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

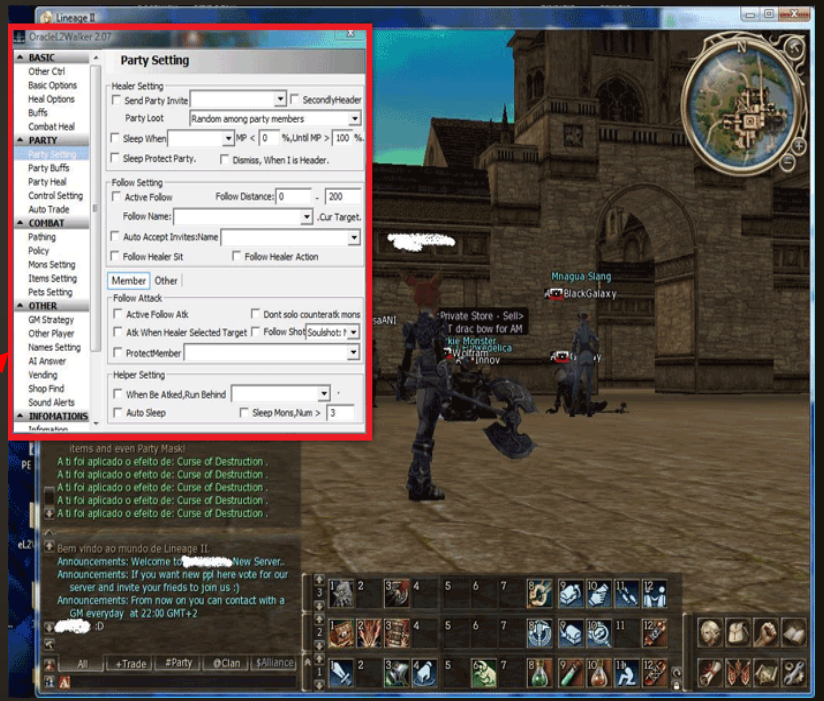
Introduction

Introduction

Game BOT

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

Bot configurations



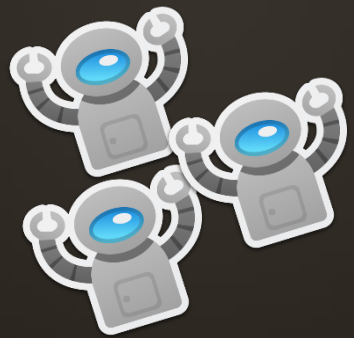
Introduction

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

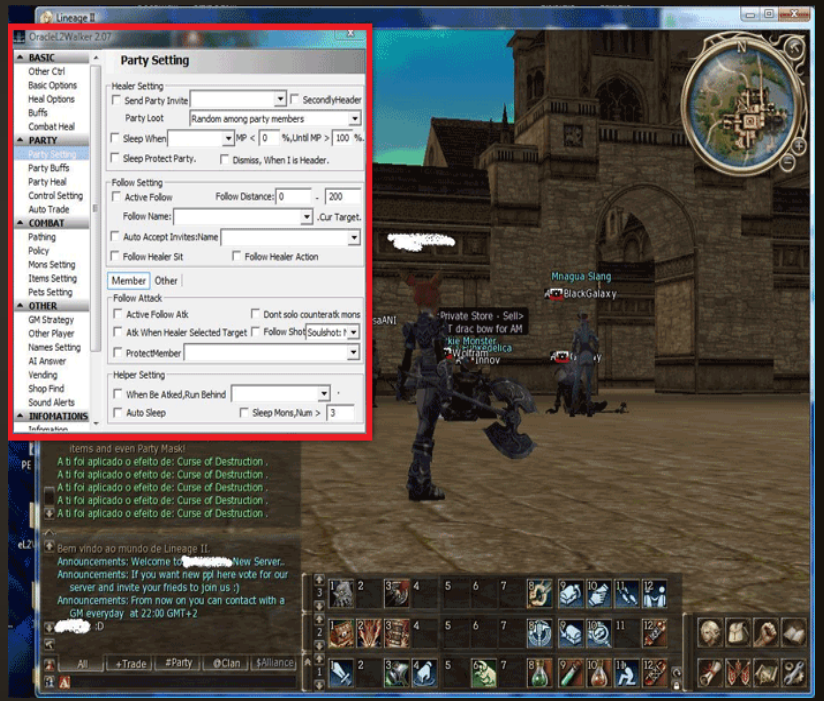
Game BOT

https://www.youtube.com/watch?v=k6tk8_R2w08

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

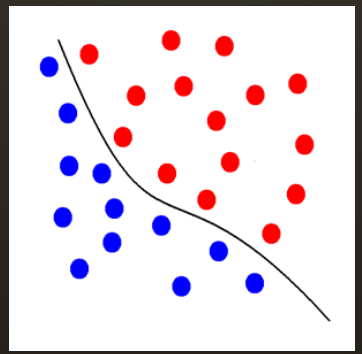
Machine Learning-based Approach

Game Logs

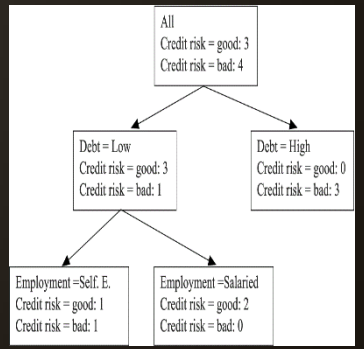
Feature Selection

