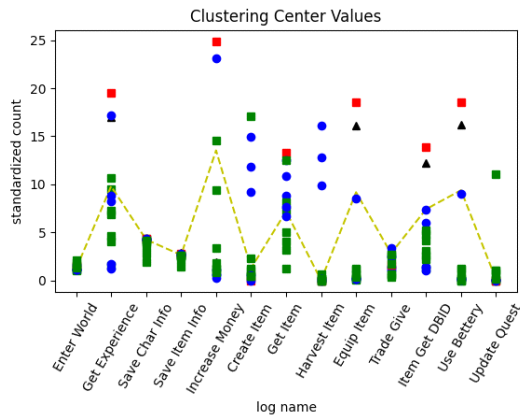


Fig. 2. Clustering centers of each clusters. Red(over 0.99) and black(over 0.65) point contains high game-bot user ratio.

로 테스트하였다.

해당 결과를 토대로 각 클러스터 그룹의 중심을 기록하여 추후 일반 유저 분류를 위해 사용할 수 있도록 하였다. 클러스터링을 통해 나온 결과를 바탕으로 정상 유저에 대해서도 같은 기준으로 클러스터 그룹 매칭을 진행하였다. 결과로 나오는 클러스터 그룹의 경우 여러 특징을 가지는 그룹들이 나오게 되며 각 그룹별로 일반 유저와 게임봇 유저의 비율은 다양한 분포를 띄게 된다. 클러스터링 그룹 K=15인 경우, Fig.2. 와 같은 분포 및 특징을 가지게 된다.

3.5 클러스터 그룹에 대한 분류기 학습

일일 단위 로그에 대하여 클러스터링이 진행되었다면 이후 각각 클러스터 그룹에 대하여 분류기를 학습시킨다. 본 연구에서 사용된 분류 알고리즘으로는 SVM(Support Vector Machine), RF(Random Forest) 두 종류를 사용하여 결과를 출력하였다.

해당 분류기는 각 클러스터 그룹마다 하나씩 생성된다. SVM의 특성 상 학습에 사용된 데이터의 라벨이 한 종류밖에 없을 경우 경계에 사용되는 벡터를 계산할 수 없기 때문에 해당 라벨로 매칭될 수 있도록 변경하였다.

3.6 일일 단위 유저 행위 분류

3.5를 이용해 나온 결과를 바탕으로 일일 분류 결과를 출력한다. 임의의 플레이어의 일일 플레이 로그

가 입력으로 들어오면 해당 유저의 1일치 로그를 이용하여 해당 일자의 어느 한 게임봇 유형에 매칭시킨다. 그 다음 해당 게임봇 유형에 대하여 분류기를 적용한다. 출력되는 판단 결과가 특정 일자에 대한 플레이어의 게임봇 여부를 판별한다.

3.7 전체 기간에 대한 게임봇 추정

3.6을 통해 나온 일일 단위 분류 결과를 종합하여 통합 분류기를 제작한다. 각 일자에 대하여 로그 하나라도 기록되어 있는 정상 유저, 게임봇 유저, 테스트 유저를 추출한다. 추출된 유저 정보를 일일 단위 분류기에 적용을 하여 해당 기간 테스트 유저에 대하여 라벨을 출력한다. 이 과정을 17일 각각에 대하여 하루 단위로 적용한 뒤 결과를 종합한다. 17일 중 T일 이상 게임봇으로 분류된 유저를 종합적으로 게임봇으로 추정한다. 테스트 결과, T=1,2에 대하여 의미 있는 값이 나왔기 때문에 이에 대한 결과를 기록하였다. 그 결과에 대하여 F1-score를 계산하고 본 논문에서 제안한 방법론이 분류 성능 향상에 미치는 영향을 추정한다.

$$F1-score = \frac{2 * Precision * Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}$$

$$TP: True Positive \quad FP: False Positive \quad (2)$$

$$FN: False Negative$$

식(2)는 F1-score 계산식을 나타내며 비교군의 경우 17일 전체 기간을 통합하여 로그 카운트를 종합한 뒤 분류를 수행하였으며, 클러스터링 과정 없이 진행하였다.

IV. 실험 결과

4.1 실험환경

본 실험에 사용된 테스트 데이터 라벨 수는 일반 유저 라벨 14885개, 봇 유저 라벨 1653개로 총 16538개의 라벨을 사용하였다. 구현을 위하여 Python 언어를 사용하였으며, 내부 패키지 중 하나인 sklearn 패키지에 존재하는 알고리즘을 사용하였다. SVM의 경우 sklearn.svm.SVC를 통해 분

류기를 제작하였으며, Random Forest 의 경우 sklearn.ensemble.RandomForestClassifier 패키지를 사용하였다.

