

# You are a Game Bot:

## Uncovering Game Bots in MMORPGs via Self-similarity in the Wild

Eunjo Lee (NCSOFT)

Jiyoung Woo (Korea University)

Hyoungshick Kim (Sungkyunkwan University)

Aziz Mohaisen (State University of New York at Buffalo)

Huy Kang Kim (Korea University)



# Contents

Introduction

Feature selection and modeling

Experiments

Model maintenance

Real-World deployment

Conclusions

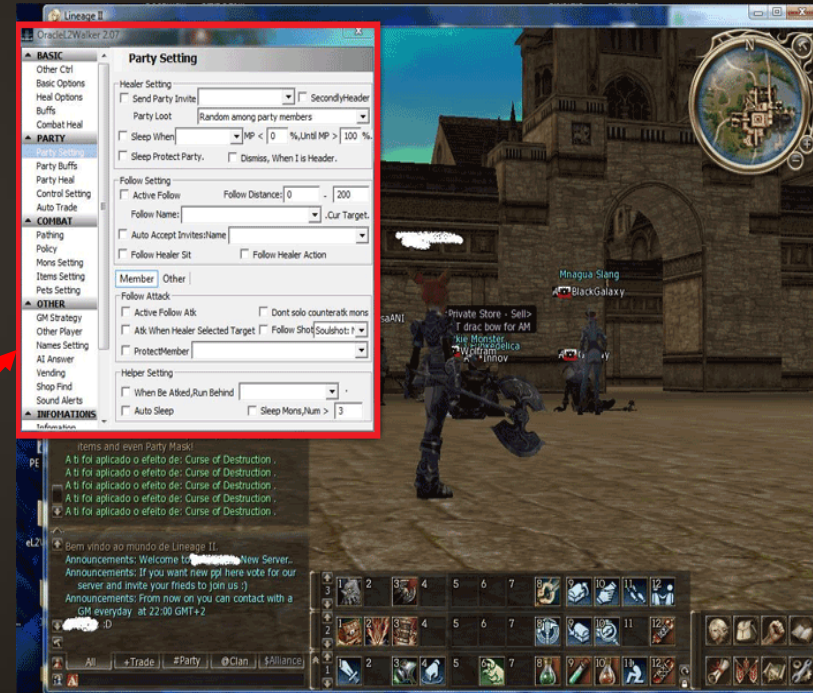
# Introduction

# Introduction

## Game BOT

- Program that plays a game autonomously (instead of human users)

Bot configurations



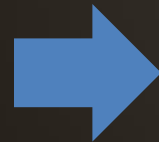
# Introduction

## Real Money Trading (RMT)

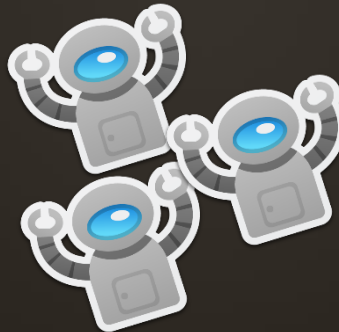
- Collect valuable items and monetize it by trading item to others



Game World



Virtual  
Assets



Virtual  
Assets



Real  
Money



# Introduction

Gold Farming Group (GFG)



# Introduction

Game BOT

[https://www.youtube.com/watch?v=k6tk8\\_R2w08](https://www.youtube.com/watch?v=k6tk8_R2w08)

# Introduction

## Game BOT

- Widespread cheating in online games
  - Collapse of an in-game economy
  - Cause a human users' churn
  - Reduce the revenue





# Introduction

## Countermeasures

- Client-side
  - Bot process detection using anti-malware programs
- Server-side
  - **Bot classification using game log analysis**

# Introduction

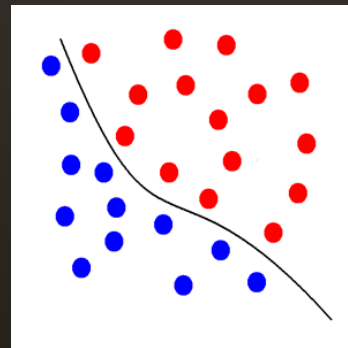
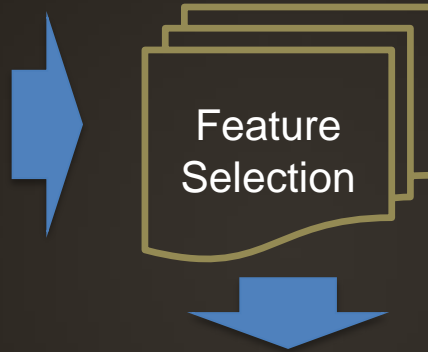
## Machine Learning-based Approach



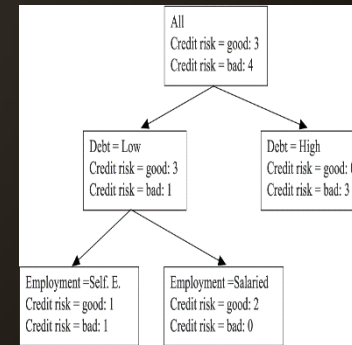
Game Logs

Character ID	T1	T2	T3	Response
686042	0	0	0	0
854209	1	1	1	3
1032131	0	0	0	0
1049483	1	1	1	3
1340479	0	0	0	0
1352850	0	0	0	0
1771815	1	1	1	3
1832497	0	0	0	0
1884884	1	1	1	3
2130576	1	1	1	3
2445903	1	0	0	1