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



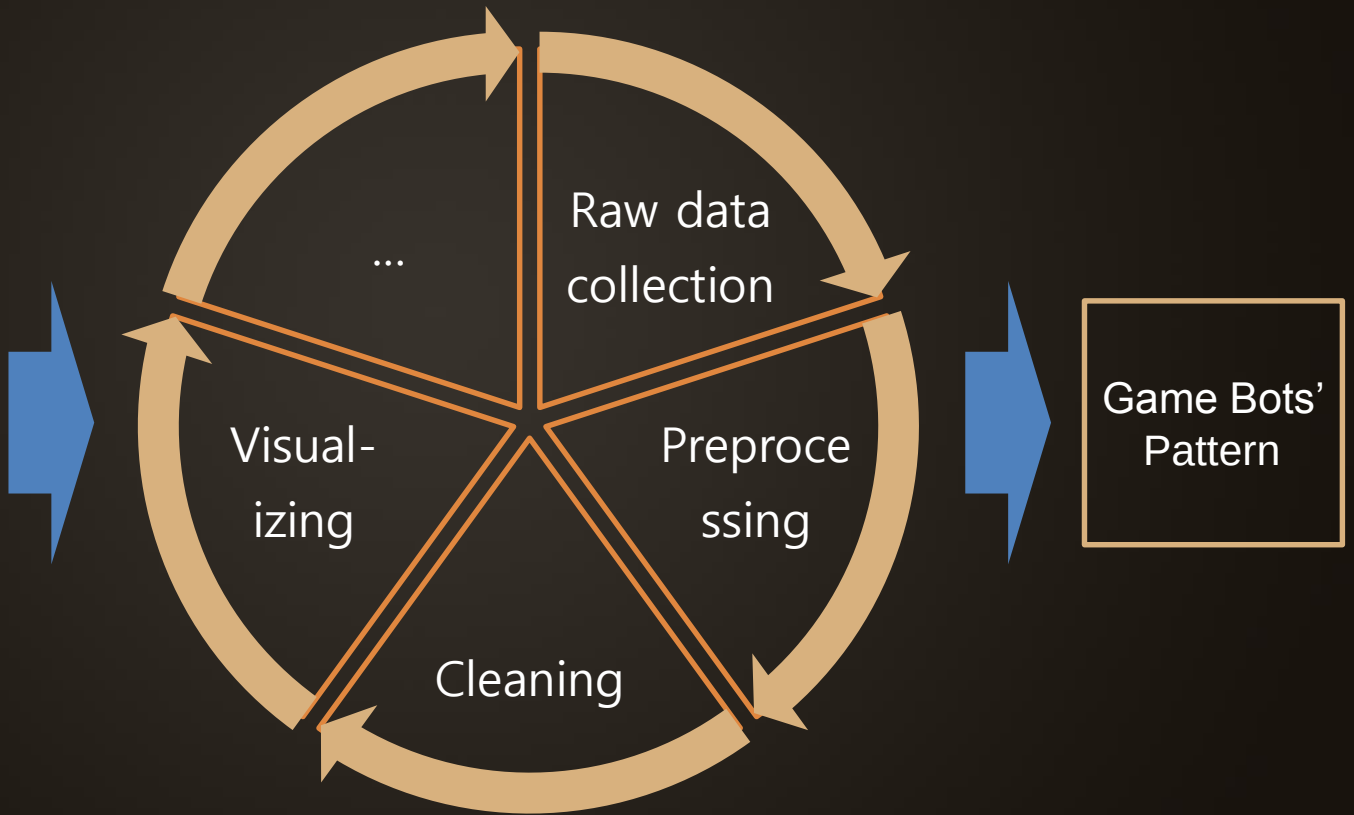
Learning Algorithm



Prediction Model

Introduction

Challenges



Introduction

Challenges

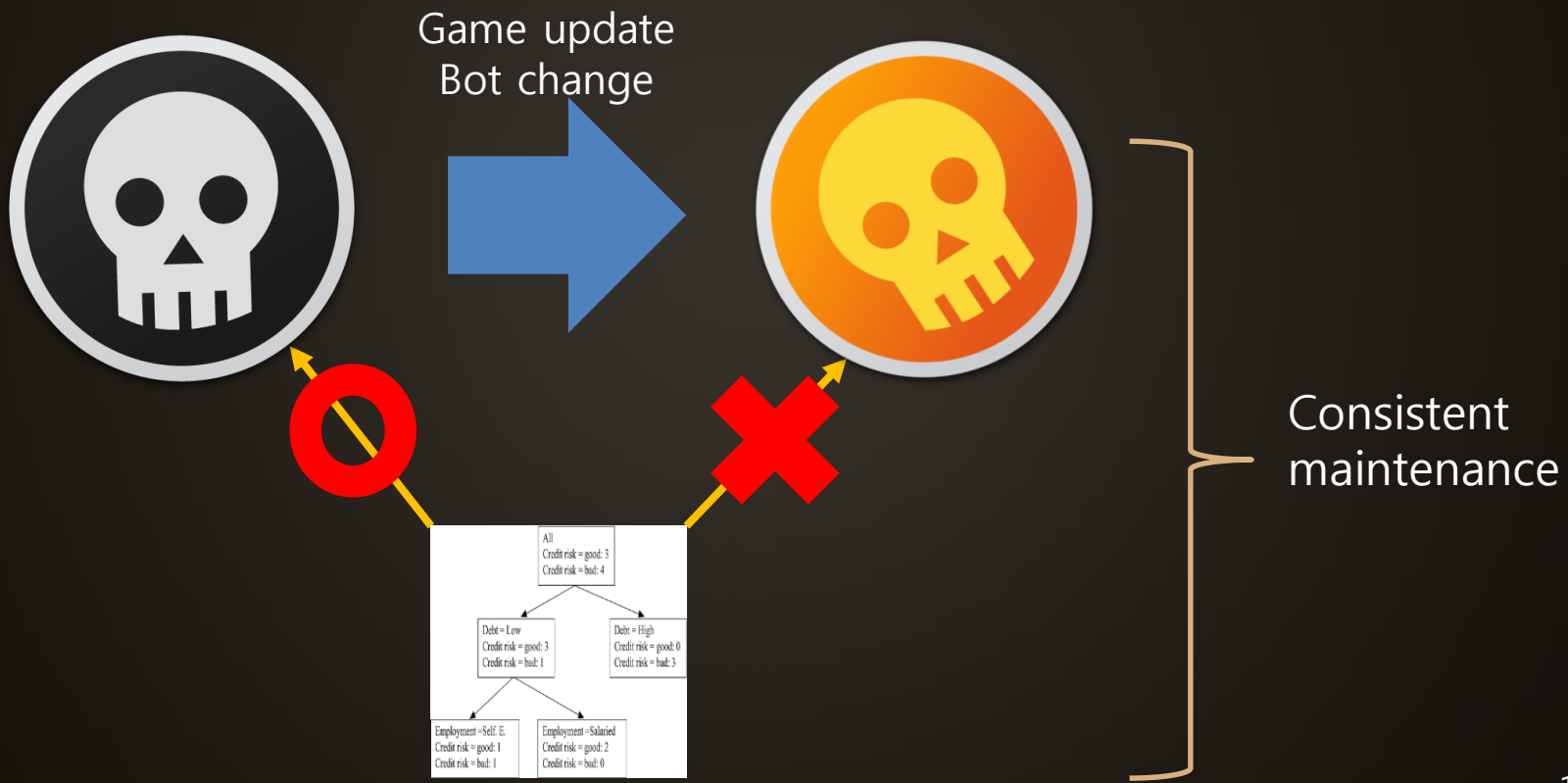
Game A Bots' Pattern

Game B Bots' Pattern

High cost
time consuming

Introduction

Challenges



Introduction

Our proposals

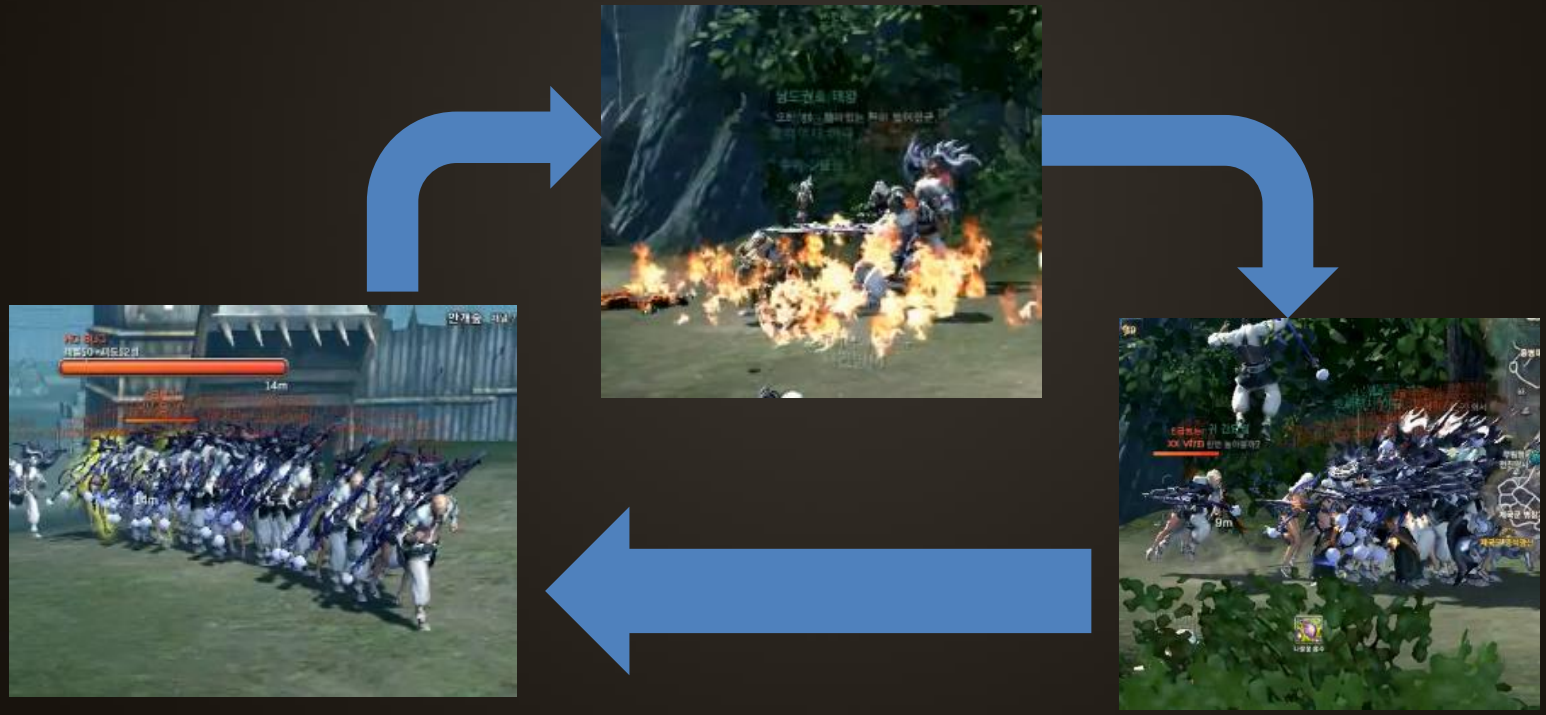
- Using self-similarity as a generic feature
  - Focus on the repetitive activities of game bots, not specific behavior
- Proposing framework to maintain a prediction model autonomously
  - Detect the change in performance of the prediction model and retrain it