4.2 실험 결과 측정 방법

해당 유저 라벨들에 대하여 10-fold validation을 적용하였으며, 평가를 위한 지표는 F1-score를 이용하였다. 테스트를 위하여 파라미터를 변경하면서 결과를 측정하였으며, 변경된 파라미터는 다음과 같다.

클러스터링 그룹 수인 $K=10,15$, 게임봇 추정으로 결론티를 수 있는 임계값 $T=1,2$, 분류 알고리즘 2가지 종류인 SVM, RF에 대하여 변경을 하며 실험 결과를 측정하였다.

또한 본 논문에서 제안한 방법론의 효과를 확인하기 위하여 같은 테스트 데이터 및 특징값을 이용하되 일일 단위 판별 및 클러스터링 기법을 적용하지 않고 전체 기간에 대하여 SVM, RF 분류 알고리즘을 바로 적용한 비교군에 대하여 F1-score를 함께 측정하였다. 실험 결과는 Table 3. 과 같다.

추가적으로, 기존 논문과의 비교를 위하여 과거 두 선행연구[1,12]에서 사용되었던 특징값을 이용하

여 같은 방식으로 클러스터링을 진행하였다. 비교를 위한 부분으로, 본 논문에서 우수한 점수를 보였던 조건인 $K=10, T=2$, RF 분류 알고리즘에 대한 점수를 추가하였다.

4.3 실험 결과 및 시사점

SVM의 경우 T 값에 따라서 Precision과 Recall의 값이 바뀌기는 하지만, F1-score의 경우에는 큰 변화가 없었으며, 본 연구에서 제안한 방식보다 적용하지 않은 방식이 오히려 결과가 더 높은 것을 확인할 수 있었다. 이는 SVM 알고리즘이 본 연구에서 제안한 방식에 적절하지 않았다고 생각된다.

RF의 경우에는 $K=10,15$ 인 경우 모두 $T=1$ 보다 $T=2$ 인 경우에 F1-score의 값이 유의미하게 상승한 것을 알 수 있다. 게임봇으로 판별하는 기준이 올라감에 따라, False Positive는 감소하였으나 False Negative가 증가하여 Precision과 Recall 값이 증감하였다. 로그 카운트 기반 게임봇 탐지 연구의 경우 일반적으로 Precision보다 Recall이 낮은 경우가 대부분이었으며, 이러한 이유로 조화평균인 F1-score의 경우 낮아지는 경우가 많이 발생하였다.

본 연구에서 제시한 방법에 따르면 더욱 자세하게 게임봇 기준을 분류함으로써 Recall 값이 높아짐을 확인할 수 있었다. 특히 RF 알고리즘을 사용하며 $K=10, T=2$ 인 경우 실험 중 가장 높은 F1-score를 보였으며, 방법론을 적용하지 않은 비교군의 결과인 0.898에 비해 0.945로 증가된 것을 확인할 수 있다.

추가적으로 선행 연구에서 사용되었던 특징값을 이용하여 마찬가지로 방식으로 F1-score를 측정하였다. 그 결과 J. Lee[1]의 논문에서는 제안한 특징값을 적용한 것과 적용하지 않은 것에 큰 차이가 없었으며 두 경우 모두 가장 높은 점수에는 미치지 못하였다. 특이점으로는 Baseline으로 사용된 방식 중 가장 높은 점수인 0.918을 보였다는 것이다. 이는 해당 논문에서 사용된 특징값이 본 논문의 방법론에는 적절한 값이 아니었지만 특징값 자체로 가지는 분류 성능에는 큰 영향을 주었다고 생각할 수 있다. 두번째로, A. R. Kang[12]의 논문에서 제안한 특징값을 이용하였을 때에는 본 논문에서 제안한 특징값에 비하여 F1-score가 낮게 나오는 것을 확인할 수 있다. 이는 해당 논문에서 제안한 특징값은 보다

Table 3. 10-fold Validation Results with Baseline(Disapplying Proposed Method)

Parameters	Precision	Recall	F1-score
SVM ($K=10, T=1$)	0.802	0.866	0.833
SVM ($K=10, T=2$)	0.943	0.752	0.836
SVM ($K=15, T=1$)	0.772	0.876	0.821
SVM ($K=15, T=2$)	0.947	0.754	0.839
RF ($K=10, T=1$)	0.888	0.952	0.919
RF ($K=10, T=2$)	0.986	0.908	0.945
RF ($K=15, T=1$)	0.859	0.948	0.901
RF ($K=15, T=2$)	0.980	0.900	0.938
SVM (Baseline)	0.952	0.754	0.841
RF (Baseline)	0.984	0.828	0.898
Past Work[1] SVM ($K=10, T=2$)	0.978	0.863	0.916
Past Work[1] SVM (Baseline)	0.978	0.866	0.918
Past Work[12] SVM ($K=10, T=2$)	0.987	0.897	0.940
Past Work[12] SVM (Baseline)	0.968	0.816	0.885

일반적인 값을 제안한 것이지만 본 논문에서 사용된 지표는 제안한 방법에서 좋은 성능을 발휘할 수 있도록 클러스터링을 더 잘 해줄 수 있는 지표를 선별하였기 때문이라고 생각할 수 있다.

