PIMan: A Comprehensive Approach for Establishing Plausible Influence among GitHub Repositories

Md Omar Faruk Rokon
Risul Islam
Md Rayhanul Masud
Michalis Faloutsos

Computer Science, University of California Riverside

Problem Definition

How can we quantify the influence among repositories in online archives like GitHub?

Input: A list of GitHub repos

Output: A directed plausible influence graph of repositories

Challenge: How do combine repo-author, author-author, and author popularity?

Contribution

Our key contribution is a directed graph of plausible influence, PIGraph

- Tuneable with influence threshold
- Reveals significant collaboration
- Reveals interesting lineage

Proposed method

Step 1: Compute three influence scores from

- repo-author interaction,
- author-author interaction, and
- author popularity

Step 2: Compute ***PIScore*** considering time and combining these three scores

Step 3: Create the Directed ***PIGraph*** based on ***PIScore***

Step (1/1): Compute repo-author interaction score, *RAI*

We consider all repo level interactions from *R2* to *R1*,

$$RAI = \frac{SS + FS + WS + CS}{4}$$

Where,

$SS \leftarrow$ *A2* stars *R1*
$FS \leftarrow$ *A2* forks *R1*
$WS \leftarrow$ *A2* watches *R1*
$CS \leftarrow$ *A2* comments on *R1*

Step (2/1): Compute author-author interaction score, *AAI*

We consider all author level interactions from *R2 to R1*,

$$AAI = \frac{FS + FS_{O_R} + SS_{O_R} + WS_{O_R} + CS_{O_R}}{5}$$

Where,

FS $\leftarrow$ *A2* follows *A1*
FS_{OR} $\leftarrow$ *A2* forks other repos of *A1*
SS_{OR} $\leftarrow$ *A2* stars other repos of *A1*
WS_{OR} $\leftarrow$ *A2* watches other repos of *A1*
CS_{OR} $\leftarrow$ *A2* comments on other repos of *A1*

Step (3/1): Compute author popularity, *APop*

1. Create a multi-digraph
2. An edge (u,v) exists when author u:
    a. Follow
    b. Fork
    c. Star
    d. Watch
    e. Comment
    f. Contribute.
3. Weight adjustment
    a. Normalize the weights in such a way that more frequent type edge gets less weight
4. Upon weight adjustment, applied weighted HITS algorithm
5. Gives us producer score (authority) and consumer score (hub)
6. Popularity of author A1,
    a. *APop = Producer score of A1 + Consumer score of A1*

Step2: Combine *RAI, AAI* and *APop*

To compute *PIScore* of *R1* to *R2*,

1. Consider repo creation time:

   R2 is created earlier than *R1*, *PIScore* = 0

2. Else

$$PIScore = w_{RA} * RAI + w_{AA} * AAI + w_A * APop$$

Where,

$w_{RA} \leftarrow$ weight for *RAI* score
$w_{AA} \leftarrow$ weight for *AAI* score
$w_A \leftarrow$ weight for *APop* score

Step3: Create the directed ***PIGraph*** *(V, E)*

$V \leftarrow$ the repository set

$E \leftarrow$ the set of edges

- We consider an edge *e* from R1 to R2 if *PIScore(R1, R2)* >= *PIT*
  - Edge weight *w(e)* = *PIScore(R1, R2)*