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

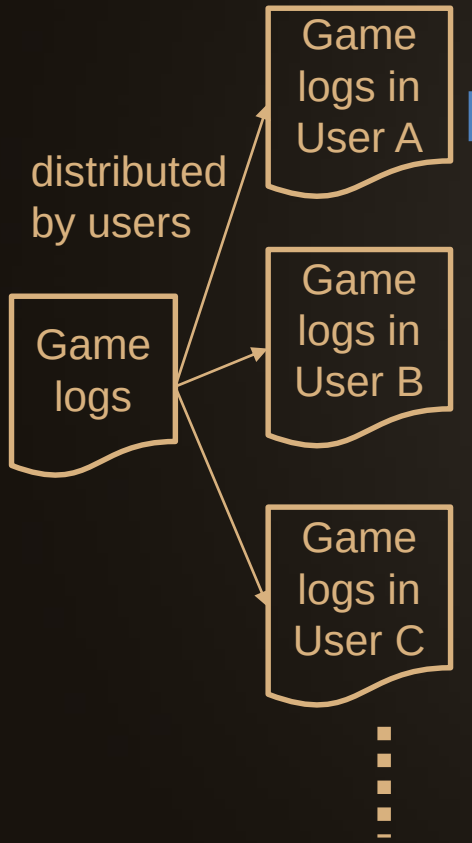
Self-similarity

Detailed process

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

Self-similarity

Generating log vectors

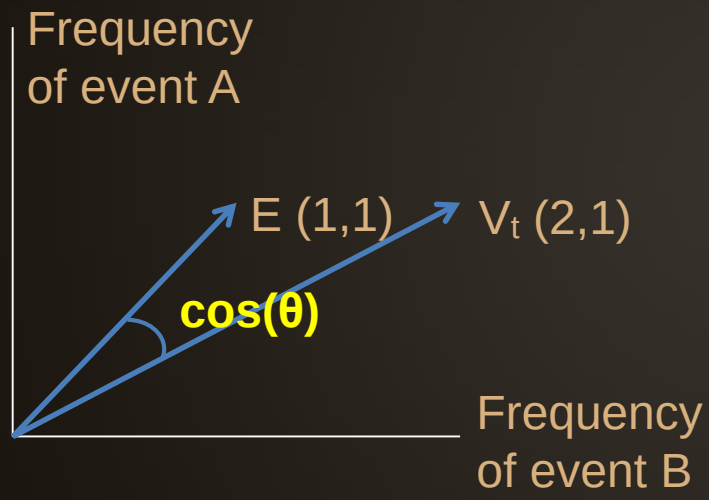


time	Event id	Character info.	Other info.
15/08/13 12:00:12.131	1205	AAA, 34	N/A
15/08/13 12:00:14.237	1204	AAA, 34	N/A
15/08/13 12:00:59.436	1208	AAA, 34	Ogre
15/08/13 12:00:59:436	1208	AAA, 34	Ogre
15/08/13 12:00:59.857	1208	AAA, 34	Troll
15/08/13 12:01:17.019	1022	AAA, 34	Ring
15/08/13 12:01:21.341	1022	AAA, 34	Sword
15/08/13 12:01:23.151	1205	AAA, 34	N/A
15/08/13 12:01:54.354	1204	AAA, 34	N/A
15/08/13 12:01:56.445	1208	AAA, 34	Wolf
15/08/13 12:02:07.351	1205	AAA, 34	N/A
15/08/13 12:02:41.847	1204	AAA, 34	N/A
15/08/13 12:02:47.650	1208	AAA, 34	Ogre
15/08/13 12:03:09.353	1208	AAA, 34	Ogre

Time period (hour:min)	Log count per event id			
	1022	1204	1205	1208
12:00	0	1	1	3
12:01	2	1	1	1
12:02	0	1	1	1
12:03	0	0	0	1

Self-similarity

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



$$\cos(\theta) = \frac{A \cdot B}{||A||||B||} = \frac{\sum A_i \times B_i}{\sqrt{\sum (A_i)^2} \times \sqrt{\sum (B_i)^2}}$$

$$= \frac{(2\times1 + 1\times1)}{\sqrt{2\times2+1\times1} \times \sqrt{1\times1+1\times1}}$$

