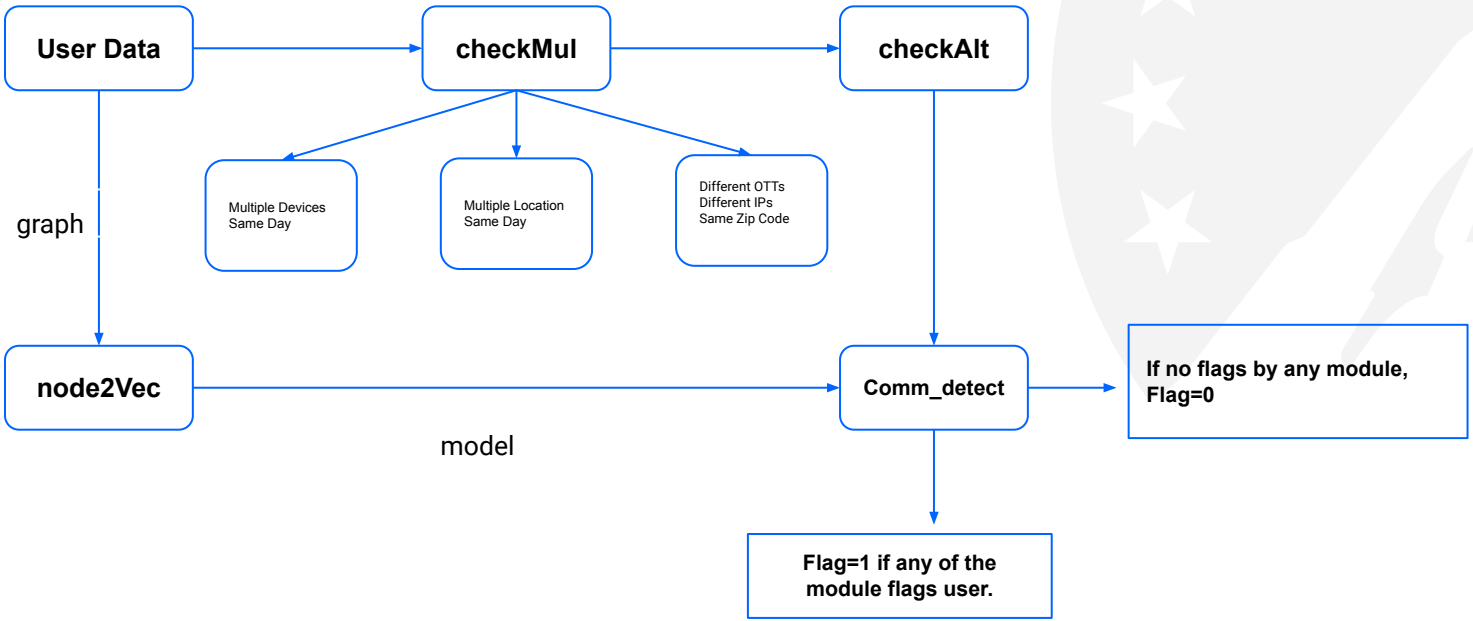
Subscription Abuse Detection System Enhancements

Zirui Zhou
University of Southern California
Summer 2022 Data Engineering Intern

Paramount+

Recap: Subscription Abuse Detection System

Password Sharing outside households to watch shows and live tv, etc.



Problems:

