AIS3

教育部資安人才培育計畫
Information Security Incubation Program

教育部先進資通安全實務人才培育計畫

113年度新型態資安實務暑期課程

聚類分析技術 - 惡意軟體變體辨識方法研究

Capa、巴卡阿卡、Jaccard、咪卡巴卡、嗝(๑>ᴗ<)ง

組別: 情資運用及防禦 - B8
組員: 鄭宇兩、曹凱翔、吳玥儒、蔡士煒

組員

鄭宇兩
貝坦

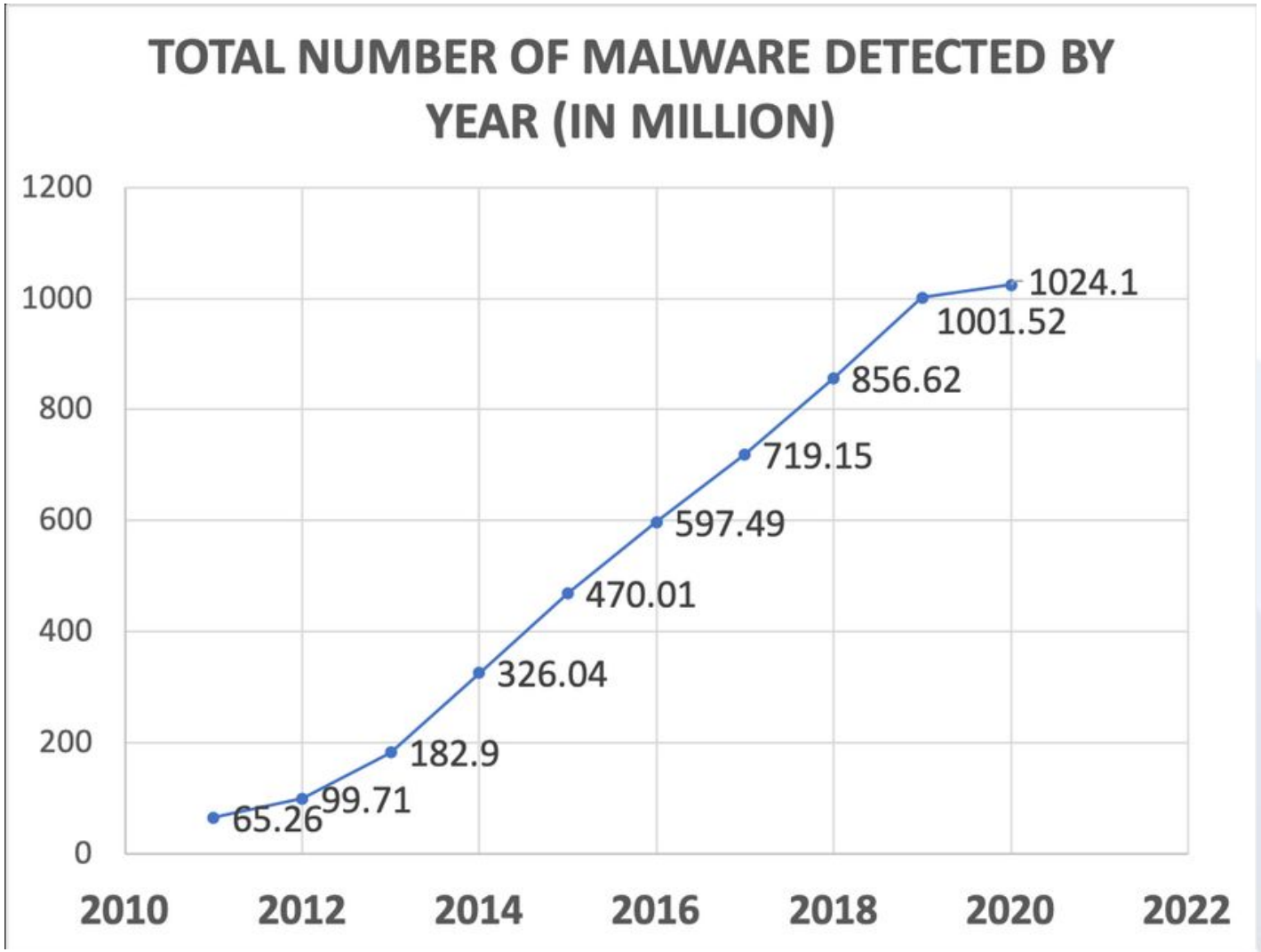
曹凱翔
l3obo

吳玥儒
slimu

蔡士煒
小蔡

目錄

動機



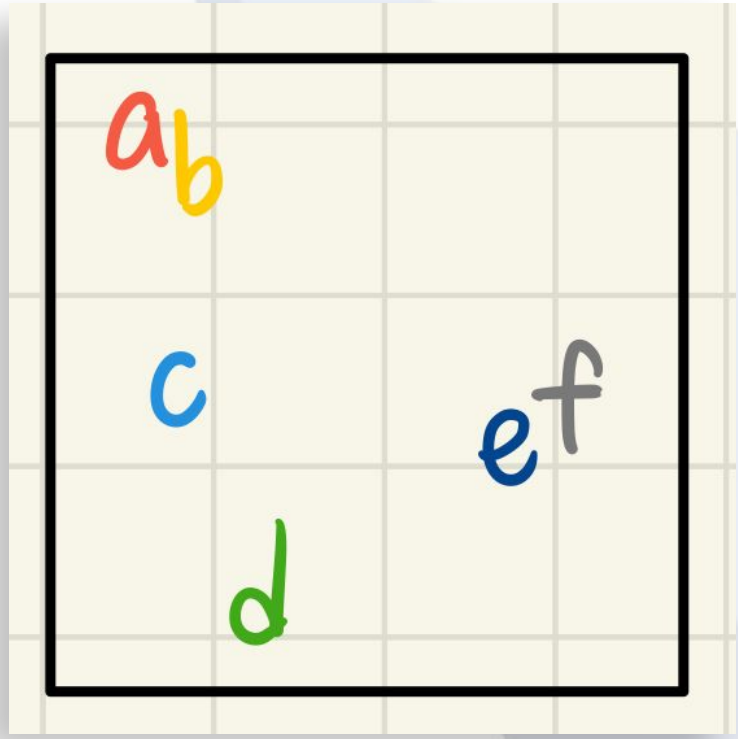
Total number of malware detected by year (in million)

目標

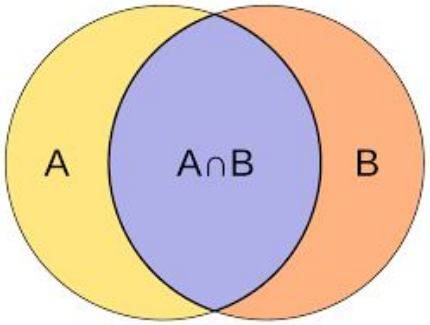
自動解析未知惡意軟體與其他惡意軟體間的 相似度

協助逆向分析人員

更有效識別和分類惡意軟體。



架構



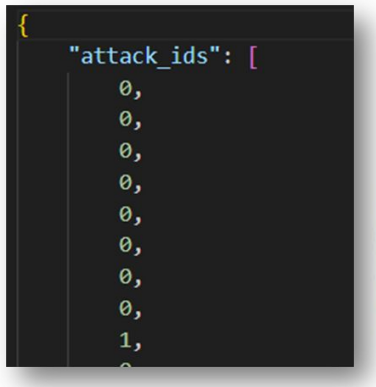
Jaccard Index

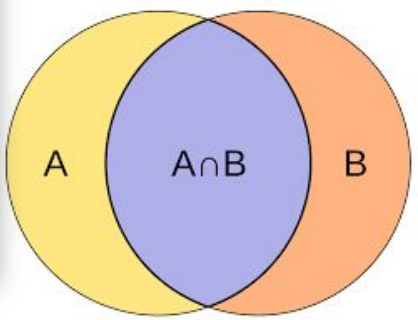
Malware → CAPA

方法一

方法二 → 文本向量化 → K-means

方法三





病毒來源

- Malware Bazzar
  - 樣本數量多且有明確分類
  - 可用API大量抓取

Date (UTC)	SHA256 hash	Type	Signature	Tags
2024-08-02 19:05	e9837fc1d609e00844525...	exe	AsyncRAT	AsyncRAT exe RAT
2024-08-02 18:50	82e4fa04f1bbb083ff8657...	elf	Mirai	mirai
2024-08-02 18:50	8184f3e0731d2df9d58c0...	elf	Mirai	mirai
2024-08-02 18:50	ddb07ed1f41dc60aa797...	elf	Mirai	mirai
2024-08-02 18:16	f451f253429a5ea6156e4...	exe		exe signed
2024-08-02 18:16	848260ba966228c4db25...	exe	CobaltStrike	CobaltStrike exe
2024-08-02 18:16	9daffaea889ac9a0a94e7...	exe	Vidar	exe vidar
2024-08-02 18:16	7f08c2afa083a9814989c...	exe	PureLogStealer	exe PureLogStealer