$$= \frac{3}{\sqrt{5} \times \sqrt{2}}$$

$$\fallingdotseq 0.948$$

Self-similarity

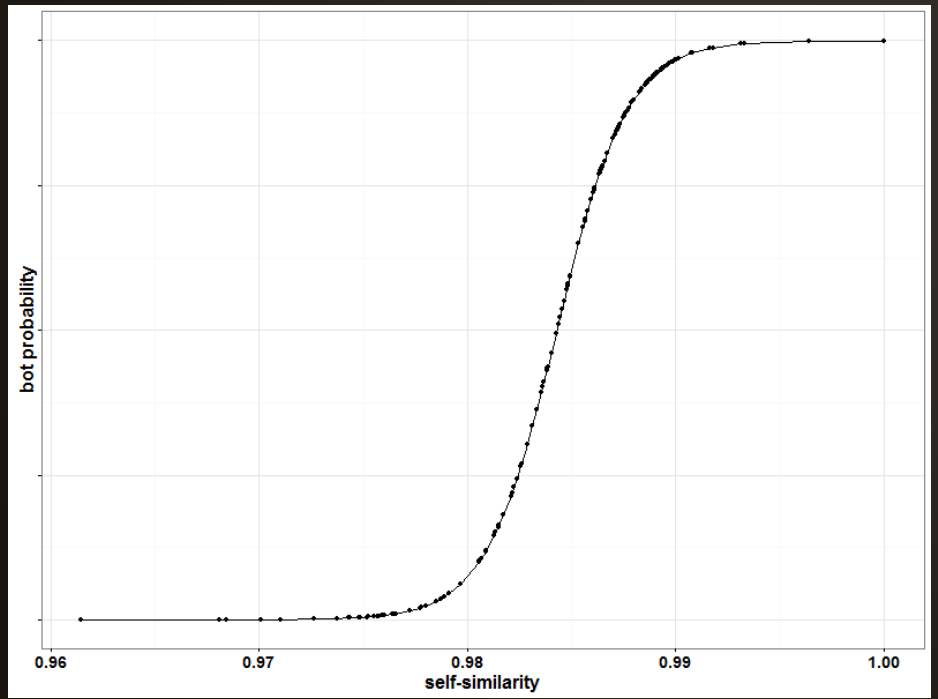
Measuring self-similarity

- 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

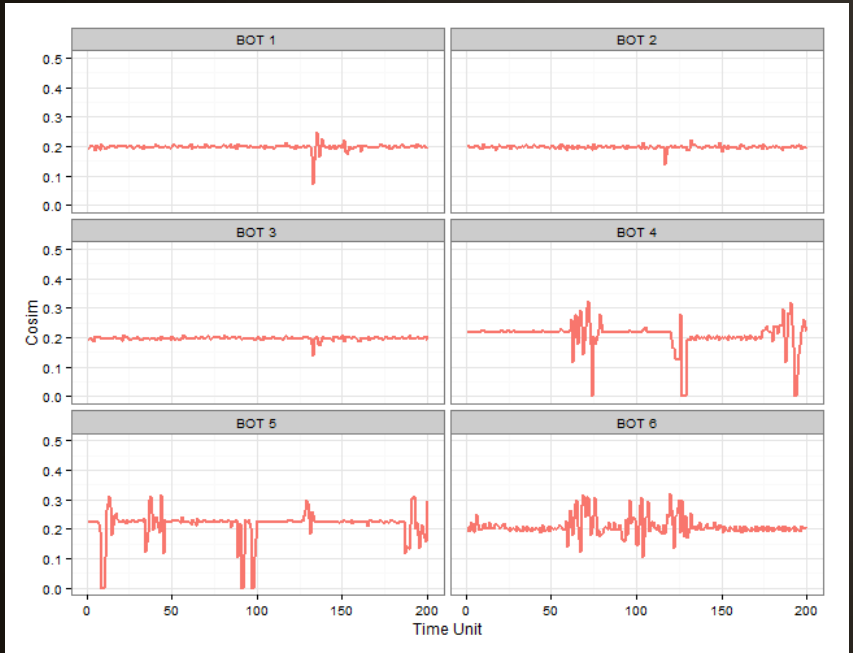
Datasets

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

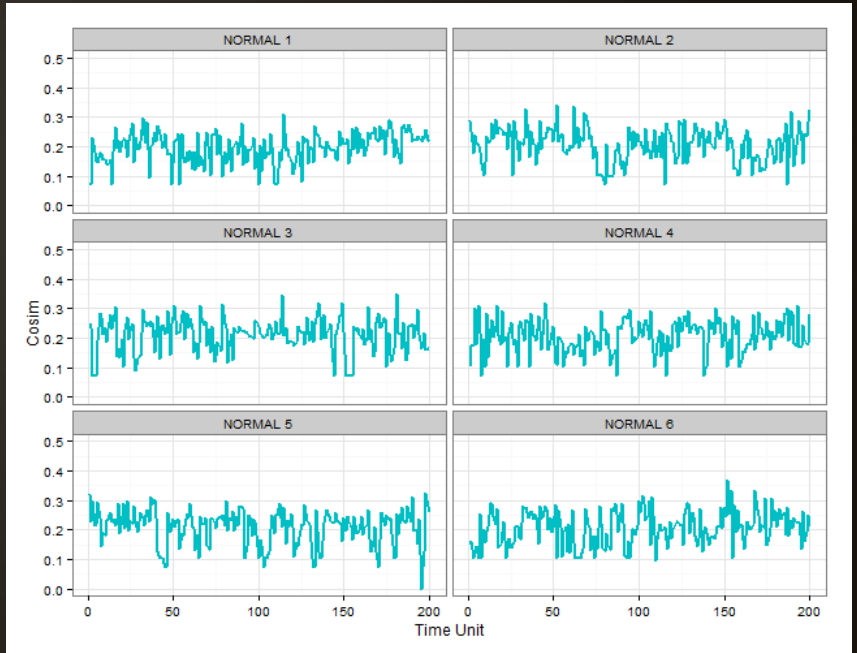
Experiments

Cosine similarities

- Bots have cosine similarities with fewer variations than human users



Bots

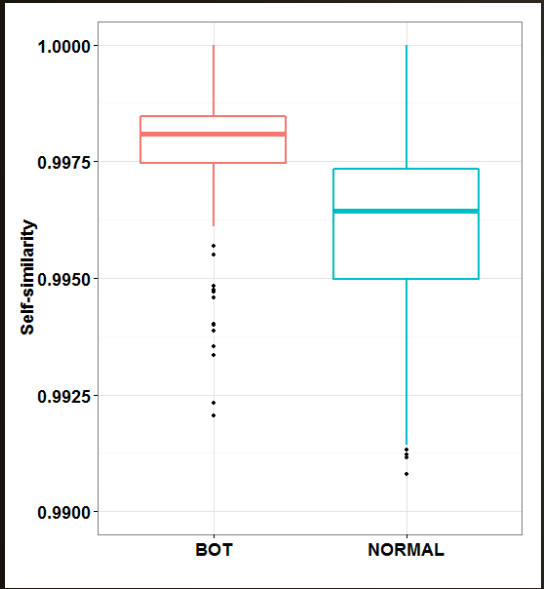


Humans

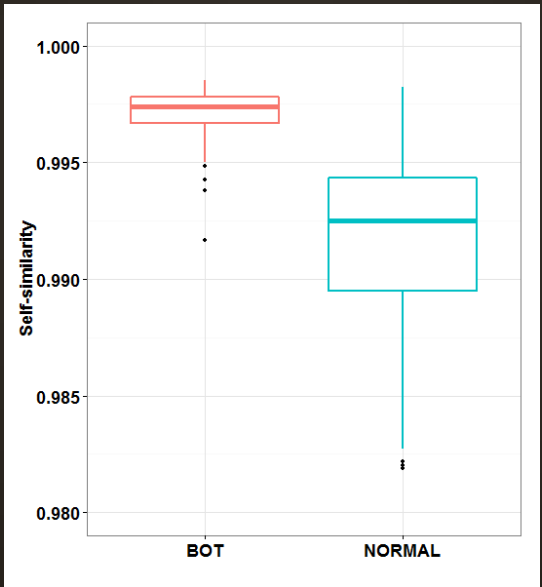
Experiments

Self-similarity

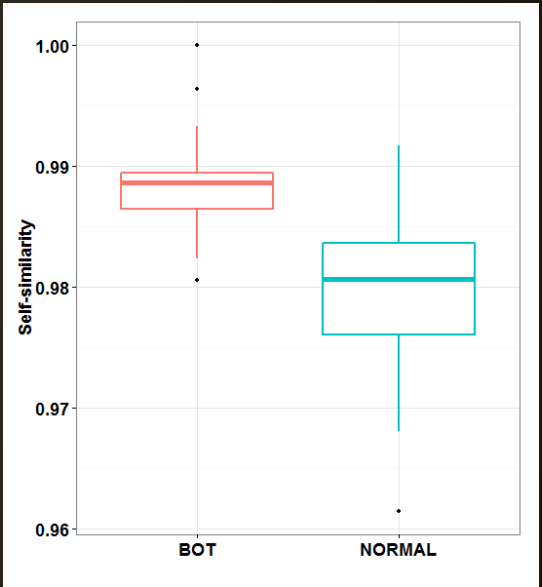
- Almost bots have higher values than human users