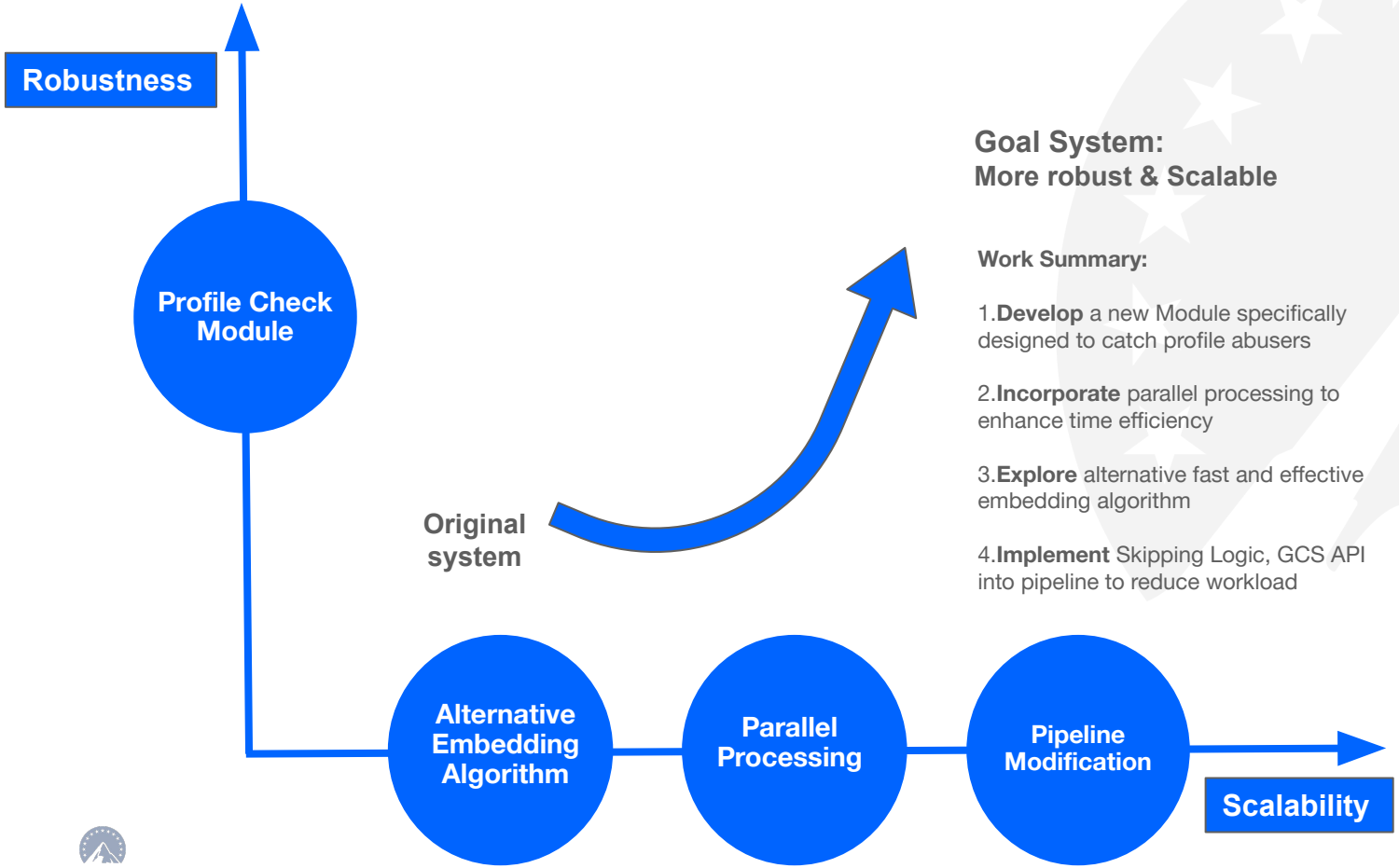
1. Potential space to **improve robustness** by further utilization of data resources - - - Profile data

Row	v69_registration_id_nbr	v126_profile_id	master_profile_ind	profile_type_cd	deleted_ind
1	45895804	12740500	true	ADULT	false
2	45895804	12740500	true	ADULT	false
3	45895804	12740500	true	ADULT	false
4	45895804	12740500	true	ADULT	false

2. **Scalability**

Paramount+

Solutions



Goal System:
More robust & Scalable

Work Summary:

1.**Develop** a new Module specifically designed to catch profile abusers

2.**Incorporate** parallel processing to enhance time efficiency

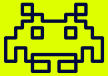
3.**Explore** alternative fast and effective embedding algorithm

4.**Implement** Skipping Logic, GCS API into pipeline to reduce workload

Profile Check Module

Number of Profiles Under the same user_id.

Number of distinct profile's ip far away from its master profiles' location



Number of distinct kids profiles' ip far away from its master adult profile's location.

Occurrence of distant, same-content and concurrent streaming for different profiles