Ground Truth



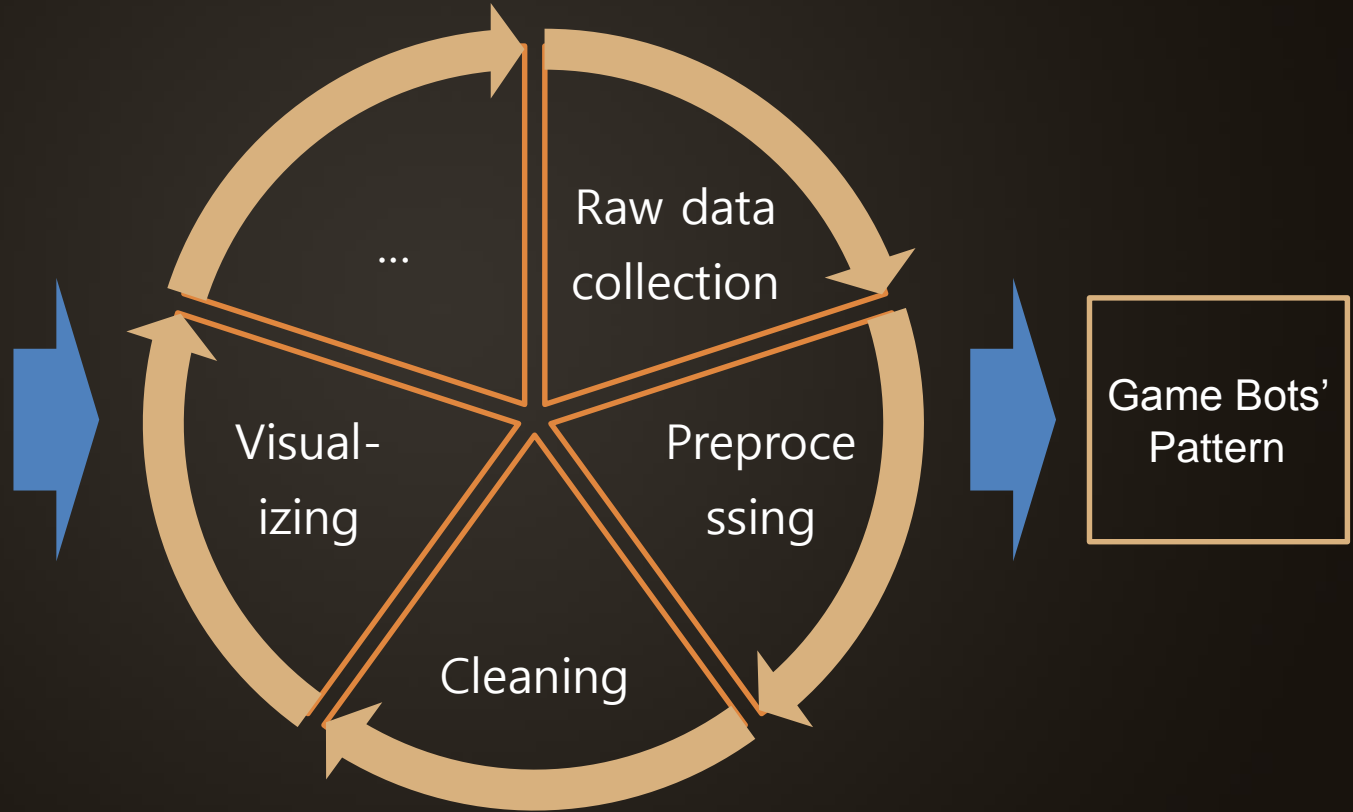
Learning Algorithm



Prediction Model

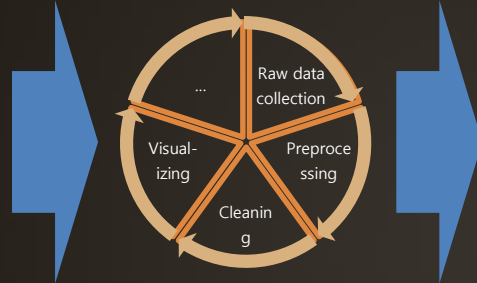
# Introduction

## Challenges



# Introduction

## Challenges



Game A  
Bots'  
Pattern



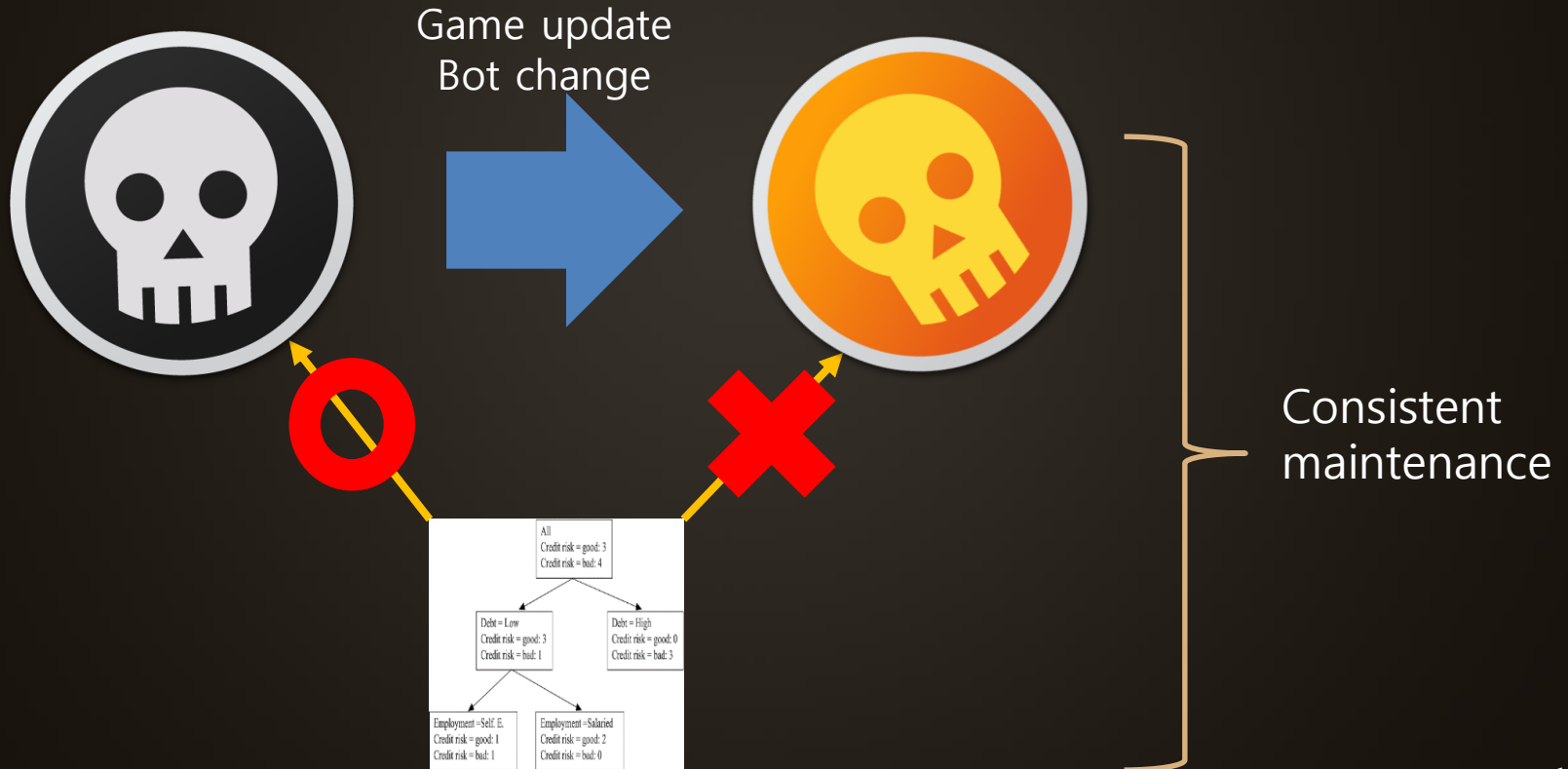
Game B  
Bots'  
Pattern

High cost  
time consuming



# Introduction

## Challenges



# Introduction

## Our proposals

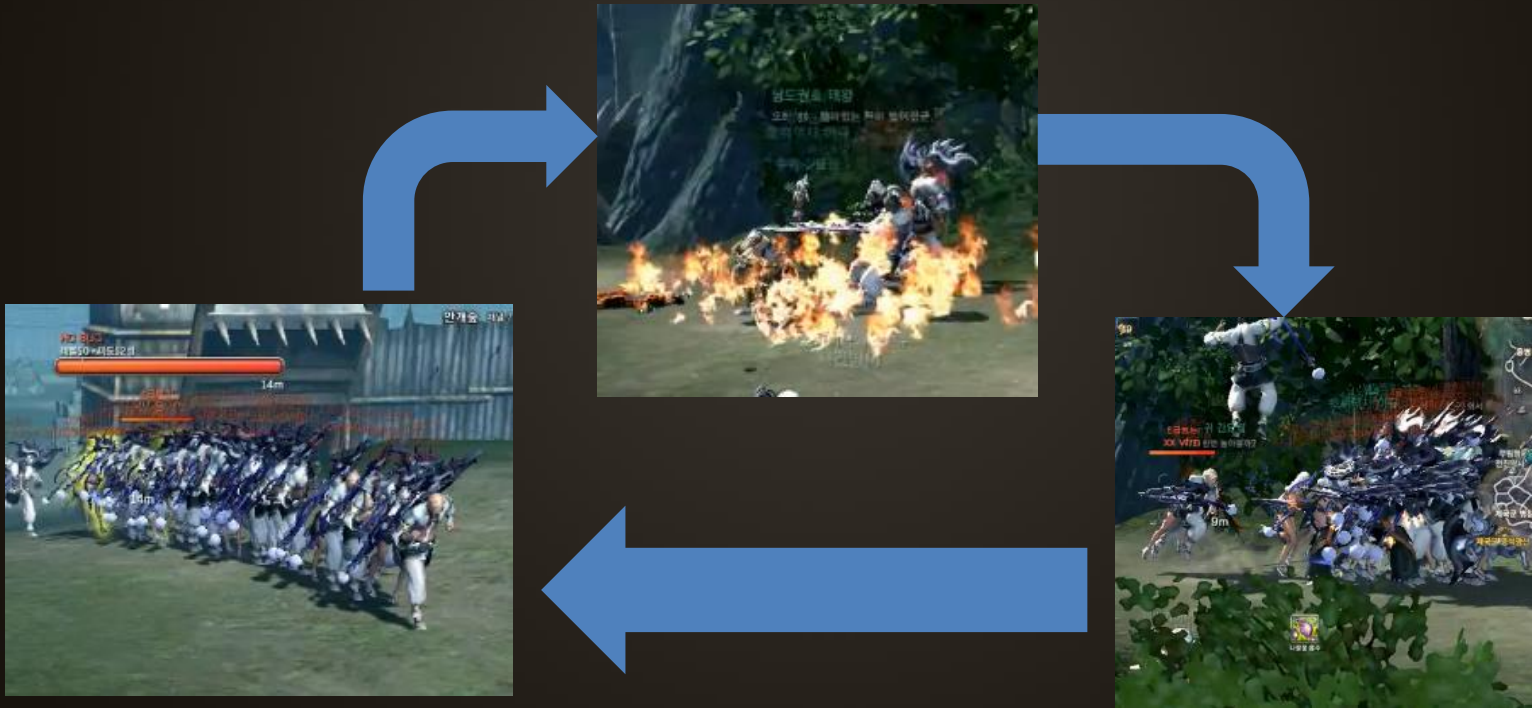
- Using self-similarity as a generic feature
  - Focus on the repetitive activities of game bots, not specific behavior
- Proposing framework to maintain a prediction model autonomously
  - Detect the change in performance of the prediction model and retrain it