Lineage



Aion



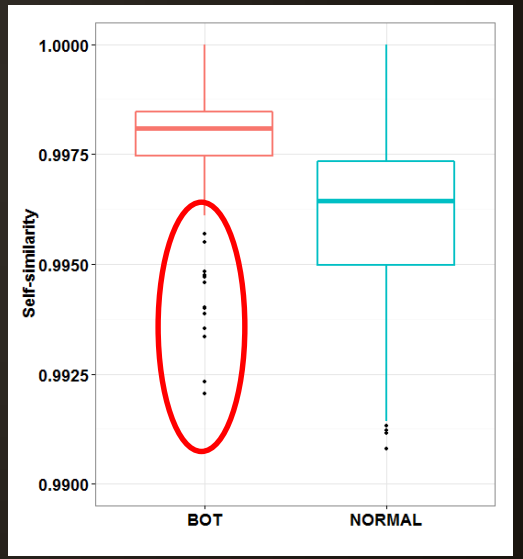
B&S

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

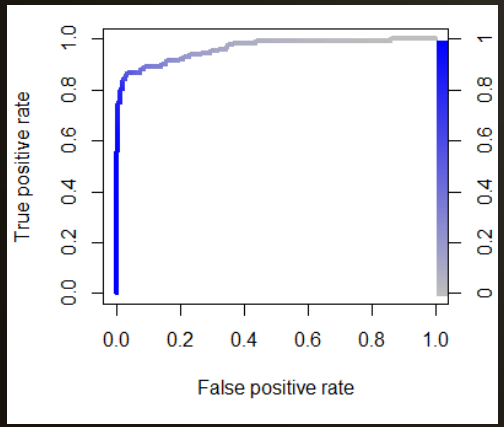


Experiments

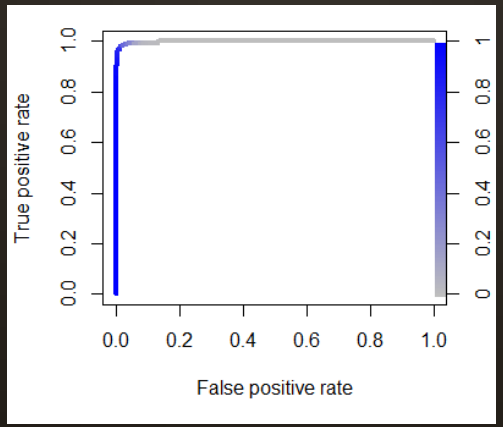
Performance evaluation

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

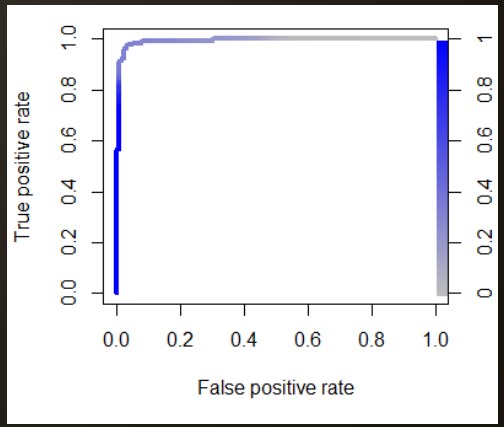
Game	BOT	Human	AUC (model 1)	AUC (model 2)
Lineage	128	149	0.8967	**0.9455**
Aion	186	160	0.9557	**0.9942**
B&S	131	129	0.8280	**0.9399**