V. 결 론

본 연구에서는 게임봇 탐지를 위해 일일 단위로 게임봇을 추정하는 방식에 대하여 제안하였으며, 이를 실제 온라인 게임 플레이 로그를 이용하여 실험 및 검증을 진행하였다. 일부 분류 알고리즘에 대하여 큰 성능 향상을 기대할 수 있는 것으로 확인이 되었으며, 일일 단위 탐지를 통한 방식이 유의미함을 보였다. Random Forest 알고리즘을 사용하여 일일 단위로 탐지를 진행 하였을 때 비교군에 비해 F1-score가 0.898에서 0.945으로 탐지 성능 향상하였다는 것을 보여주었다. 또한 과거 선행연구와의 비교를 통하여, 본 논문에서 제안한 방식 및 특징값 선정의 우수성을 확인하였다.

본 연구에서 제안한 일일 단위 게임봇 탐지 방법은 특정 유저 그룹만을 추출하여 탐지하는 방식이 아닌 전체 유저에 대한 탐지 방식으로 범용적이며, 기타 선행 연구에서 제안한 방법들에 대하여 혼합하여 적용할 수 있다. 또한 게임봇 행동 양상과 일반 유저 행동 양상이 섞여 있는 플레이어 및 새로 생성된 계정 등 플레이 스타일이 탐지 기간 동안 불규칙한 경우에 대하여 큰 효과를 발휘할 수 있을 것으로 기대된다.

References

- [1] 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 & Cryptology*, 25(5), pp. 1077-1084, Oct. 2015.
- [2] Y. Chung, C. Park, N. 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*, 35(6), pp. 1058-1067, Dec. 2013.
- [3] A.R. Kang, J. Woo, J. Park and H.K. Kim, "User behavior analysis for online game bot detection," *Journal of The Korea Institute of information Security & Cryptology*, 22(2), pp. 225-238, Apr. 2012.
- [4] R.V. Yampolskiy, and V. Govindaraju, "Embedded noninteractive continuous bot detection," *Computers in Entertainment*, vol. 5, no. 4, pp. 1-11, Mar. 2008.
- [5] K.T. Chen, A. Liao, H.K.K. Pao and H.H. Chu, "Game bot detection based on avatar trajectory," *Entertainment Computing ICEC, LNCS 5309*, pp. 94-105, Sep. 2008.
- [6] S. Mitterhofer, C. Platzer, C. Kruegel and E. Kirda, "Server-side bot detection in massive multiplayer online games," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 29-36, Jun. 2009.
- [7] A.R. Kang, H.K. Kim and J. Woo, "Chatting pattern based game bot detection: do they talk like us?," *KSII Transactions on Internet and Information Systems*, 6(11), pp. 2866-2879, Nov. 2012.
- [8] K. Woo, H. Kwon, H.C. Kim, C.K. Kim and H.K. Kim. "What can free money tell us on the virtual black market?," *ACM SIGCOMM*, vol. 41, no. 4, pp. 392-393, Aug. 2011.
- [9] 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*, pp. 1825-1834, Apr. 2018.
- [10] H. Kwon, K. Woo, H. Kim, C.K. Kim, and H.K. Kim, "Surgical strike: a

- novel approach to minimize collateral damage to game bot detection,” Annual Workshop on Network and Systems Support for Games, pp. 1-2, Dec. 2013.
- [11] 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 Proc. NDSS Internet Society, pp. 1 - 15, May 2017.
- [12] 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, pp. 523, Dec. 2016.

〈저자소개〉



김 주 환 (Joo Hwan Kim) 정회원
 2019년 2월: 고려대학교 사이버국방학과 졸업
 2020년 3월~현재: 고려대학교 정보보안학과 석·박사통합과정
 <관심분야> 정보보호, 정형검증, 데이터 보안



최 진 영 (Jin-Young Choi) 종신회원
 <관심분야> 정형기법, 시큐어 소프트웨어공학, 소프트웨어보증
 1982년: 서울대학교 컴퓨터공학과 학사
 1993년: 미국 University of Pennsylvania 전산학 박사
 1996년~2015년: 고려대학교 컴퓨터학과 교수
 2015년~현재: 고려대학교 정보보호대학원 교수
 <관심분야> 정형기법, 소프트웨어보안, 시큐어소프트웨어공학, 소프트웨어보증