# Feature Selection and Modeling

# Self-similarity

## Definition

- Measurement of the similarity of periodic actions per user





# Self-similarity

## Motivation and consideration

- Intrinsic attributes
  - Bot programs **repeat routines** using predetermined settings
  - Human users may exhibit similar behavior, but **not for long period of time**
- Stability
  - **Little effect** of game update or bot program changes
  - Considering **various actions** rather than a single action
- Computing efficiency
  - Easy to apply distributed algorithms (i.e. **MapReduce**) for log processing

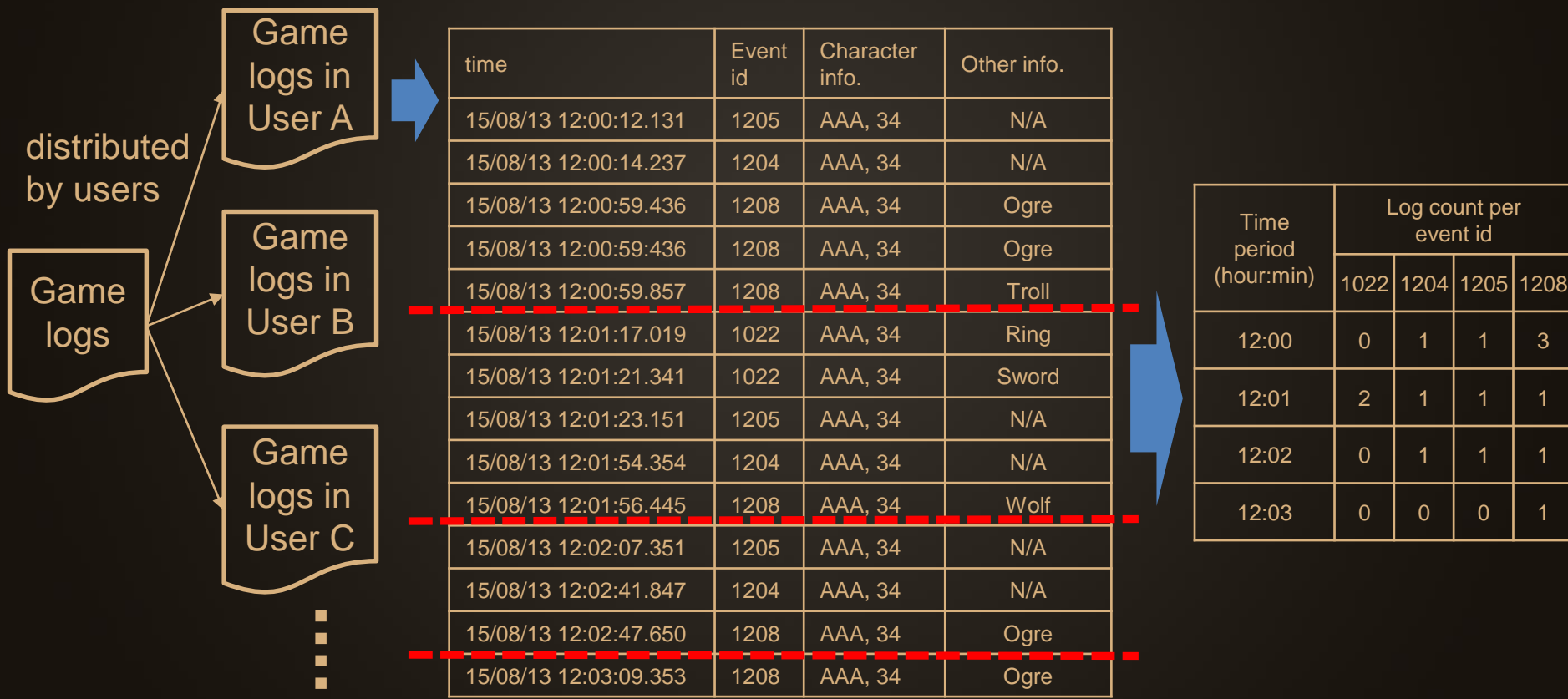
# Self-similarity

## Detailed process

- Generating log vectors
- Measuring cosine similarity
- Measuring self-similarity

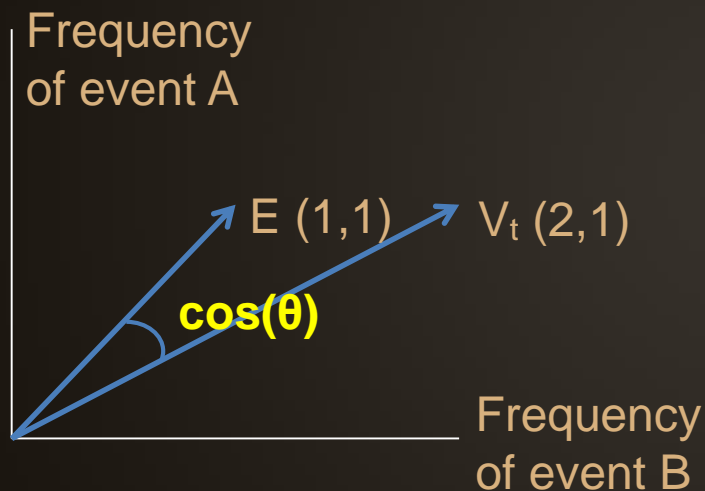
# Self-similarity

## Generating log vectors



# Self-similarity

Measuring the cosine similarity between log vector( $V_t$ ) and unit vector( $E$ )



$$\begin{aligned}\cos(\theta) &= \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum A_i \times B_i}{\sqrt{\sum (A_i)^2} \times \sqrt{\sum (B_i)^2}} \\ &= \frac{(2 \times 1 + 1 \times 1)}{\sqrt{2 \times 2 + 1 \times 1} \times \sqrt{1 \times 1 + 1 \times 1}} \\ &= \frac{3}{\sqrt{5} \times \sqrt{2}} \\ &\approx 0.948\end{aligned}$$

# Self-similarity

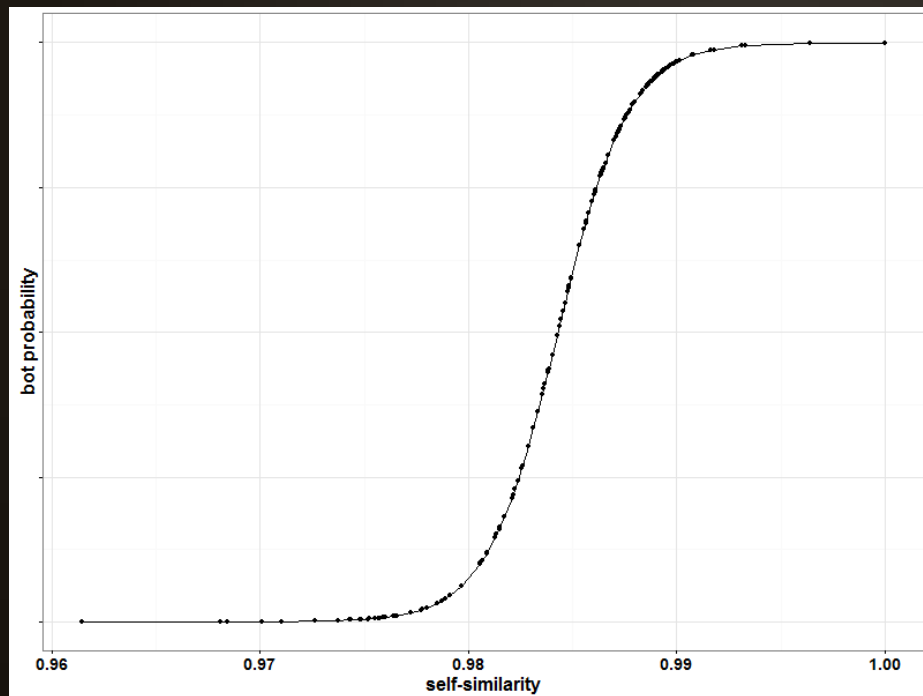
## Measuring self-similarity

- Measuring std. of cosine similarity and transforming using the following model
  - $H = 1 - \frac{1}{2}\sigma$ , ( $0.5 \leq H \leq 1$ ,  $\sigma$ : *std. deviation of cosine similarity*)

