

## 마우스 동작 기록 기반 비정상 게임 이용자 감지를 위한

### 단일 클래스 분류 기법

송민준<sup>0</sup>, 김인기\*, 김범준\*, 전영훈\*\*, 곽정환(교신저자)\*\*

<sup>0</sup>한국교통대학교 소프트웨어학과,

\*한국교통대학교 교통에너지융합학과,

\*\*한국교통대학교 소프트웨어학과

e-mail: totonim31@gmail.com<sup>0</sup>, cv2@kakao.com\*, zhskfkseh@gmail.com\*,

0hoon.jeon@gmail.com\*\*, jgwak@ut.ac.kr\*\*

## One-Class Classification based on Recorded Mouse Activity for Detecting Abnormal Game Users

Minjun Song<sup>0</sup>, Inki Kim\*, Beonjun Kim\*, Younghoon Jeon\*\*, Jeonghwan Gwak(Corresponding Author)\*\*

<sup>0</sup>Dept. of Software, Korea National University of Transportation,

\*Dept. of IT·Energy Convergence, Korea National University of Transportation,

\*\*Dept. of Software, Korea National University of Transportation

### ● 요약 ●

최근 온라인 게임 산업이 급속도로 확장됨과 더불어 Gamebot과 같은 비정상적인 프로그램으로 인한 게임 서비스 피해사례가 급격하게 증가하고 있다. 특히, 대표적인 게임 장르 중 하나인 FPS(First-Person Shooter)에서 Aimbot의 사용은 정상적인 이용자들에게 재미 요소를 잃어버리게 하고 상대적 박탈감을 일으켜 게임의 수명을 줄이는 원인이 된다. 비정상 게임 이용자의 근절을 위해서 메모리 변조 및 불법 변조 프로그램 접근 차단 기법과 불법 프로그램 사용의 패턴 모니터링과 같은 기법들이 제안되었지만, 우회 프로그램 및 새로운 패턴을 이용한 비정상적인 프로그램의 개발에는 취약하다는 단점이 있다. 따라서, 본 논문에서는 정상적인 게임 이용자의 패턴만 학습함으로써 비정상 사용자 감지를 가능하게 하는 딥러닝 기반 단일 클래스 분류 기법을 제안하며, 가장 빈번하게 발생하는 치트(Cheat) 유형인 FPS 게임 내 Aimbot 사용 감지에 초점을 두었다. 제안된 비정상 게임 이용자 감지 시스템은 정상적인 사용자의 마우스 좌표를 데카르트 좌표계(Cartesian coordinates)와 극좌표계(Polar coordinates)의 형태로 패턴을 추출하는 과정과 정상적인 마우스 동작 기록으로 부터 학습된 LSTM 기반 Autoencoder의 복원 에러에 따른 검출 과정으로 구성된다. 실험에서 제안된 모델은 FPS 게임 내 마우스 동작을 기록한 공개 데이터셋인 CSGO 게임 데이터셋으로 부터 학습되었으며, 학습된 모델의 테스트 결과는 데카르트 좌표계로부터 훈련된 제안 모델이 비정상 게임 이용자를 분류하는데 적합함을 입증하였다.

**키워드:** 이상 탐지(anomaly detection), 게임봇(Gamebot), One Class Classification, Autoencoder, 데카르트 좌표계(Cartesian coordinates), 극좌표계(Polar coordinates)

## I. Introduction

전 세계에서 인터넷과 PC의 보급으로 인하여 게임에 대한 접근성은 점점 증가하는 추세이다. 게임을 하는 이용자가 증가함에 따라 게임 산업은 점점 더 크게 발전하고 있다. 그중 First Person Shooter(FPS)는 표 1과 같이 게임에서 가장 큰 이익을 받는 장르이다. 하지만 인기를 받는 만큼 Gamebot과 같은 불법 변조 프로그램에 대한 취약성

은 항상 존재한다. Gamebot과 같은 시스템적으로 비정상적인 도움을 받는 이용자들을 치트유저로 정의하며, 비정상 이용자로 인한 재미를 잃어버린 플레이어들의 감소는 게임개발사의 직접적인 영향을 미치게 된다. 이에 게임개발사는 Gamebot과 같은 시스템 플레이를 제한하고, 비정상 이용자를 분류하기 위해 Anti-Cheat 프로그램을 도입하여