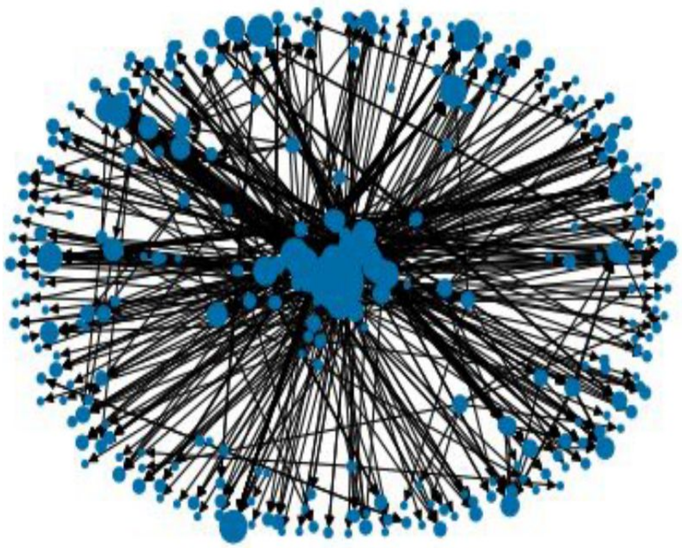
Dataset

Size: 2089 Java malware repositories

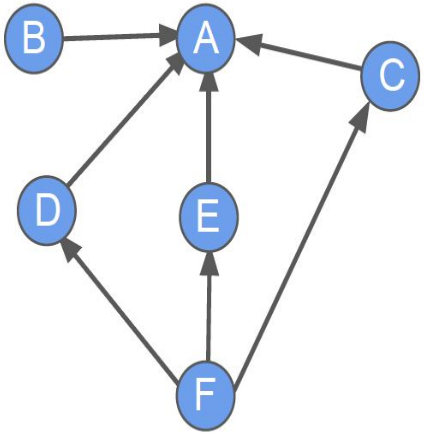
We collect and store

1. Repository level interaction
2. Author level interaction

PIGraph: Tuneable with a threshold

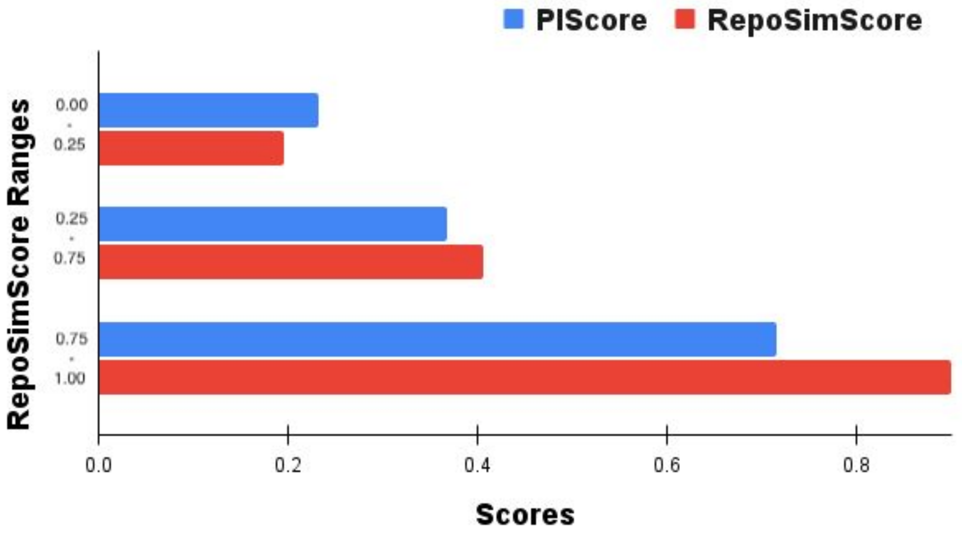


Dense graph with 426 nodes, and 1191 edges where PIScore >= 0.25



Sparse graph with 6 nodes, and 7 edges where PIScore >= 0.7

Relationship between influence and similarity: highly correlated



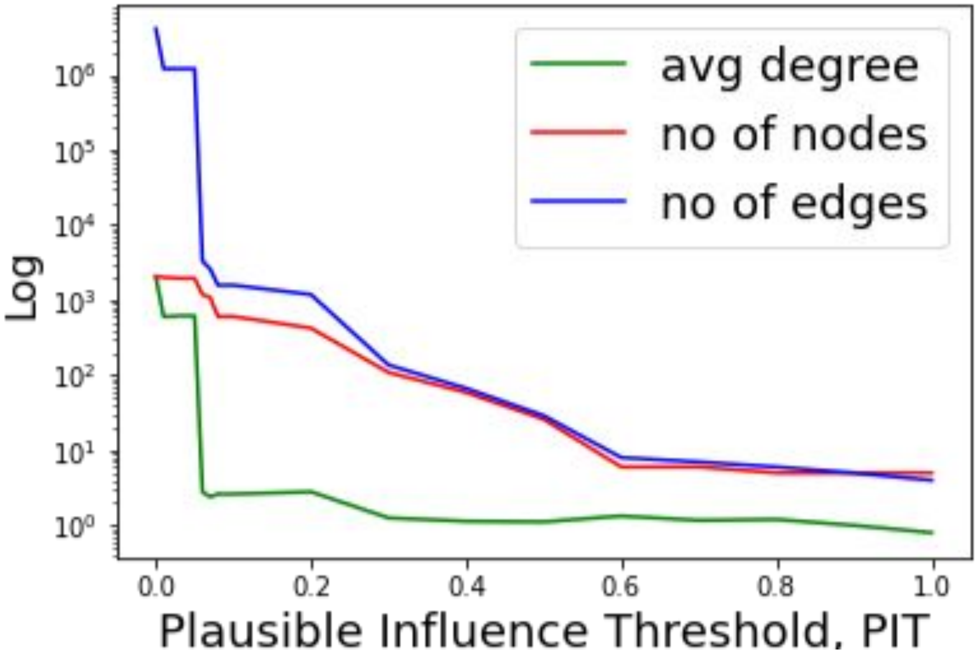
Highly correlated where a Spearman coefficient ρ = 0.79 with p-value = 1.26e-19.