# Modeling and Evaluation

## Modeling

- Logistic regression
  - Calculating the probability of a character being a game bot



# Experiments

# Experiments

## Datasets



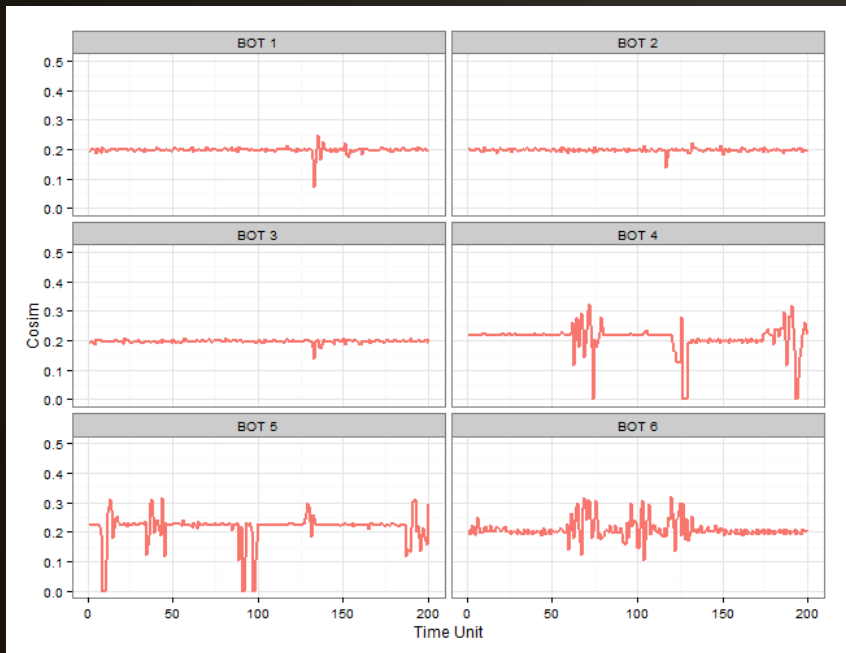
	Lineage	Aion	B&S
Release year	1997	2008	2012
Daily active users	300K	200K	100K
Concurrent users	150K	80K	50K



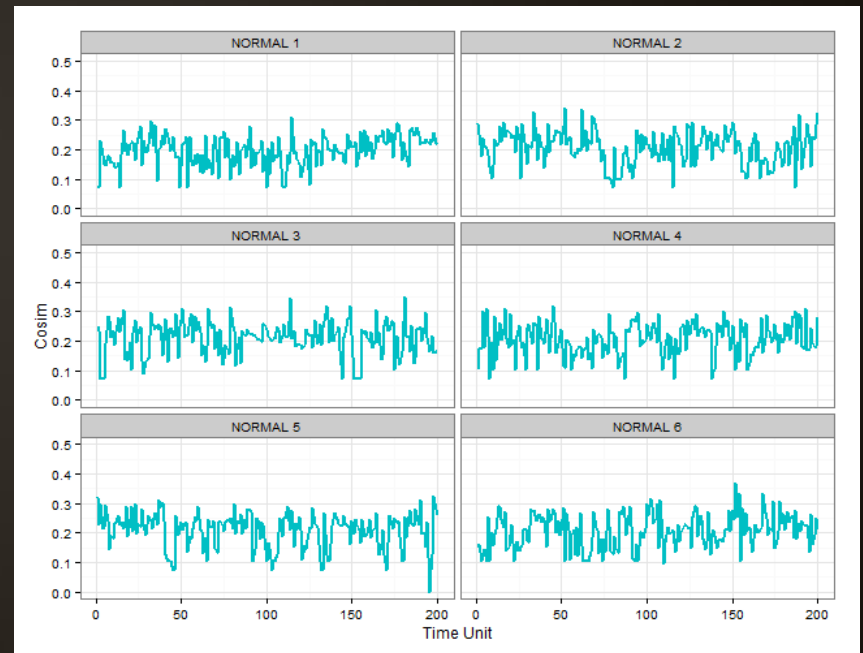
# Experiments

## Cosine similarities

- Bots have cosine similarities with fewer variations than human users



Bots

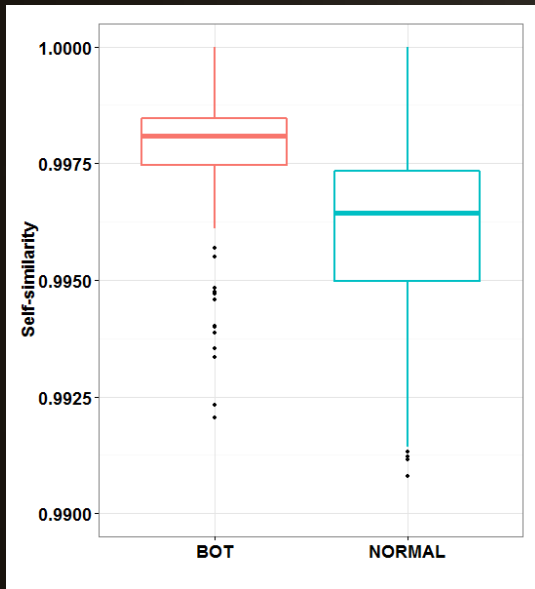


Humans

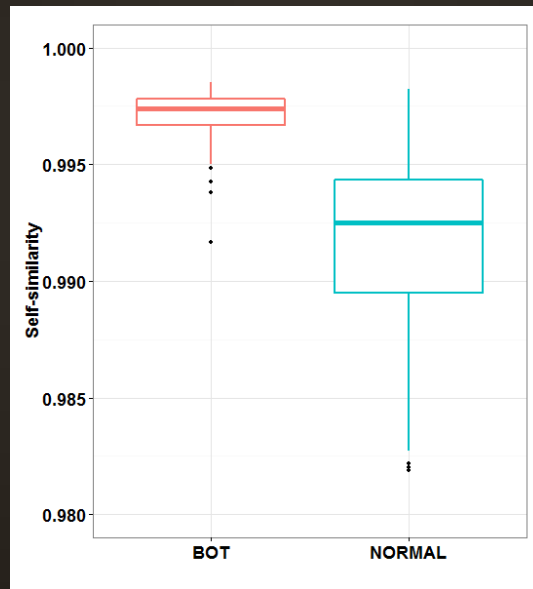
# Experiments

## Self-similarity

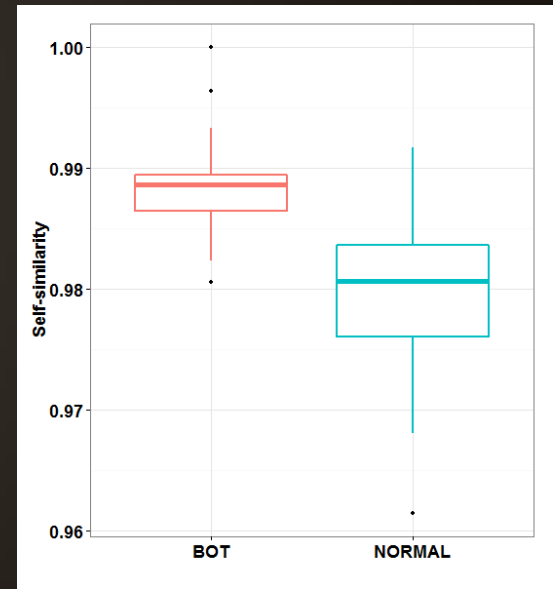
- Almost bots have higher values than human users