원천적으로 접근을 제한하고 있다. 하지만, Anti-Cheat를 우회하는 비정상 이용자를 검출하는 방식은 S/W와 인력을 함께 투입하여 검출하는 방식이며 이러한 방식은 막대한 시간적, 경제적 비용이 발생한다. 따라서, 게임개발사는 비정상 사용자 탐지에 비용을 최소화 하기 위해 다양한 정책을 적용하였지만, 지속적으로 발전하는 치트 엔진과 Anti-Cheat 대응을 위한 한정적인 자원으로 인해 실시간 대처성의 부족함과 예방 정책의 한계를 보여준다. 이러한 문제점들을 해결하기 위해 본 논문에서는 Gamebot의 한 종류인 Aimbot을 탐지하기 위해 정상적인 이용자의 마우스 좌표를 데카르트 좌표계와 극좌표계 거리로 변화하여 마우스 동작 기록의 분포를 학습시켜 비정상 이용자를 탐지하는 Autoencoder 기반 딥러닝 모델을 제안한다.

Table 1. 2021 장르별 e스포츠 실태 조사[1]

게임 장르	종목	선수	비중
FPS (First-Person Shooter)	배틀그라운드	58	49.8
	오버워치	56	
	레인보우식스	46	
	발로란트	38	
	포트나이트	8	
MOBA (Multiplayer Online Battle Arena)	리그오브레전드	91	25.8
	와일드리프트	16	
Racing	카트라이더	41	9.9
RTS (Real-Time Strategy)	스타크래프트	15	3.6
기타	-	45	10.9
합계	-	414	100

## II. Related Works

기존 지능형 Anti-Cheat 개발을 위한 연구사례로 궤도 데이터를 통한 Support Vector Machine(SVM)과 K-Nearest Neighbor(KNN)와 같은 머신러닝 기반 Gamebot 분류 기법과[2] 심층학습 기반 방법론을 사용하여 Gamebot을 탐지하는 연구가 진행되었다[3]. 하지만 이러한 방법론들은 한정적인 도메인에서만 적용된다는 한계점이 존재한다. 또한, 마우스의 동작 기록은 물리적인 위치에서 변화하기 때문에 좌표계의 전처리를 통한 특징 생성 단계는 매우 중요하다. 따라서, 본 논문에서는 FPS에서 공통으로 사용하는 마우스와 타겟 좌표를 이용하여 시계열(Timeseries) 데이터 기반 Long-short Term Memory(LSTM) 기반 Autoencoder 구조에 적용하여 Aimbot을 한정적인 도메인이 아닌 여러 Task에서도 강인하게 동작하는 딥러닝 모델을 구축하기 위한 연구를 진행하였다. 또한, 데이터 전처리 과정에서 물리적인 X, Y 좌표를 그대로 사용하는 것이 아니라, 다양한 좌표계에 적용하였으며, 데카르트 좌표계가 비정상 사용자를 추출할 때 적절한 특징을 생성하는 효과를 부여하는 것을 확인할 수 있다.

## III. Proposed Method

### 1. Dataset

#### 1.1 Coordinate Transformation

데카르트 좌표계와 극좌표계를 선정한 이유는 데카르트 좌표계는 우리가 쉽게 유클리드 공간을 나타내는 좌표계이며 게임 개발 엔진에서 제공하는 마우스 좌표 입력값 역시 X, Y 데카르트 좌표로 설정으로 한다. 따라서 게임엔진 로그 데이터를 가져왔을 때 바로 적용할 수 있기 위해 데카르트 좌표계를 선정하였다. 극좌표계는 이전 데이터와 현재 데이터의 차이를 통해 나타낼 수 있는 것이 한정적이다. 하지만 극좌표계로 표현한다면 반지름과 각도로 방향과 힘의 크기로 나타낼 수 있다. 현재 마우스 방향의 움직임에 대한 작용하는 힘의 크기와 방향을 얻을 수 있다. 따라서 2가지의 좌표계를 선택하여 실험을 진행하였다. Fig. 1은 극좌표계에서 (A)는 Training data, (B) Test data이다. 초록색은 Normal class이며, 파란색은 Abnormal class에 해당하는 데이터이다. Fig. 2는 (A) 데카르트 좌표 기반 L2-Distance이며, (B)는 극좌표 기반 L2-Distance의 샘플을 나타낸다.