Paramount

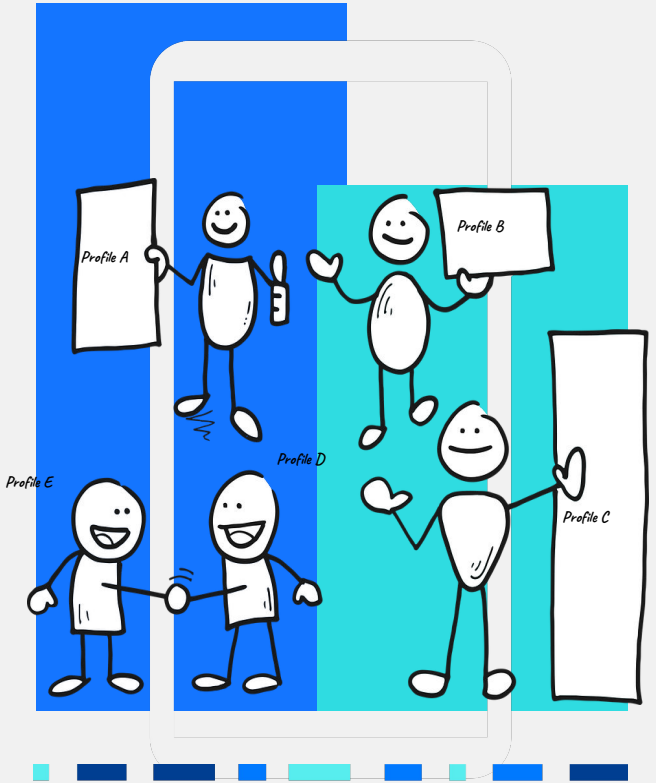
Scenario 1

Number of Profiles

Under the same user account, if there are too much profiles, we flag

Threshold Analysis: p_cnt=5 (99 percentile)

```
pertentile:0, :1.0
pertentile:25, :1.0
pertentile:50, :1.0
pertentile:75, :2.0
pertentile:85, :2.0
pertentile:90, :3.0
pertentile:95, :3.0
pertentile:96, :4.0
pertentile:97, :4.0
pertentile:97.5, :4.0
pertentile:98, :4.0
pertentile:98.5, :4.0
pertentile:99, :5.0
pertentile:99.5, :5.0
pertentile:100, :42.0
```



Paramount

Scenario 2

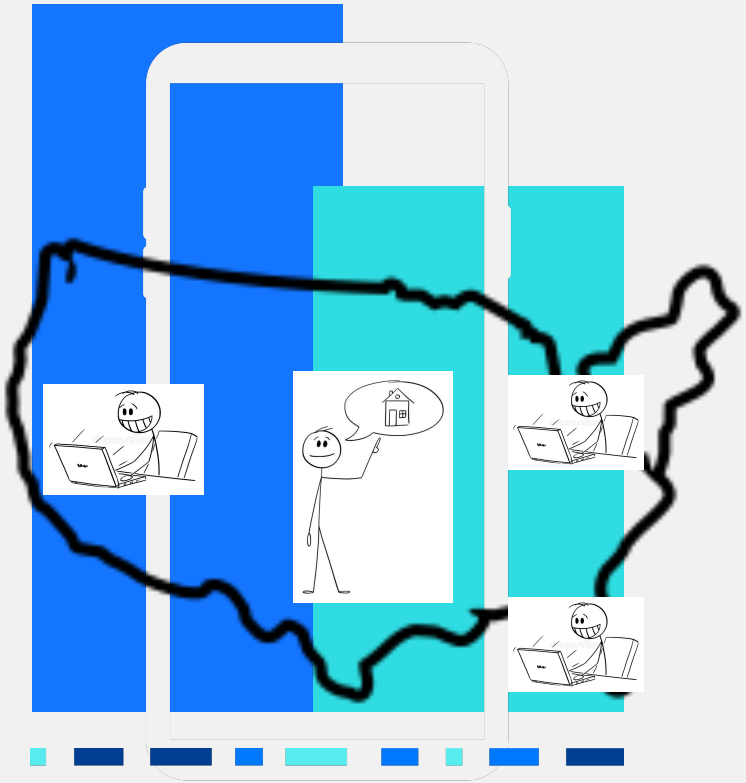
Non-Master Profile's IP count

Generally, master profile is the true account owner.
If the Non-Master profile has too much IP that sees the following conditions, we flag.

1. At least 50 miles away from Master Profile's frequent location
2. At least 3 times of the distant streaming activity
3. Exceeds threshold

Threshold Analysis: sub_ip_cnt=5 (In the sample we use 97.5 percentile)

```
pertentile:0, :0.0
pertentile:25, :0.0
pertentile:50, :0.0
pertentile:75, :1.0
pertentile:85, :1.0
pertentile:90, :2.0
pertentile:95, :4.0
pertentile:96, :4.0
pertentile:97, :5.0
pertentile:97.5, :5.0
pertentile:98, :6.0
pertentile:98.5, :7.0
pertentile:99, :8.0
pertentile:99.5, :11.0
pertentile:100, :63.0
```