Lineage



Aion



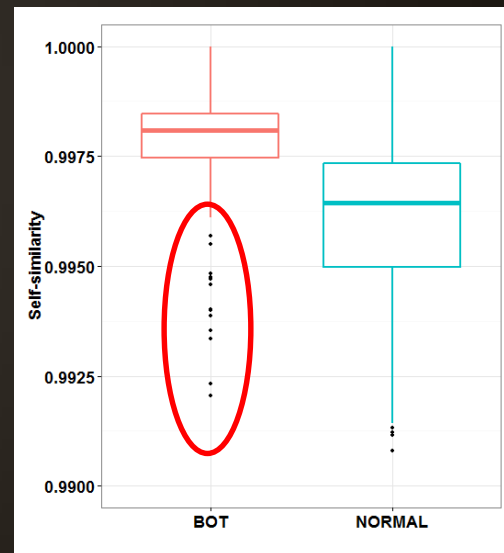
B&S

# Feature selection

## Additional feature selection

- Exceptional cases – short time playing or no activities over long time
- Outliers

No.	Field name	Description
1	self_sim	Self-similarity
2	cosim_count	Count of a set of log vectors
3	cosim_uniq_count	Unique count of a set of log vectors
4	cosim_zero_count	Count of data in which cosine similarity is zero
5	cosim_mode	Count of data that appears most often in a set of log vectors
6	total_log_count	Total count of logs generated by user
7	main_char_level	Character level
8	total_use_time_min	Play time during certain period per user
9	npc_kill_count	NPC kill count
10	trade_get_count	Count of trade in which user takes item
11	trade_give_count	Count of trade in which user gives items
12	retrieve_count	Count of activity in which user retrieve items from warehouse
13	deposit_count	Count of activity in which user deposits items to warehouse
14	log_count_per_min	Average count of logs are generated per minute

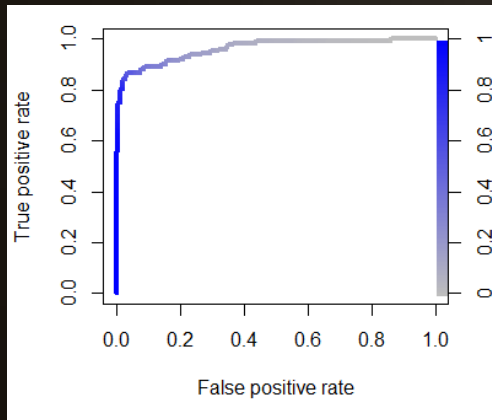


# Experiments

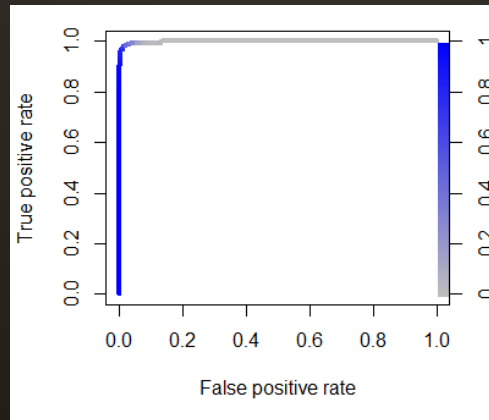
## Performance evaluation

- Model1: using only self-similarity. Model2: using all features

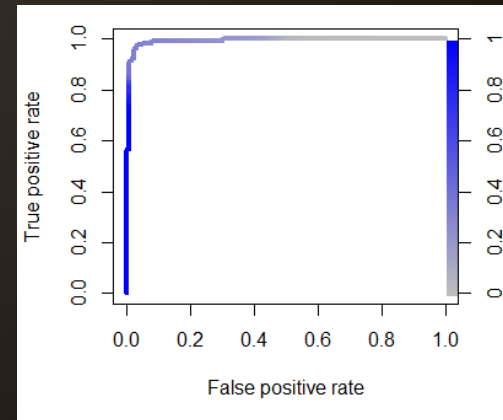
Game	BOT	Human	AUC (model 1)	AUC (model 2)
Lineage	128	149	0.8967	<b>0.9455</b>
Aion	186	160	0.9557	<b>0.9942</b>
B&S	131	129	0.8280	<b>0.9399</b>



Lineage



Aion



B&S

# Model Maintenance

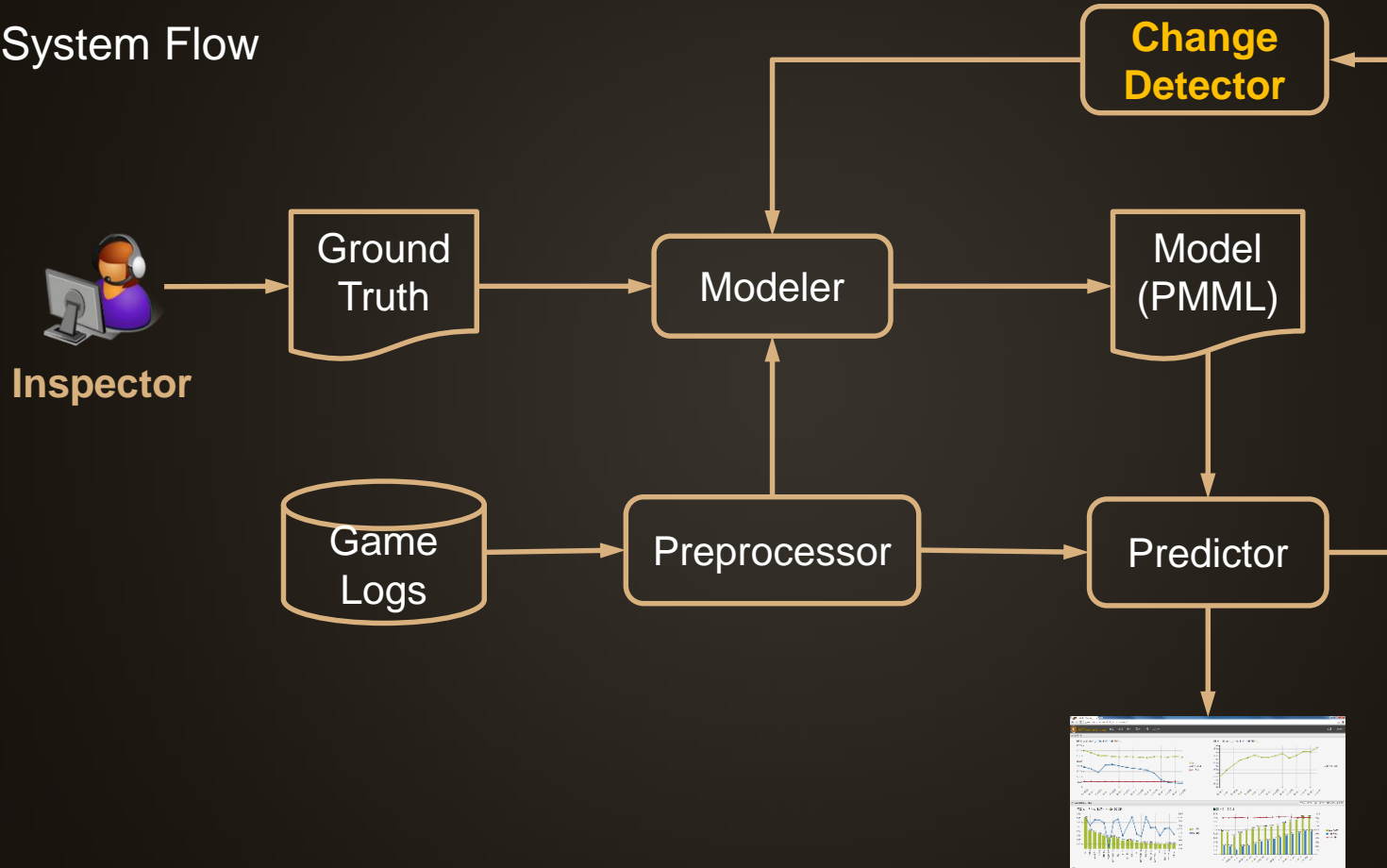
# Model maintenance

## Motivation and consideration

- How to optimize the time for retraining
  - Too often -> high cost
  - Too rare -> obsolete model
- How to retrain a model autonomously