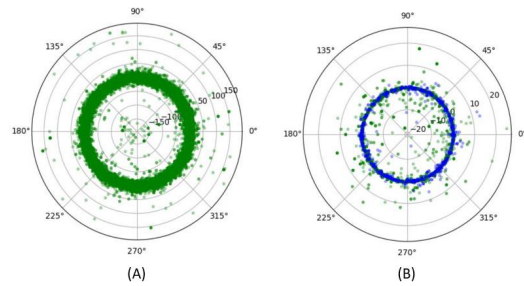


Fig. 1. (A) Training data (B) Test data

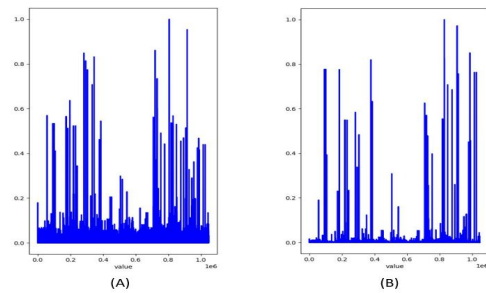


Fig. 2. (A) Cartesian, (B) Polar Based L2-Distance data sample

Table 2. Dataset Description

Type	Data shape (Number of data, Timesteps, Input size)
Training	(943260, Timesteps, 1)
Validation	(104806, Timesteps, 1)
Test	(4389, Timesteps, 1)

(A)				(B)			
UXposition	UYposition	TXposition	TYposition	UXposition	UYposition	TXposition	TYposition
0	0.011	80.02	21.573	0	0	0.741	-11.744
-0.016	0.016	80.132	21.663	0	0	0.741	-11.548
0	0	80.29	21.759	0	0	0.741	-11.315
0	-0.016	80.422	21.963	0	0	0.741	-11.136
-0.011	-0.077	80.527	22.142	0	0	0.741	-10.995
-0.055	-0.17	80.549	22.268	0	0	0.742	-10.885
-0.049	-0.264	80.63799	22.371	0	0	0.742	-10.793
0	-0.385	80.827	22.396	0	0	0.742	-10.744
0.027	-0.209	81.013	22.957	0	0	0.742	-10.691
0	0.038	81.111	23.893	0	0	0.742	-10.648
0	0	81.227	24.557	-0.275	-0.275	0.193	-11.119
0	0	81.338	25.299	0	-0.829	0.468	-12.454
0	0	81.448	26.006	0	-0.824	0.468	-13.187
0	0	81.563	26.609	0	-0.879	0.468	-14.053

Fig. 3. 데이터 예시. (A) Normal (B) Abnormal

### 1.2 데이터 수집 및 전처리

본 논문에서 제안하는 One-Class Classification LSTM Autoencoder 기반 마우스 동작 기록을 통한 비정상 사용자 탐지 모델을 학습하기 위해 Kaggle에 공개된 "CSGO cheating dataset"을 사용하였다. 비정상 사용자가 가지는 데이터는 현실 세계에서 직접 Labeling 하기 어려우므로, 학습 단계에서는 정상데이터만 입력하고, 추론 단계에서는 비정상 사용자 데이터를 입력한다. 학습이 완료된 LSTM Autoencoder는 입력된 데이터를 Reconstruction 하는 과정에서 비정상 사용자의 데이터 분포를 입력받으면, 입력 데이터의 Reconstruction Error가 커짐에 따라, 일정 Threshold를 통해 비정상을 탐지할 수 있게 된다. Fig. 3과 Table 2는 정상 사용자와 비정상 사용자의 마우스 동작 기록 데이터셋 샘플이다. Aimbot과 같은 비정상 사용자는 비인기프로그램의 도움을 통해 정상적인 사용자가 게임 시스템 및 물리적으로 발생시킬 수 없는 마우스 동작 기록 데이터가 발생한다. 이때, 현재 마우스 동작 기록이  $t$ 일 때, 다음 동작 기록인  $t+1$ 에서의 급격한 차이를 One Class LSTM Autoencoder의 입력으로 사용한다. 따라서 데이터 전처리 방법은 아래의 수식과 같다.

$$X : (x_t - x_{t+1}) \quad (1)$$

$$Y : (y_t - y_{t+1}) \quad (2)$$

(1), (2)의 수식은 현재 시간과 다음 시간의 차를 통해 X, Y 좌표를 계산하는 수식이다. 이후 X, Y로 전 처리된 데이터를 통해 L1 및 L2-Distance로 변환하였다. 본 논문에서는 데이터의 좌표를 데카르트 좌표(Cartesian coordinates)와 극좌표(Polar coordinates)로 나누어 실험을 진행하였다.