Paramount

Scenario 3

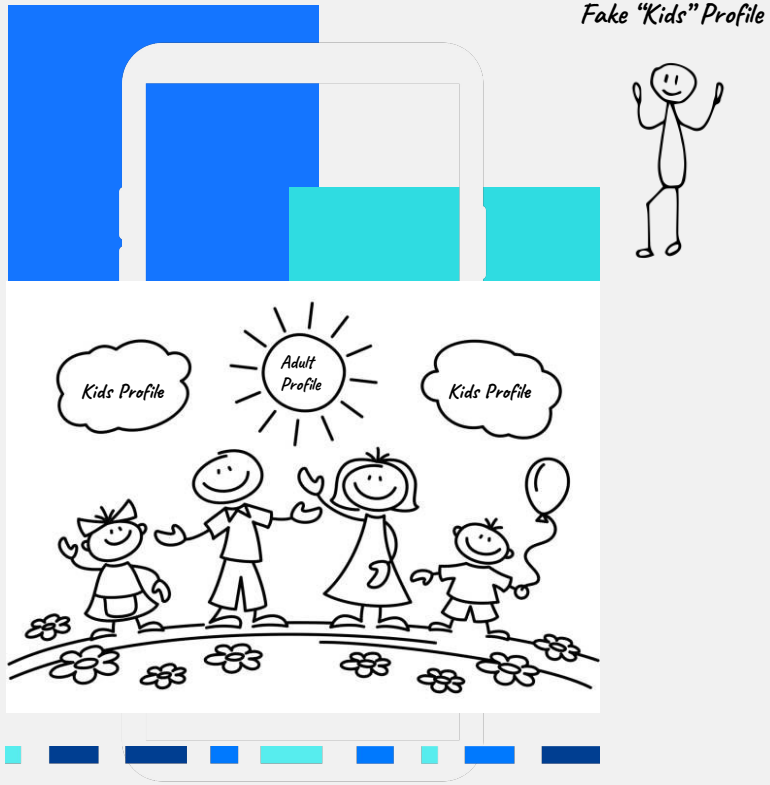
Kids Profile's IP count

Kids profile is designed for content rate and censorship.
If the Profile is recognized as kid profile and sees the following conditions,we flag.

1. At least 10 miles from its Master Adult Profile frequent Location
2. At least 3 times of the distant streaming activity
3. The distinct ip count exceeds the threshold

Threshold Analysis: kid_ip_cnt=1 (99 percentile, in the sample we use 2)

```
pertentile:0, :0.0
pertentile:25, :0.0
pertentile:50, :0.0
pertentile:75, :0.0
pertentile:85, :0.0
pertentile:90, :0.0
pertentile:95, :0.0
pertentile:96, :0.0
pertentile:97, :0.0
pertentile:97.5, :0.0
pertentile:98, :0.0
pertentile:98.5, :1.0
pertentile:99, :1.0
pertentile:99.5, :2.0
pertentile:100, :40.0
```



Paramount

Scenario 4

Same-Content Concurrent Streaming

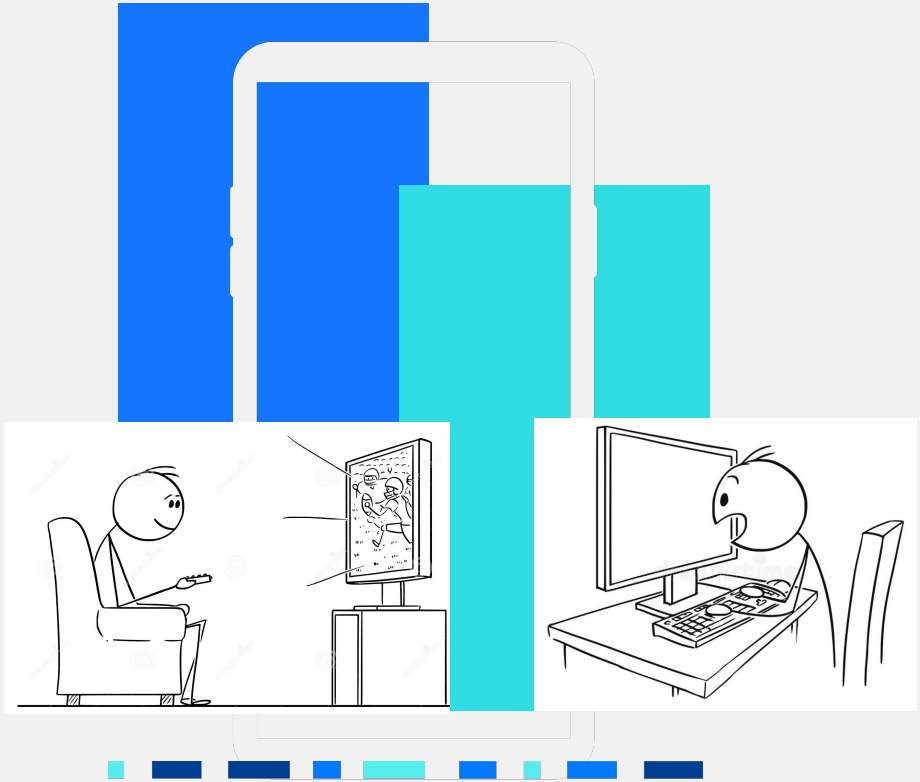
Same-content Concurrent Streaming usually happens when there is a live show or viral content that released weekly.