# Model maintenance

System Flow

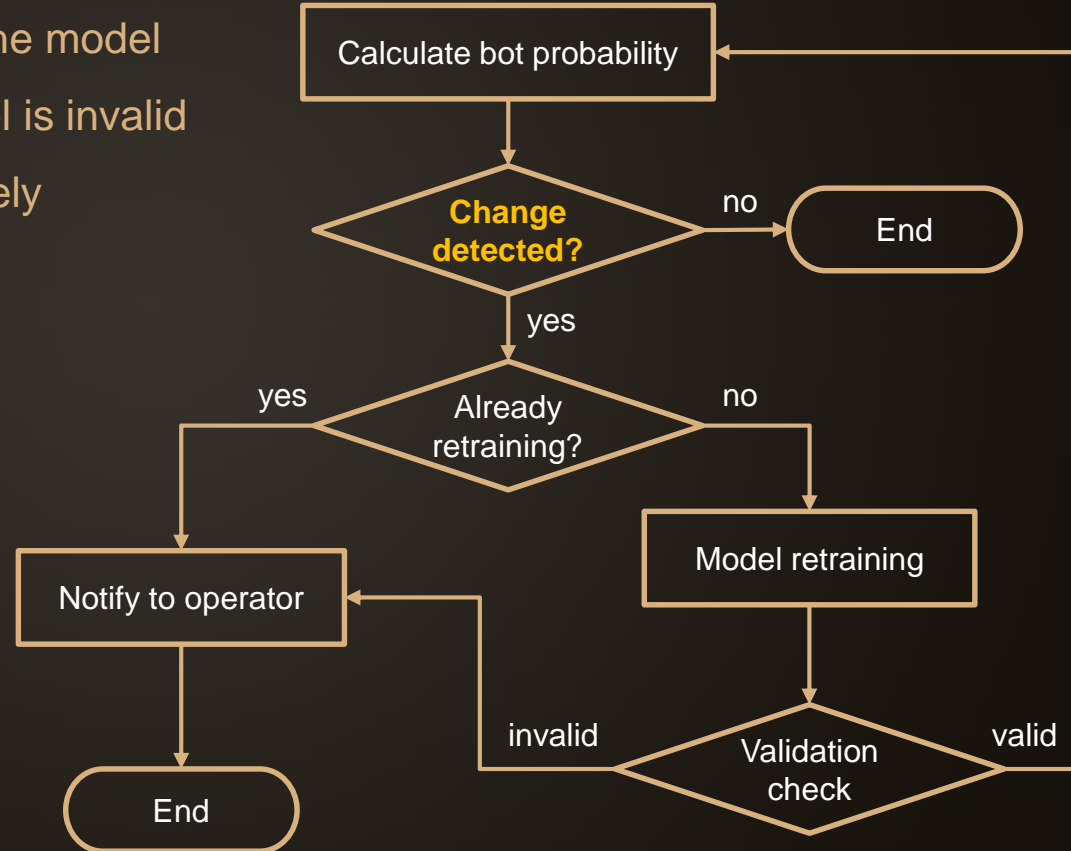


**BOT Detection System**

# Model maintenance

## Logic Flow

- If change is detected, retraining the model
- Notifying to operator, if new model is invalid or change is detected consecutively

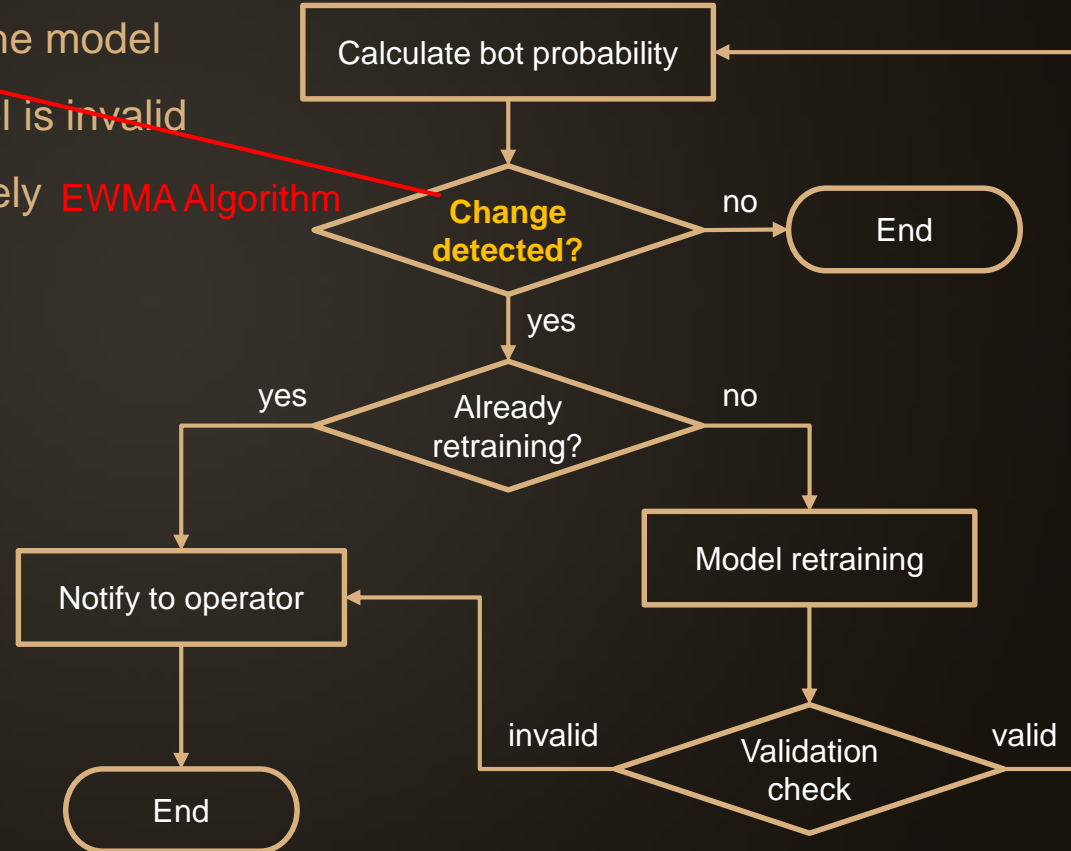




# Model maintenance

## Logic Flow

- If change is detected, retraining the model
- Notifying to operator, if new model is invalid or change is detected consecutively EWMA Algorithm




# Model maintenance

## EWMA algorithm

- Calculating the correlation coefficient of bot probability between time  $t$  and  $t-1$

User	Bot probability (time $t$ )	Bot probability (time $t-1$ )
A	0.99	0.95
B	0.95	0.92
C	0.23	0.25
D	0.55	0.55
...	...	...



Correlation coefficient

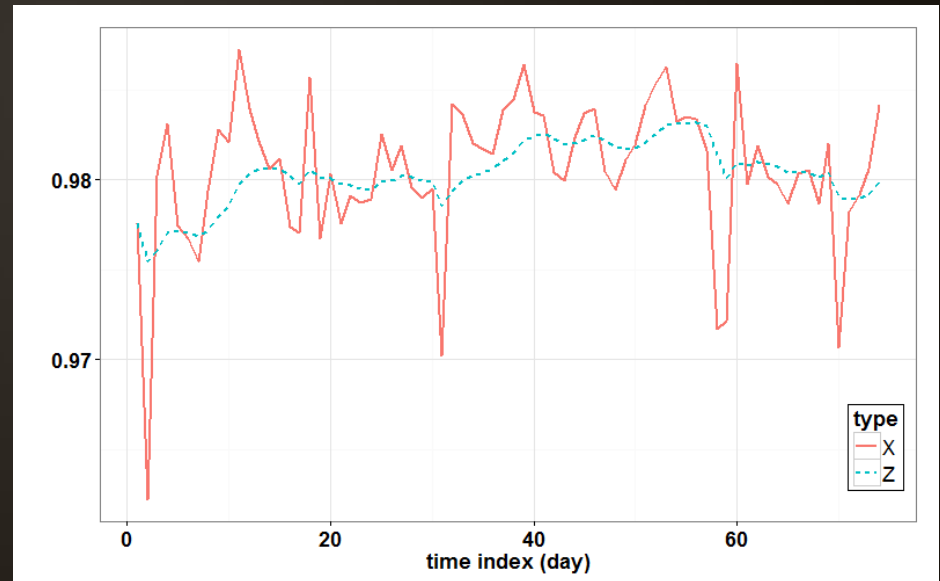
# Model maintenance

## EWMA algorithm

- Calculating the correlation coefficient of bot probability between time  $t$  and  $t-1$



- Calculating the weighted moving average of coefficients



(X: coefficient, Z: moving average)