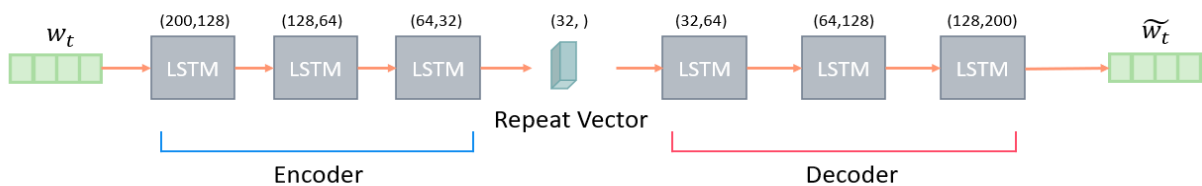


Fig. 4. One Class LSTM Autoencoder Architecture

### 1.3 Timesteps

Timestep이란 사용자와 맵의 상호작용에 대한 업데이트 빈도를 나타내는 수치이다. 즉, 모니터 화면에서 보이는 프레임 수(Frame Per Second)를 넘어서 실제로 서버와 사용자에게 초당 얼마나 해당 프레임대로 명령이 전달되고 있는지에 대한 개념이다. 데이터셋의 Timestep은 192Ticks으로 되어있지만 실제 게임에서는 64Ticks 또는 128Ticks로 진행된다. 따라서 게임에서는 지원되지 않는 Tickrate가 데이터셋에 적용이 되어, 본 논문에서는 Timesteps을 '100, 150, 200, 250, 500, 750'으로 조정하여 진행하였다. 게임이 64Ticks로 구성이 되어있다면 Timesteps이다. Timesteps가 100이면 실제 시간으로 1.5초 정도의 시간이 된다. 실제 시간으로는 짧은 시간이기 때문에 점진적으로 Timesteps을 큰 값으로 증가시켜 실제 시간에 영향을 준 값으로 비웠다. 실험 결과에서는 가장 좋은 성능을 갖는 Timesteps만 서술하였다.

## 2. LSTM Autoencoder 기반 단일 클래스 분류

본 논문에서 One Class LSTM Autoencoder를 이용하였으며 학습에 사용된 하이퍼파라미터는 Table 4와 같다. Loss는 Mean Squad Error를 사용하였고 좌표계마다 Distance의 차이 분포를 이용하여 특징을 학습하였다. 이에 따라 복원 능력에 대한 차이를 이용하여 비정상 사용자를 분류하였다. Table 3은 실험 환경을 나타내는 테이블이다. Fig. 4은 학습에 사용된 모델의 Architecture이며 Table 3과 4는 모델의 최적화를 위한 실험 환경과 하이퍼파라미터 값이다.

Table 3. System Configuration

Item	Value
CPU	Intel i9-12900K
Memory size	128GB
GPU	NVIDIA GEFORCE RTX 3090
Python version	3.8.10
Tensorflow version	2.8.0 + CUDA 11.8
OS	Ubuntu Server 18.04 LTS

Table 4. 모델 최적화를 위한 하이퍼파라미터

Optimizer	Learning Rate	Batch size
Adam	0.0001	1024

### 3. 성능 평가 실험

Table 5는 Timestep을 50단위로 100부터 500까지 설정하여 얻어진 좌표별 Distance 성능의 최적점을 비교한 표이다. 실험 결과에서

보여지듯이, 각 방법론들은 약 80%의 정확도를 보여주며, L1-Distance가 성능 향상에 크게 기여하였다. 또한, Figs. 5과 6은 각각 L1-distance을 기반 데카르트 좌표계와 극좌표계를 이용하였을 때 모델의 confusion matrix이며 대다수의 데이터 샘플들을 올바르게 분류하였음을 보여준다.