It is a very strong signal of password sharing, if the occurrence exceeds the threshold with the following condition, we flag

1. Streaming with different profile
2. At least 10 miles away from each other

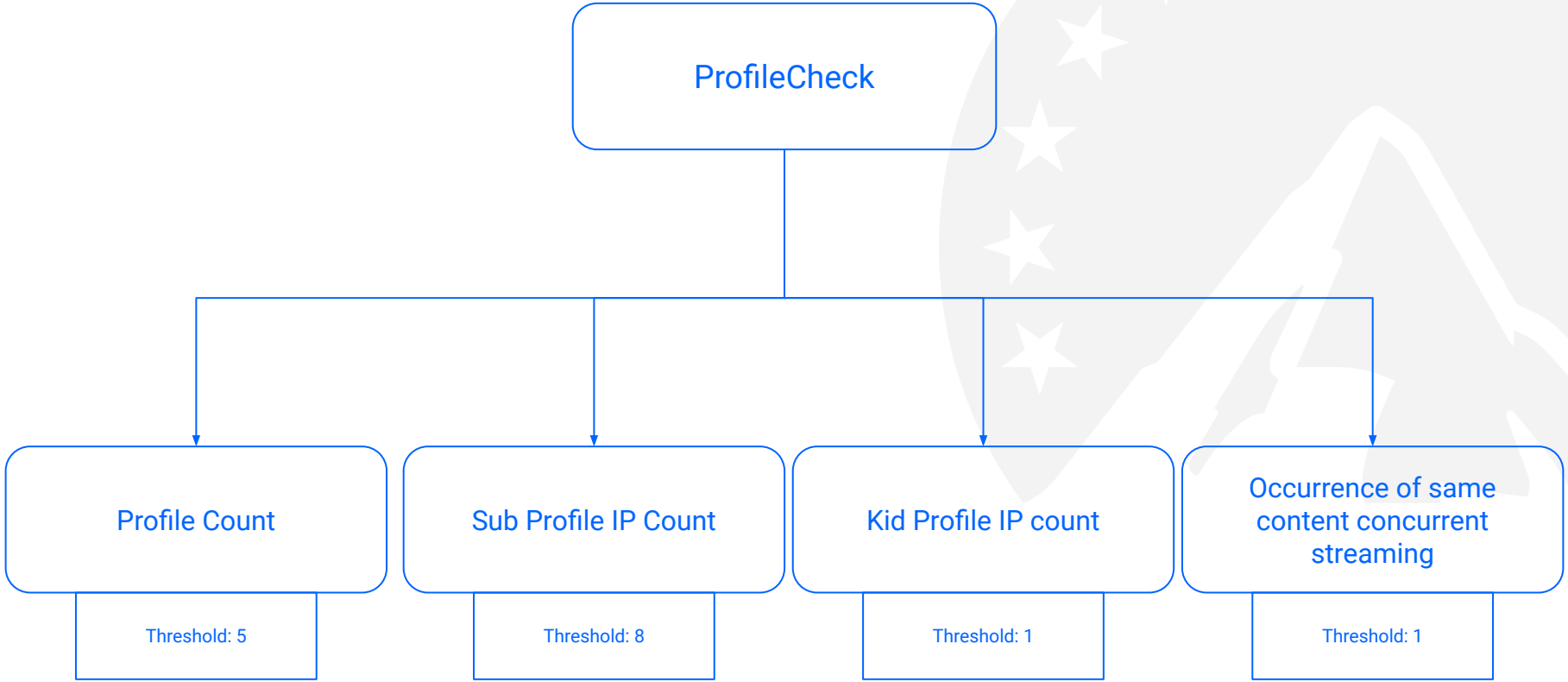
Threshold Analysis: ccrnt_strm_cnt=1 (99 percentile)

```
pertentile:0, :0.0
pertentile:25, :0.0
pertentile:50, :0.0
pertentile:75, :0.0
pertentile:85, :0.0
pertentile:90, :0.0
pertentile:95, :0.0
pertentile:96, :0.0
pertentile:97, :0.0
pertentile:97.5, :0.0
pertentile:98, :0.0
pertentile:98.5, :0.0
pertentile:99, :1.0
pertentile:99.5, :1.0
pertentile:99.6, :2.0
pertentile:99.7, :2.0
pertentile:99.8, :3.0
pertentile:99.9, :5.0
pertentile:100, :177.0
```