Lineage



Aion



B&S

Model Maintenance

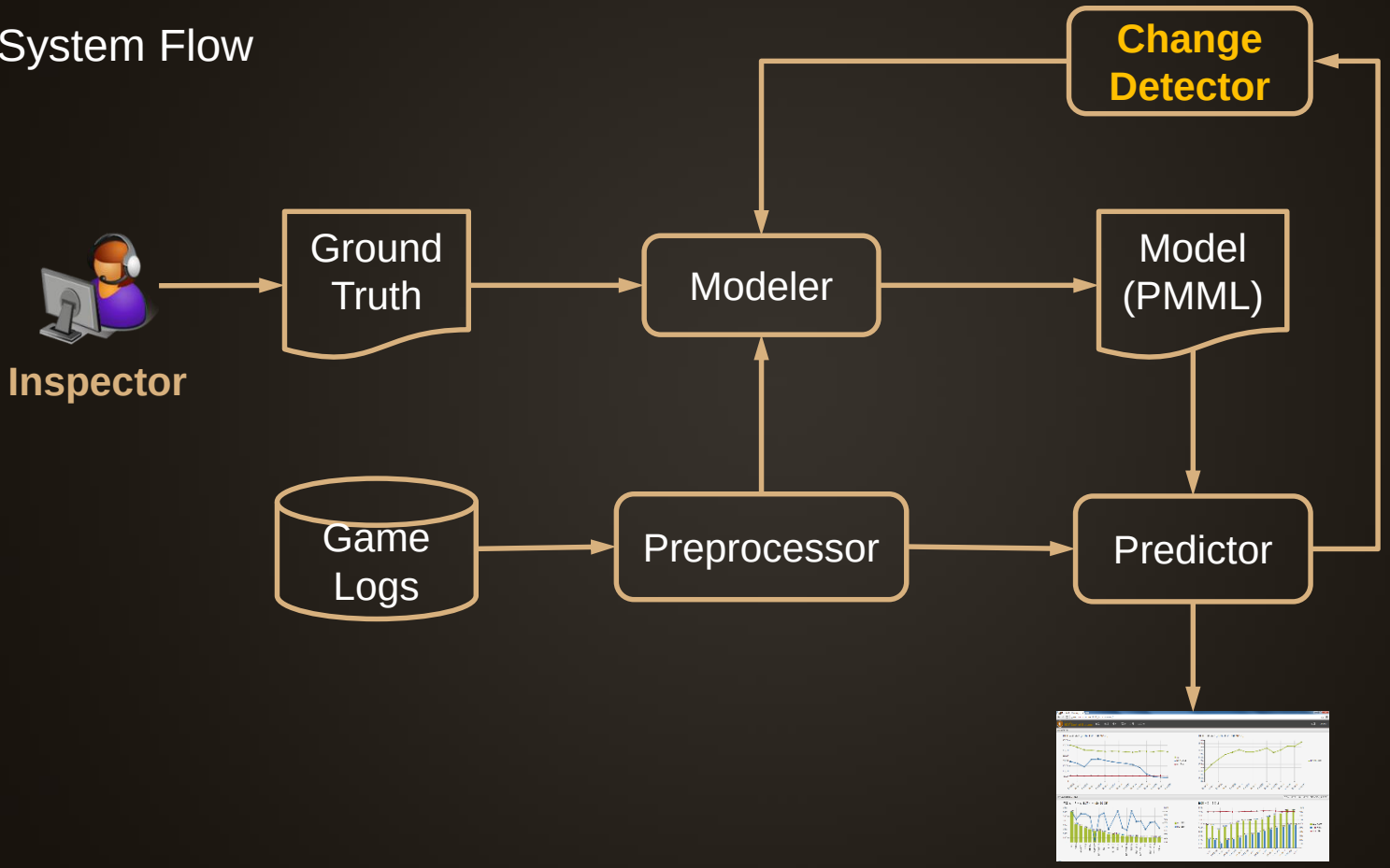
Model maintenance

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

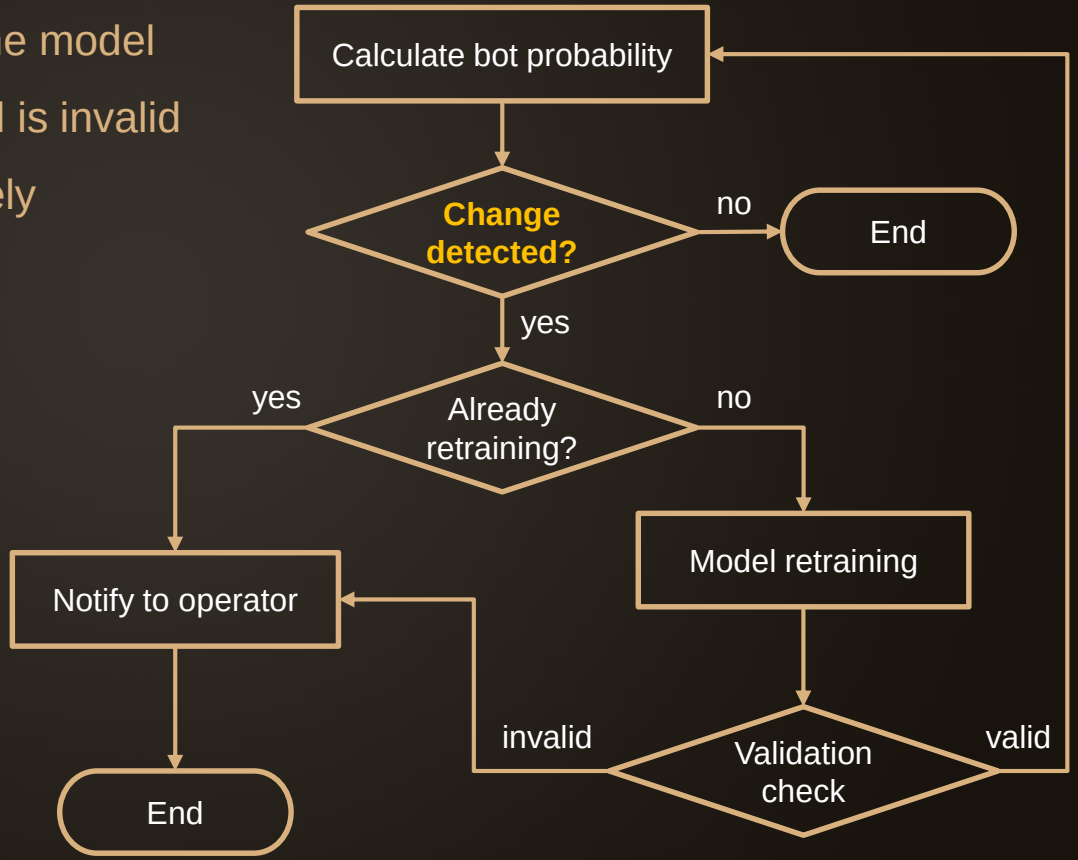
System Flow



Model maintenance

Logic Flow

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

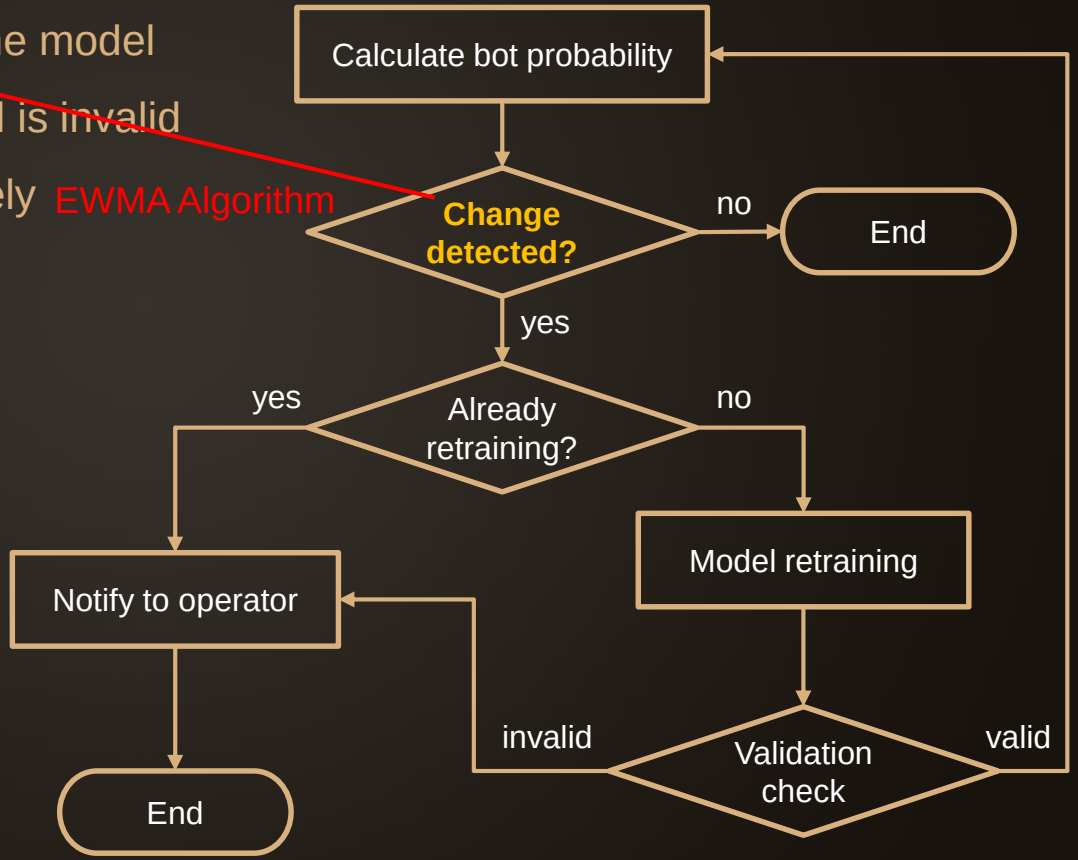


Model maintenance

Logic Flow

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

EWMA Algorithm

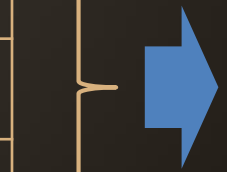


Model maintenance

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
…	…	…



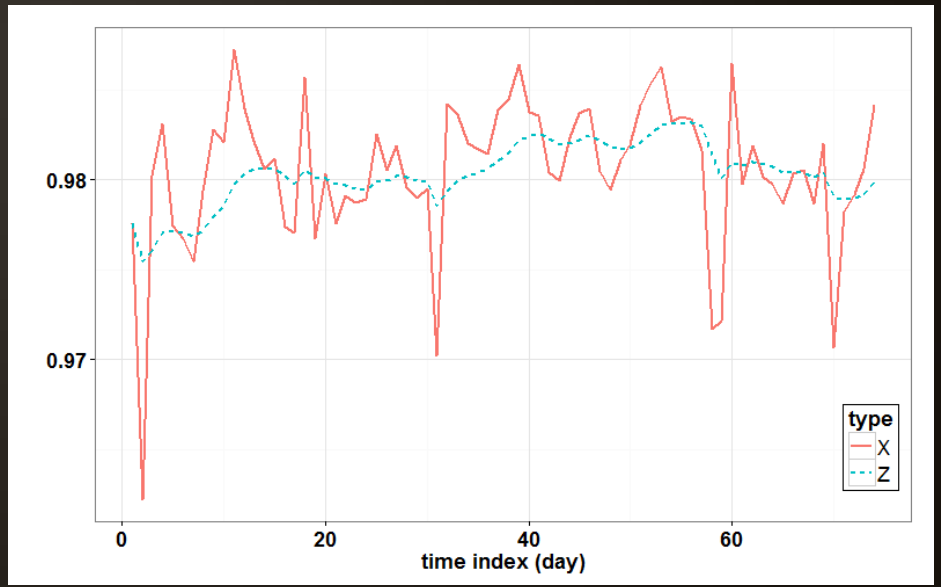