Table 5. 제안 모델 성능 평가 실험 결과

Coordinate	Distance	Timesteps	Accuracy	F1-Score
Cartesian	L1	150	0.8216	0.8130
	L2	250	0.7047	0.7159
Polar	L1	200	0.8081	0.8004
	L2	200	0.6322	0.6448

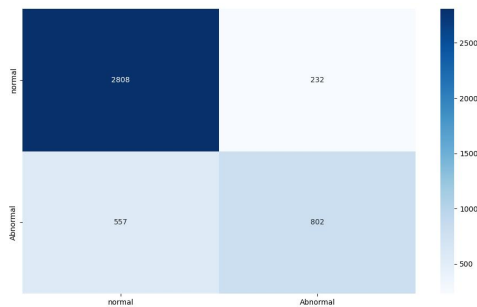


Fig. 5. Confusion matrix based on L1-Distance in Cartesian coordinates

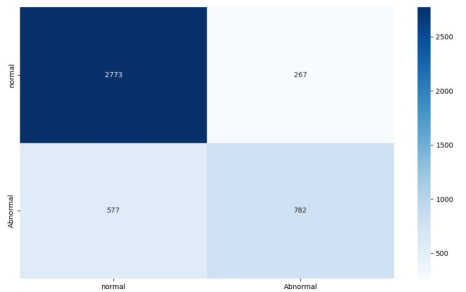


Fig. 6. Confusion matrix based on L1-Distance in polar coordinates

비정상 사용자 분류 기준은 정상데이터의 분포만을 학습하였고, Reconstruction Error의 Threshold를 통해 예측되는 데이터에서 Threshold보다 미만이면 비정상적으로 분류하였다. 비정상 사용자는 마우스를 움직이지 않다가 목표 오브젝트가 화면에 나타나는 순간에 비인가프로그램이 마우스를 급격하게 움직이기 때문에 일반적인 마우스 움직임의 분포를 크게 벗어나는 동향을 가지기 때문에 One Class LSTM Autoencoder 기반 딥러닝 모델이 좋은 성능을 나타내는 것을 확인하였다. 결론적으로, One Class Classification 기법을 사용한 비정상 사용자 분류를 위한 LSTM Autoencoder는 사용자 마우스 좌표를 분석하는데 극좌표계 보다 데카르트 좌표계의 성능이 더 우수하다는 것을 알 수 있다.

#### IV. Conclusions

본 논문에서는 One Class LSTM Autoencoder를 사용하여 사용자 마우스 동작 기록 좌표를 기반으로 한 비정상 사용자 탐지 모델을 제안하였다. 실험 결과는 극좌표계 보다 데카르트 좌표계가 더 좋은 분류 성능을 나타냄을 확인하였다. 본 논문에서 제안한 모델은 FPS 게임 중 한 도메인에 한정된 것이 아니라 사용자 마우스 기록 데이터 기반 모든 FPS 게임에 비정상 사용자를 분류하는 데 활용될 수 있다. 추후 연구에서는, Self-Attention 등의 기법을 추가한 모델을 통한 실험과 효과적인 Augmentation 기법을 연구/개발하며, LSTM 의 Time Series 데이터 기반 모델 성능 평가를 통한 분석을 진행할 것이다.

#### ACKNOWLEDGEMENT

This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education [2021RIS-001(1345341783)]

#### REFERENCES

- [1] T. G. Kim, H. I. Kim, W. M. Kang, D. Y. Jang, M. R. Jeong and S. W. Kim, "The 2021 Survey on the Korean e-Sport Industry." KOREA CREATIVE CONTENT AGENCY, pp. 124, Nov, 2021.
- [2] H. K. Pao, K. T. Chen and H. C. Chang, "Game Bot Detection via Avatar Trajectory Analysis," in IEEE Transactions on Computational Intelligence and AI in Games, vol. 2, no. 3, pp. 162-175, Sept. 2010.
- [3] M. Willman, 'Machine Learning to identify cheaters in online games', Dissertation, 2020.
- [4] P. Perera, P. Oza, and V. M. Patel. "One-class classification: A survey," arXiv preprint, 2021.