# Model maintenance

## EWMA algorithm

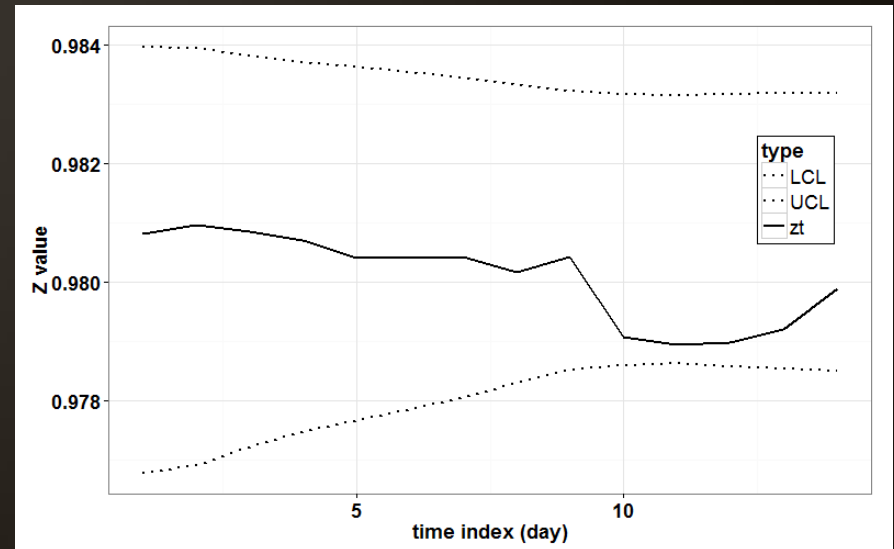
- Calculating the correlation coefficient of bot probability between time  $t$  and  $t-1$



- Calculating the weighted moving average of coefficients



- Measuring upper an lower control limits



# Model maintenance

## EWMA algorithm

- Calculating the correlation coefficient of bot probability between time t and t-1



- Calculating the weighted moving average of coefficients



- Measuring upper an lower control limits



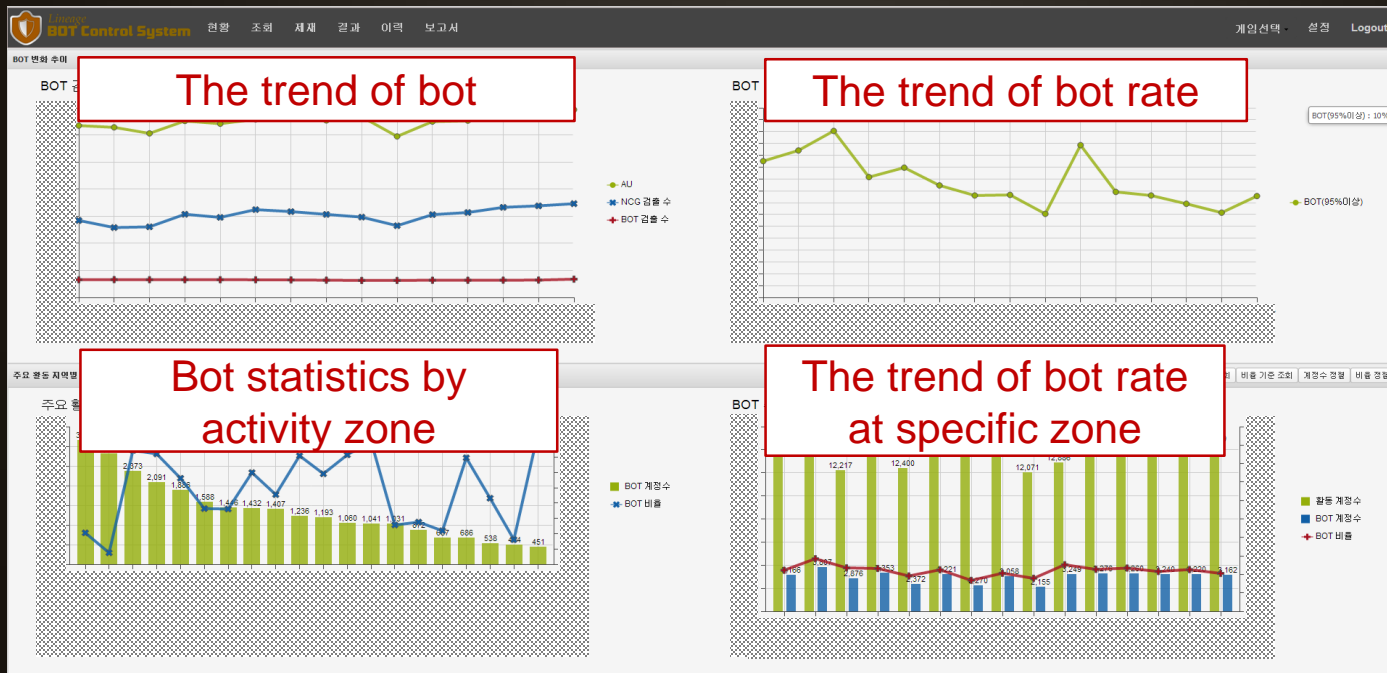
- Retraining the model, unless  $LCL < Z_t < UCL$

# Real-World Deployment

# Real-World Deployment

## BOT detection system – dashboard

- Provide the trend of numbers or rates of BOT, and the chart of BOT statistics by main activity zone



# Real-World Deployment

## BOT detection system – search and filter

- Search and filter the list of accounts to ban

The screenshot displays the 'BOT Control System' interface. On the left, the 'Search Options' panel includes fields for '일자' (Date), 'BOT 확률' (BOT Probability), '자기 유사도' (Self-similarity), 'Log ID (8530)', 'IP 중복 수' (IP Duplication Count), '획득 태어나' (Acquisition Birth), '최빈 캐릭터 레벨' (Least Frequent Character Level), 'NCG 패턴 ID', '주요 활동 지역' (Main Activity Area), '메인 캐릭터 직업' (Main Character Job), '메인 캐릭터 시대' (Main Character Era), '최빈 플레이어 시대' (Least Frequent Player Era), and '고객 등급' (Customer Grade). The 'Search Results' table on the right lists various account details. Two red callout boxes are overlaid on the interface: one on the search options and another on the search results table.

**Fill in the conditions to filter accounts to ban**

**Print the list of accounts to ban according to the search conditions**

일자	계정 ID	계정명	BOT 확률	자기 유사도	메인 캐릭터 서...	메인 캐릭터 이름	메인 캐릭터 레벨	메인 캐릭터 직업	최초 접속일	총 플레이 시간(...)	주요 활동 지역	고객 등급	이용권 종류	한일 결제 금
20150824			0.998586	0.997864	2		77	3	2002-10-28 15:...	1436	게임물	GOLD	90일 무제한	70400
20150824			0.997234	0.998447	23		55	2	2002-04-15 03:...	1436	게임물	FAMILY	7일 15시간 기...	29700
20150824			0.996279	0.998633	45		78	4	2002-03-02 22:...	1432	게임물	SILVER	90일 무제한	0
20150824			0.992925	0.998338	20									
20150824			0.998782	0.998494	4									
20150824			0.996723	0.998605	31									
20150824			0.997956	0.998699	4									
20150824			0.994786	0.99844	42									
20150824			0.998426	0.998785	45									
20150824			0.996618	0.998383	42									
20150824			0.998135	0.998551	4									
20150824			0.99785	0.998628	4									
20150824			0.993649	0.998694	45									
20150824			0.993352	0.998694	45									
20150824			0.998421	0.998854	9		69	3	1999-11-30 23:...	1440	게임물	GOLD	30일 무제한	29700
20150824			0.998523	0.998074	2		79	4	2005-01-11 15:...	1434	게임물	GOLD	90일 무제한	70400
20150824			0.999154	0.998557	4		78	2	2002-06-06 18:...	1436	게임물	SILVER	90일 무제한	0
20150824			0.996389	0.998838	18		75	2	2002-09-26 23:...	1430	게임물	GOLD	90일 무제한	0
20150824			0.998985	0.998789	2		72	3	2003-05-30 19:...	1418	게임물	SILVER	90일 무제한	70400
20150824			0.998757	0.997841	27		77	4	2006-02-16 20:...	1434	게임물	GOLD	90일 무제한	0
20150824			0.999989	0.998541	44		56	2	2006-07-04 22:...	1440	게임물	FAMILY	30일 무제한	109700
20150824			0.991572	0.99761	20		71	7	2006-10-23 00:...	1440	게임물	GOLD	90일 무제한	0
20150824			0.991716	0.998708	38		70	3	2007-07-24 14:...	1440	게임물	GOLD	90일 무제한	0
20150824			0.997635	0.997988	2		80	4	2008-08-03 16:...	1420	게임물	GOLD	90일 무제한	70400
20150824			0.998408	0.998519	4		78	2	2009-03-18 18:...	1430	게임물	SILVER	90일 무제한	0
20150824			0.990803	0.998306	20		56	3	2009-10-08 20:...	1440	게임물	GOLD	90일 무제한	422400
20150824			0.995184	0.997678	39		75	4	2010-03-03 13:...	1440	게임물	SILVER	90일 무제한	211200
20150824			0.996059	0.997842	2		74	4	2010-03-09 03:...	1440	게임물	GOLD	90일 무제한	3000
20150824			0.994547	0.998227	2		74	4	2010-03-14 02:...	1440	게임물	GOLD	90일 무제한	3000
20150824			0.995169	0.99782	39		73	4	2010-03-18 05:...	1440	게임물	SILVER	90일 무제한	3000
20150824			0.992925	0.998752	11		76	3	2010-03-19 11:...	1439	게임물	GOLD	90일 무제한	0
20150824			0.990933	0.998615	11		77	3	2010-03-19 11:...	1438	게임물	GOLD	90일 무제한	0
20150824			0.990814	0.998552	11		77	3	2010-03-19 12:...	1440	게임물	GOLD	90일 무제한	0
20150824			0.990959	0.998542	11		77	3	2010-03-19 12:...	1439	게임물	GOLD	90일 무제한	0
20150824			0.997859	0.997897	32		71	4	2010-03-22 20:...	1440	게임물	SILVER	90일 무제한	143800