Correlation coefficient

Model maintenance

EWMA algorithm

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

- Calculating the weighted moving average of coefficients

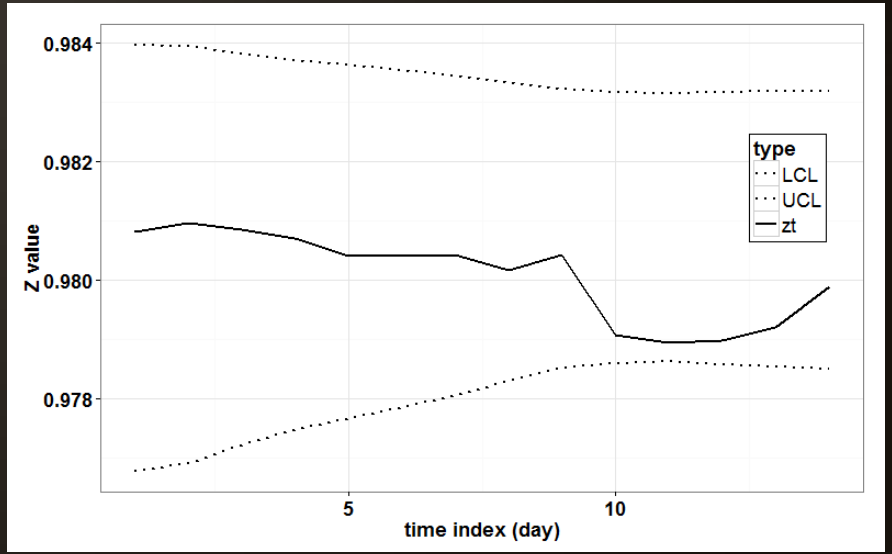


(X: coefficient, Z: moving average)

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

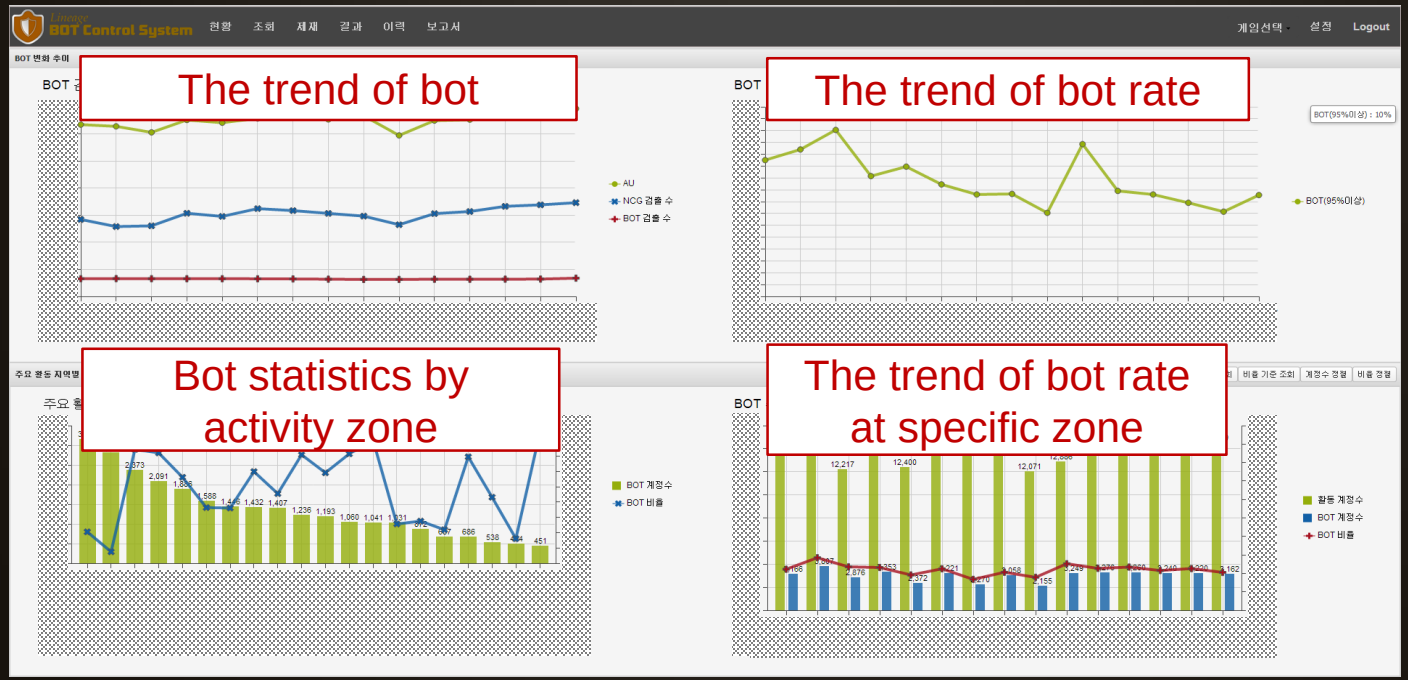
- Retraining the model, unless $LCL < Z_t < UCL$

Real-World Deployment

Real-World Deployment

BOT detection system – dashboard

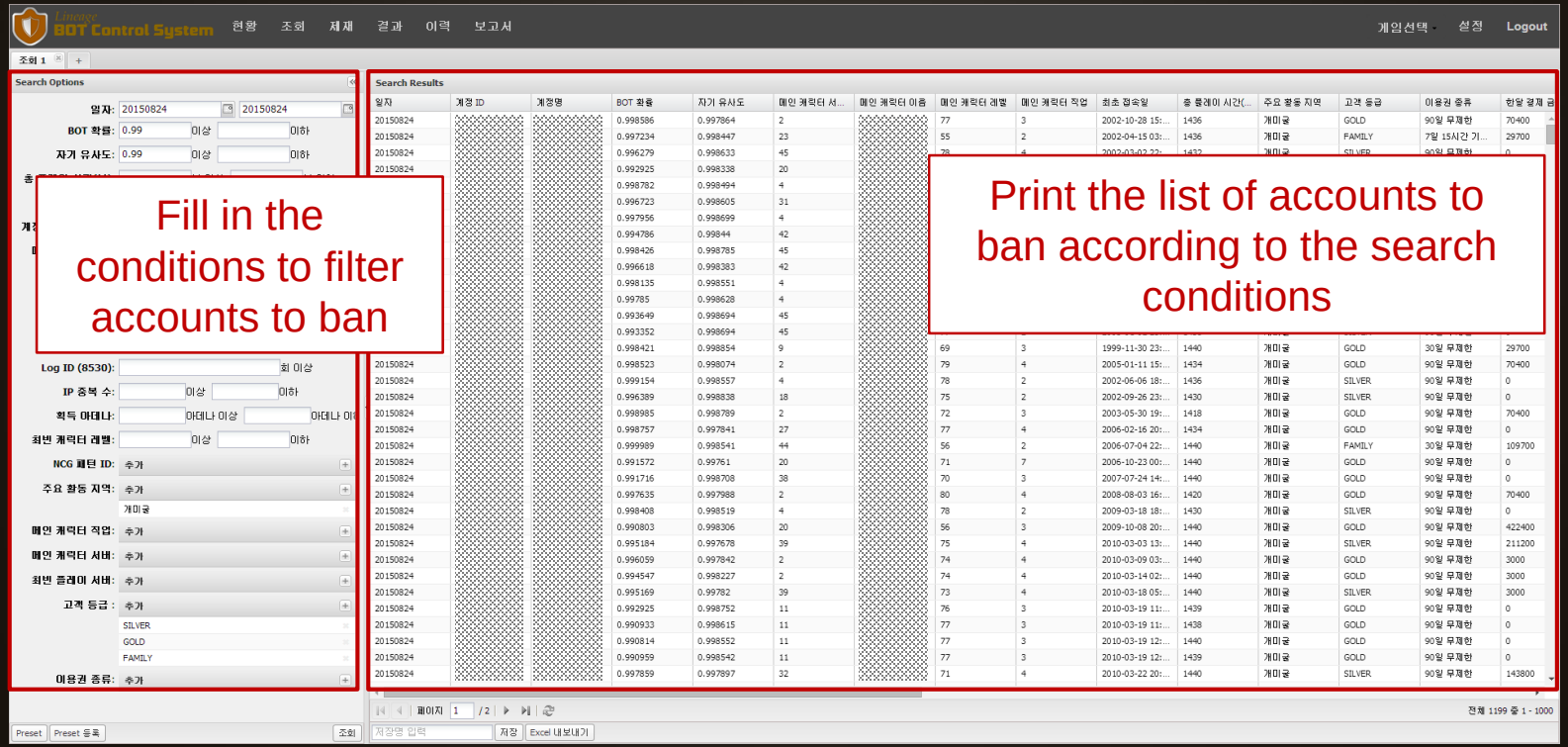
- Provide the trend of numbers or rates of BOT, and the chart of BOT statistics by main activity zone