Paramount

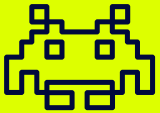
Overview: Profile Check Module



Alternative Embedding Algorithm

Embedding Algorithm:
Node2Vec vs GGVec vs PRONE

GGVec Hyper parameters Tuning

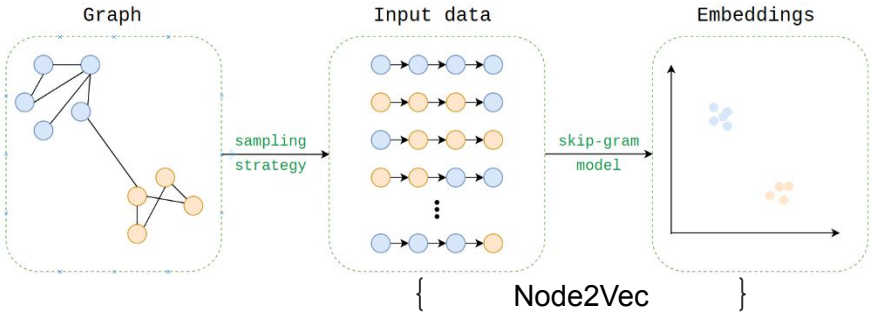


GGVec Node2vec Comparison

Paramount

Embedding Algorithms

Transform information carried in viewing pattern graph into structured data; Represent nodes in low dimensional vectors form.



Save Embeddings → Community Detection Module

Optional Algorithms:

Node2Vec: Most Popular, in use, time costly in our case

GGVec:A flexible algorithm. Best used on large graphs and for visualization.

ProNE:The fastest and most reliable sparse matrix/graph embedding algorithm.

Node2vec	GGvec	ProNE
70772066	70719223	70719223
70990328	70772066	70772066
71062077	70855697	70855697
71310731	70990328	70921967
71449026	71449026	71310731
71504677	71717953	71643093
71717953	71785686	71672628
71785686	71794933	71717953
71794933	71899139	71785686
72005769	72005769	71794933
8048182	8048182	71899139
9071770	8295940	8048182
		8295940
		9630504

Sample Size	1 Day; 50k users;				
	Total Catch	Reliable Catch	Extra Catch	Prct Reliable	Time for embedding Algorithm
GGVec	12	12	0	100%	21s
Node2vec	12	9	3	75%	>30mins
ProNE	14	10	4	71%	6s

GGVec Parameters Tuning

n_components: Number of individual embedding dimensions.

Learning rate: Optimization learning rate.

tol_samples : Optimization early stopping criterion.
This is the number of epochs to sample for loss stability.
Once loss is stable over this number of epochs we stop early.

Negative Ratio:Negative sampling ratio.
Setting this higher will do more negative sampling.
This is slower, but can lead to higher quality embeddings.

Max Epoch: Stopping criterion

order: Meta-level of the embeddings. Improves link prediction performance.
Setting this higher than 1 ~quadratically slows down algorithm

Some Advice from documentation:

1.keep order at 1.

2.keep negative_ratio low (~0.05-0.1),

3.keep learning_rate high (~0.1)

4. Use aggressive early stopping values.

Sample size	Paramters					
	N_components	total_samples	negative ratio	Max epoch	Flagged	Time for embeddings
50k users 30 days	20	Default	Default	200	2103	3m
100k users 30 days	20	30	0.05	150	4267	6m
	30	30	0.05	200	4341	8m
	15	30	0.05	100	3892	4m

GGvec vs. Node2vec

Time: 18s
Sample:50k users 1 day

Toal Caught: 16997 **(11k also caught by Node2Vec)**
Sample:300k users 30 days

Time: 40 mins
Sample:50k users 1 day

Total Caught: 17012
Sample:300k users 30 days

Paramount

Parallel Processing

Joblib Parallel into Data Processing & Abuse Detection

Explore optimal n_jobs set in Parallel Function

Joblib Parallel into Data Processing & Detection Process

Old: All Data - - -> Data Processing - - - > Abuse Detection

Now: All Data - - -> Split by user Log - - > Parallel Data Processing & combine target dataframes

Semi Dataframes - - -> Split by user Log - - - > Parallel Detection & combine results

Local Env: RAM 8GB	Detection	Parallel Detection
(Sample Size: 50k users 1 day data)	50m	12m
	Preprocessing	**Parallel Preprocess**
(Sample Size: 100k users 30 days data)	6h30m	3h20m

Sample size: 300k users 30 days	
N_job	time per task
50	0.0578
70	0.0597
100	0.0614
200	0.0616
500	0.0564
1000	0.0574