# Conclusion

# Conclusion

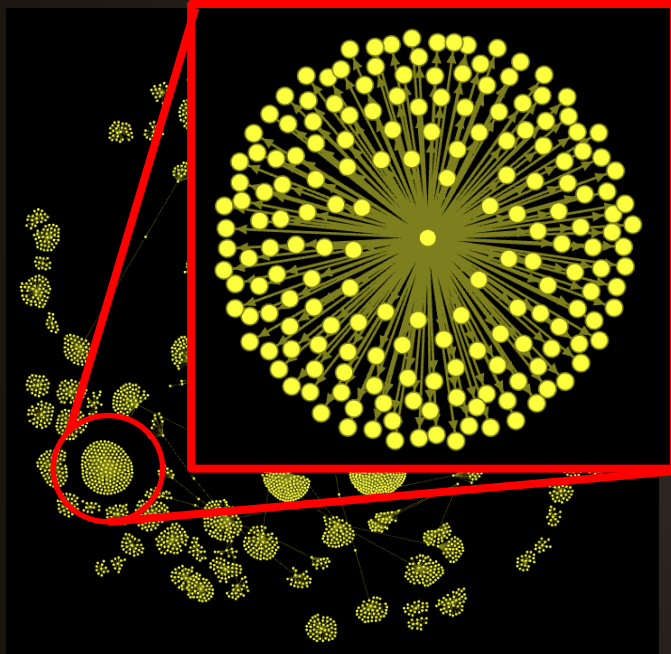
## Contributions

- We proposed self-similarity as a feature and demonstrated its effectiveness **with real datasets**
- We proposed a bot detection framework that includes a detection model **maintenance process**
- We implemented the proposed framework and **utilized it for live MMORPGs**

# Conclusion

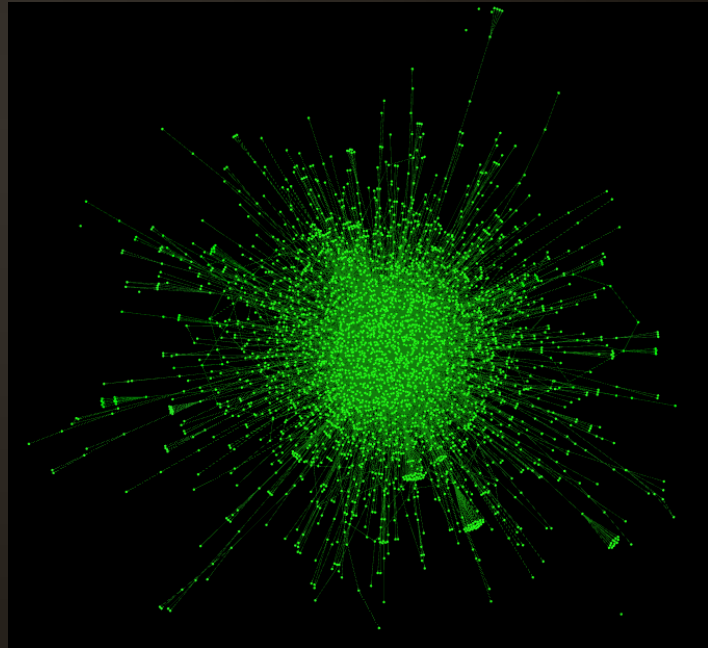
Future works – short-time playing bot

- Undetected massive number of bots playing for less than 10 hours per week
- Star-shaped trading network structure



Short-time playing BOT

VS

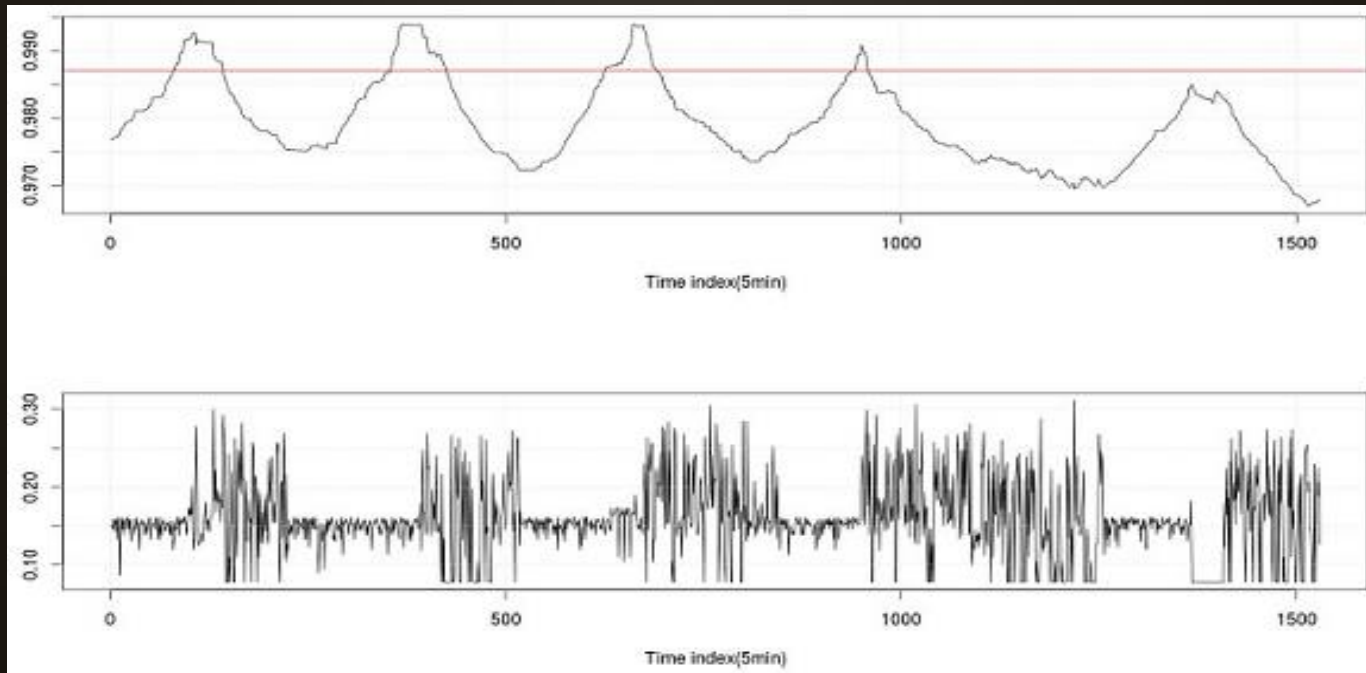


Human users

# Conclusion

Future works – occasional bot users

- Human players playing for hours and then turning on a bot for a few hours
- Self-similarities have pulse, if we use short period of time for aggregation



# Questions and Answers