- Randomly selected 90 pairs of repos
- From 3 ranges of repo similarity
- Get the influence scores
- Plot the average of these two scores

Consider repos X = “androidtrojan1/android trojan” and Y = “vaginessa/android-overlay-malware-example”

- X and Y are highly similar SimScore = 0.9
- X has influenced Y with a PIScore = 0.85
    - Author “vaginessa” interacts with “androidtrojan1” in multiple ways
    - “Vaginessa” follows, stars, and forks 5 repos of “androidtrojan1”

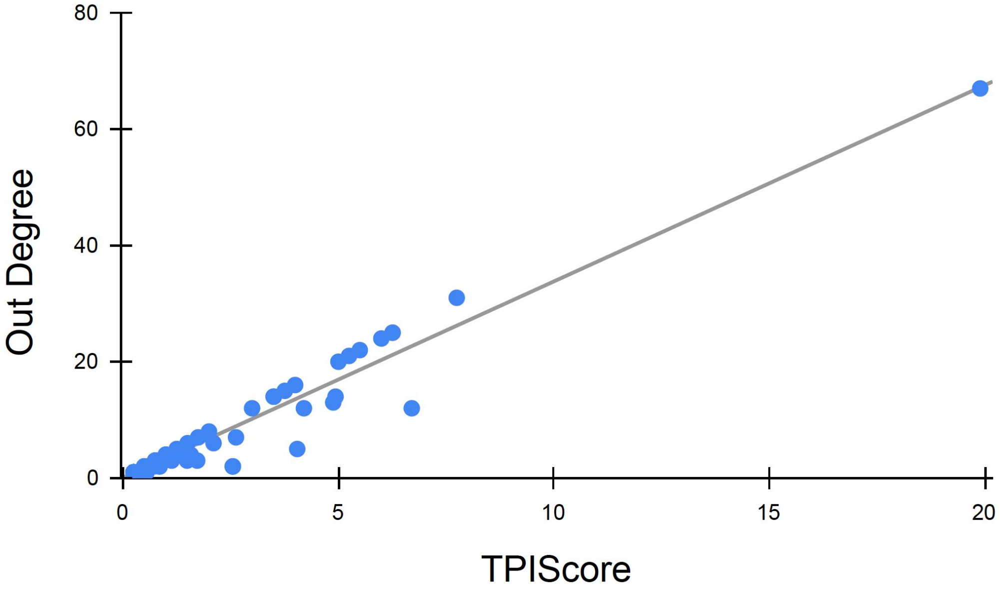
Effect of PIT: Increasing threshold reduces the network size



- Guidance for selecting PIT value
- Knee between 0.2 and 0.4
- Select PIT = 0.25 for non-trivial influence

The distribution and intensity of influence: # of directly influenced repos follow skewed distribution

- 39% repos with zero direct influence
- 8% repos influence at least 20 repos
- Most influential repo influences 67 repos



Number of directly influenced repositories (Outdegree) vs Total Plausible Influence (TPIScore) exhibits a linear correlation

Evidence of collaboration: Creates highly collaborative clusters

Considering PIGraph with significant influence threshold PIT = 0.25, we get

- 28 connected components
  - 7% of them have more than 15 repos
  - 71% of them have less than 5 repos

Manually validated 2 components

- Component 1
  - Contains 16 repos focused on android malware
- Component 2
  - Contains 235 repos from 3 malware families: keylogger, botnet and trojan

Lineage: Highly influential repos spawn multiple repo “families”

Information Collection Malware
vaginessa/android-overlay-malware-example
June 17, 2015

User Credential Stealing Malware

1. darrylburke/AndroidMalware-Example, Dec 22, 2016
2. CCrashBandicot/AndroidKey-Logger - Jan 30, 2017
3. chaojianhu/Android-Side-Channel-Malware - May 25, 2018
4. chaojianhu/Malware-of-Android - May 25, 2018

Ransomware Malware

1. LucaBongiorni/AndroMalware - Dec 12, 2015
2. sivazozo/Android-RansomWare - May 18, 2018
3. 5l1v3r1/JavaRansomware - March 18, 2020

Remote Access Trojan (RAT) Malware

1. scrack/android_trojan - Sept 14, 2016
2. ClovisIRex/AndroidRAT - Sept 12, 2017
3. 5l1v3r1/android-trojan-service - June 17, 2020

We find 19 repos that have directly influenced at least 10 repos

Influence vs Popularity: Influence provides a significantly different perspective compared to popularity!

No	Influential repositories using PIMan	Popular repositories using *RepoPop*
1	00aj99/AndroidMalware-Example	tiagorlampert/sAINT
2	CCrashBandicot/android-_trojan	adonespitogo/AdoBot
3	CCrashBandicot/Android-KeyLogger	M1Dr05/IsTheApp
4	molotof/sAINT	tomgersic/AndroidKey-Logger
5	5l1v3r1/AndroidRansom-Ware	Mandyonze/Droid-Sentinel
6	CristianTuretta/MAD-Spy	PanagiotisDrakatos/Java-Ransomware
7	tiagorlampert/sAINT	harshalbenake/Android-Elite-Virus
8	Mandyonze/Droid-Sentinel	moloch–/Yoshimi-Botnet
9	androidtrojan1/ android_trojan	androidtrojan1/ android_trojan
10	un4ckn0wl3z/Psyber-Project	siberas/sjet

Top 10 influential repositories identified by PIMan and popularity metric RepoPop

Conclusion

Our approach aims to develop methods to identify;

- Inter-repository social-level influence
- Flexible and powerful representation of influence using PIGraph
- Lineage and families of influence

THANK YOU

FEEL FREE TO ASK ANY QUESTION