Real-World Deployment

BOT detection system – search and filter

- Search and filter the list of accounts to ban



Conclusion

Conclusion

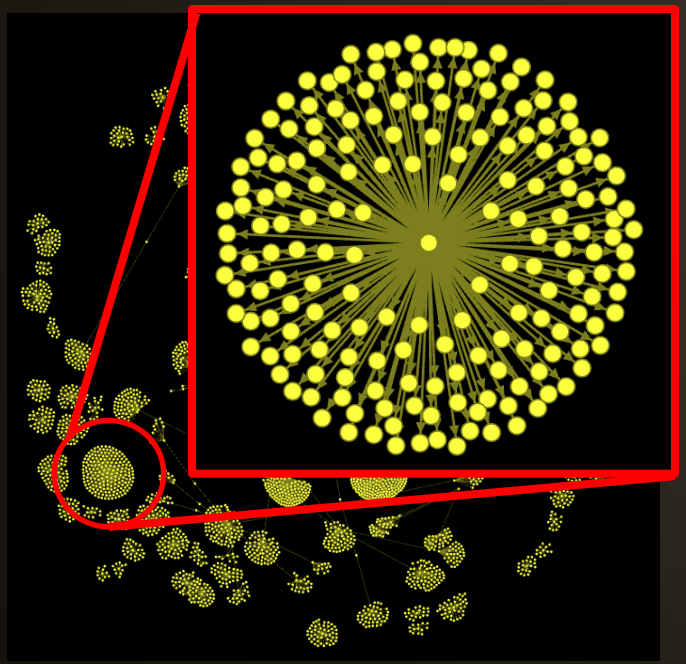
Contributions

- 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

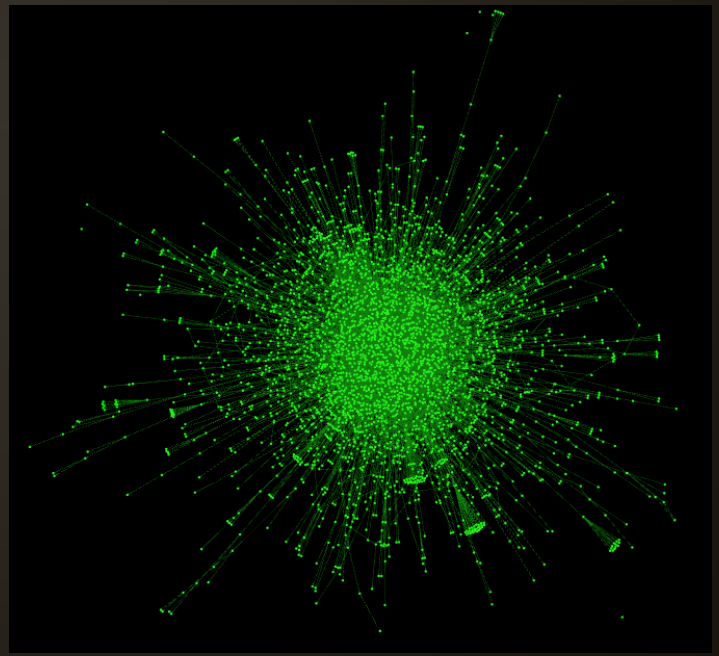
Future works – short-time playing bot

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



Short-time playing BOT

VS

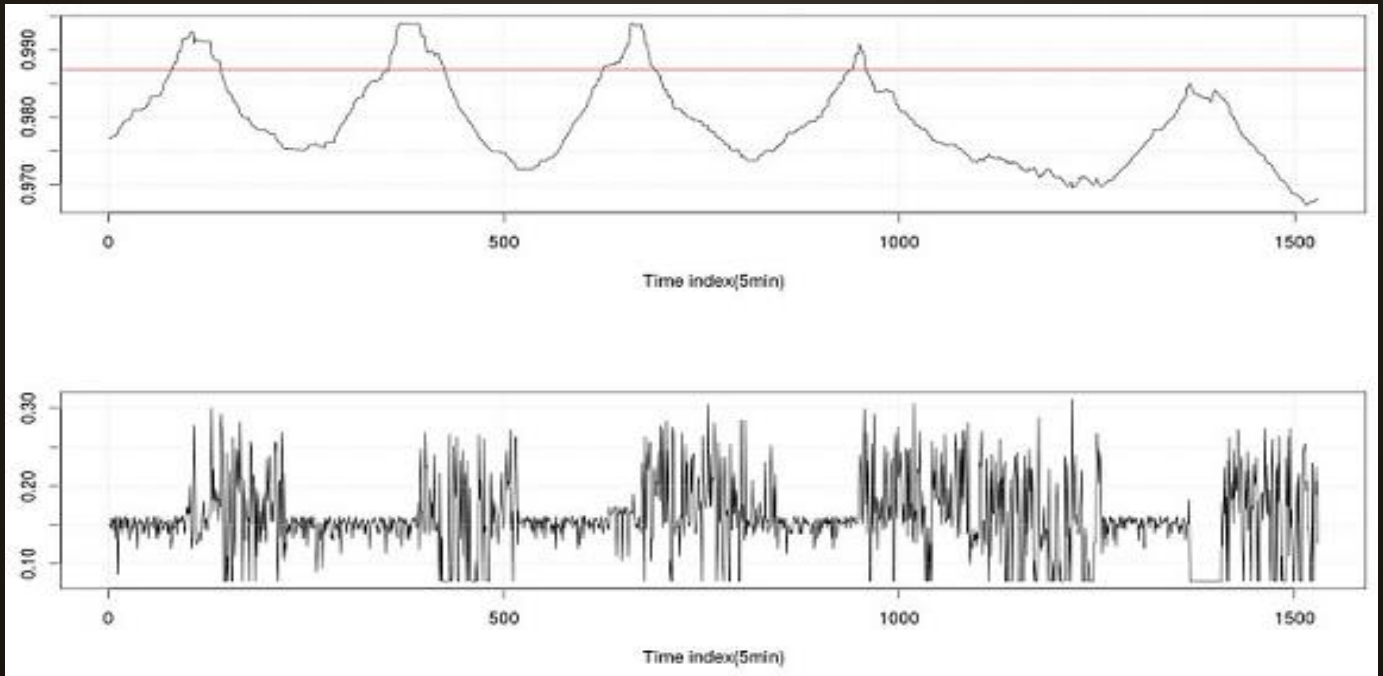


Human users

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