Pipeline Modification

Data fetching Optimization

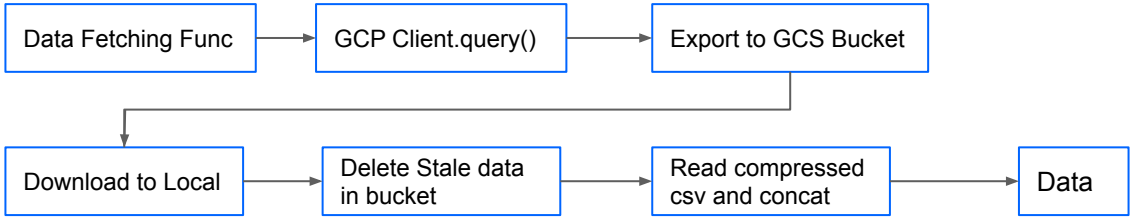
Skipping Logic Optimization

Data Fetching Optimization

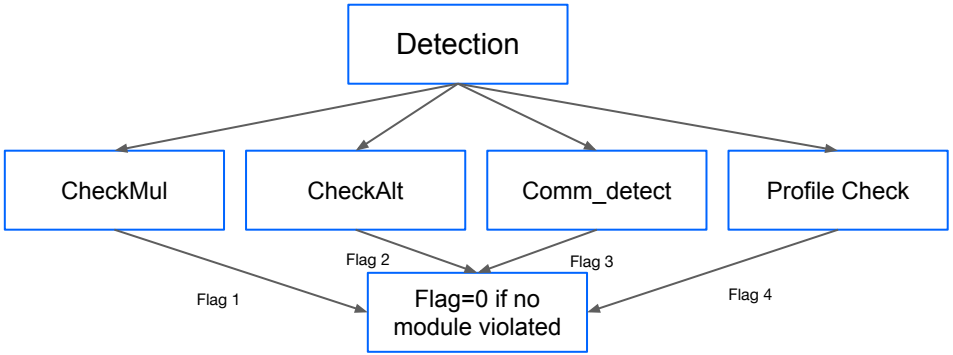
Sample Size: 100k 30 days Local 8GB RAM

Old: Time Spent: 1h20m

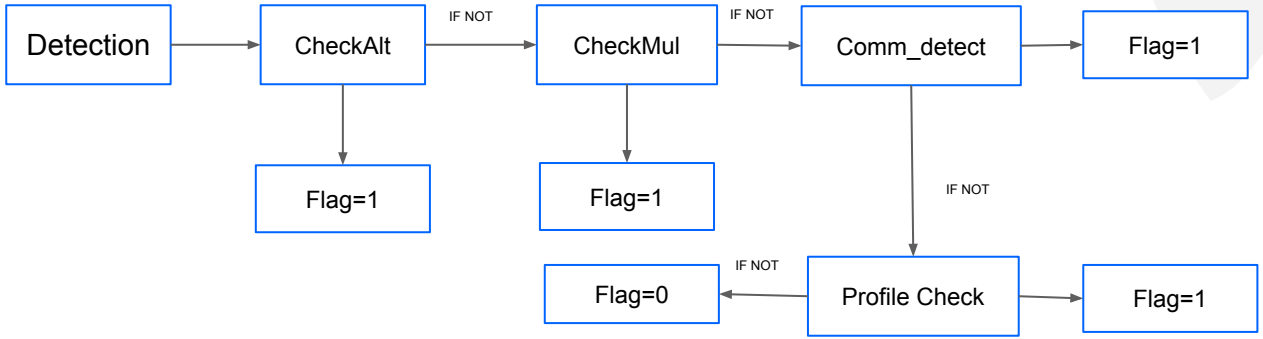
Now: Time Spent: 8m



Skipping Logic Optimization



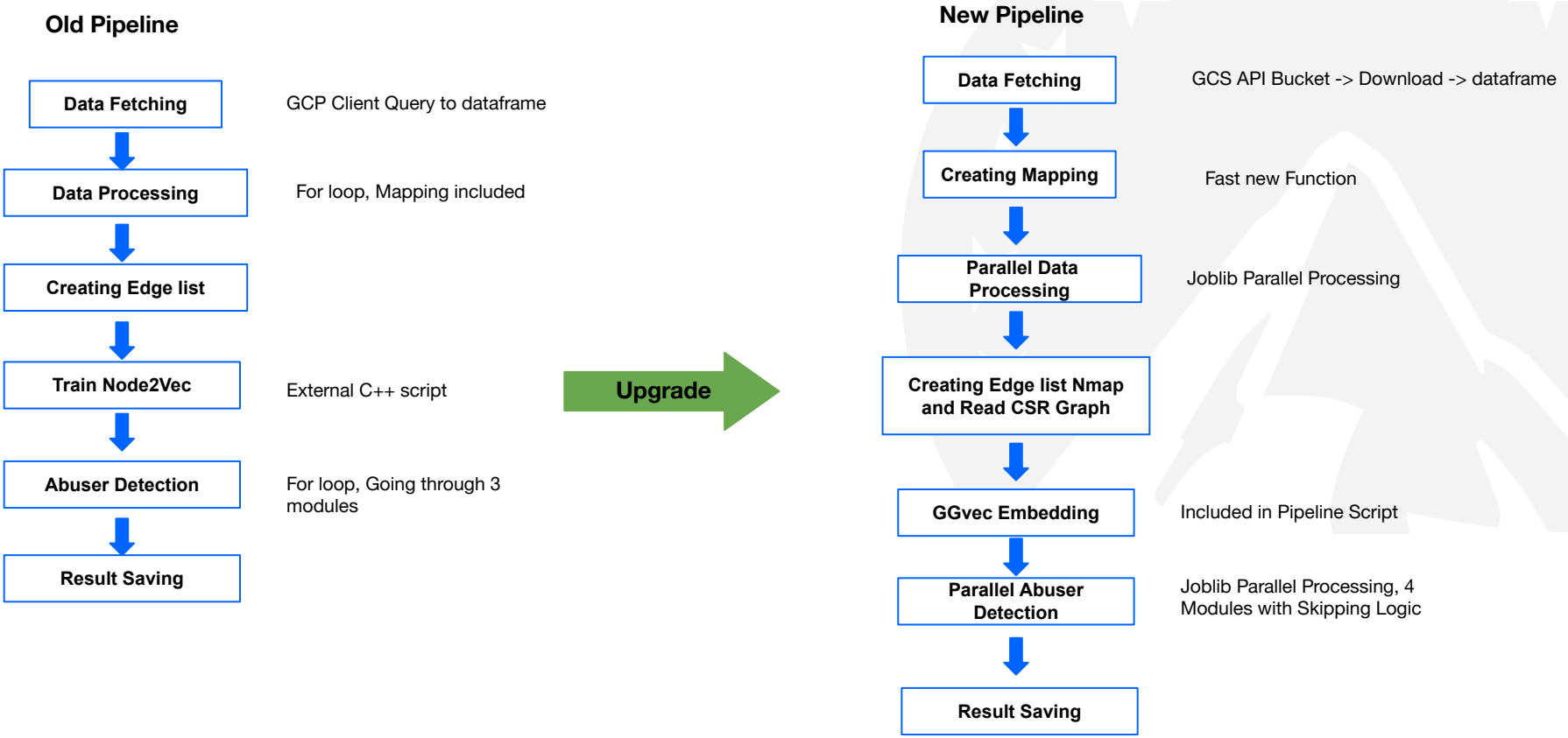
Old:
All records need to go through all modules



Now:
Records go CheckAlt first, if not flagged then go CheckMul, then Comm_detect, then profile, once one module in front has flagged, skip the rest modules

Same applies to Profile Check Module itself

Pipeline Comparison



Impacts

Scalability Improvements

Old Pipeline	Time spent	New Pipeline	Time spent	New Pipeline	Time spent
(sample size)100k 30 days		(sample size)100k 30 days		(sample size)300k 30 days	
Data Fetching	1h20m	Data fetching	3m	Data fetching	6m
Data Processing	6h30m	Creating Map	2s	Creating Map	7s
Creating Edgelist	8m	Parallel Data Processing	1h30m	Parallel Data Processing	4h40m
Train Node2vec	>10h	Creating Edgelist Nmap & Read CSR Graph	20s	Creating Edgelist Nmap & Read CSR Graph	1m30s
Detection(3 modules)	3h50m	GGVec Embeddings	40s	GGVec Embeddings	2m10s
Result Saving	1m	Parallel Detection	1h8m	Parallel Detection(4 Modules)	3h40m
		Result Saving	1s	Result Saving	1s
Total	23h	Total	2h40m	Total	8h30m

Robustness Improvements

(sample size: 300k users 30 days)	Old System	New System
Total Caught	41k	36.5k
CheckMul	8.1k	8.1k
CheckAlt	15.7k	15.7k
Profile	22.2k	8.6k
Comm	17k	17k

We had a **coarse** profile module in the old system, now replaced with **well-rounded and rational** profile modules to reduce potential false positive

Future Work:

1. Come up with more scenarios to enhance detection power
2. Plan actionable initiatives to tackle flagged users
3. Further Tuning GGVec Parameters to help achieve its maximum representational power
4. Improve scalability by exploring new parallel methods or by optimizing code logic
5. Building out Airflow Pipeline to make it run monthly

Paramount+

Thank You.

Q & A

Paramount+