CAPA分析架構

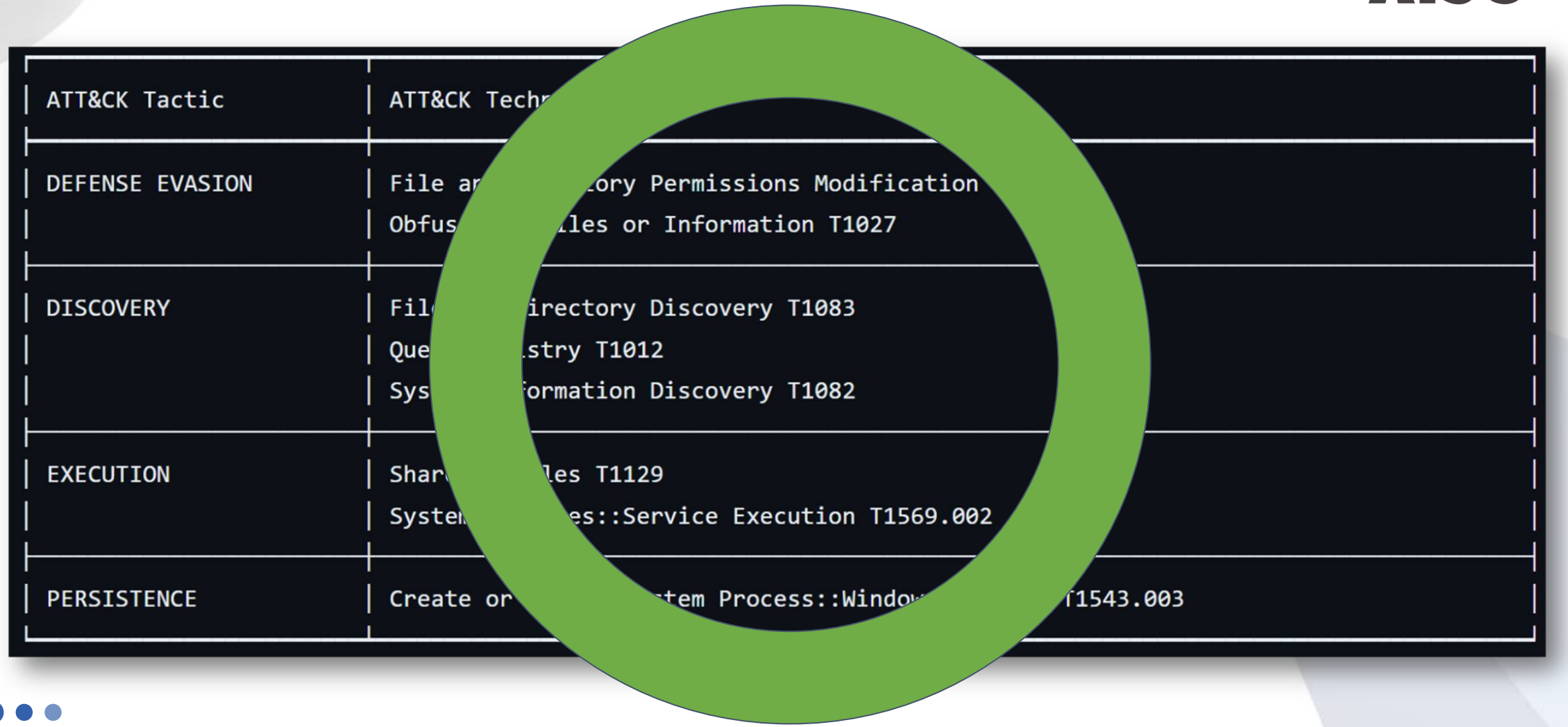
- 基本資料
- MITRE ATT&CK
- MBC(Malware Behavior Catalog Objective)
- Capability

CAPA分析資料那麼多 要選哪些？

基本資料

md5	8 06d8
sha1	5ff4
sha256	ed01ebfb 0439c6e5babe8e080e41aa
analysis	static
os	windows
format	pe
arch	i386
path	C esk xe

ATT&CK Tactic



ATT&CK Tactic	ATT&CK Techn
DEFENSE EVASION	File ar ory Permissions Modification Obfus les or Information T1027
DISCOVERY	Fil irectory Discovery T1083 Que istry T1012 Sys formation Discovery T1082
EXECUTION	Shar les T1129 Syste es::Service Execution T1569.002
PERSISTENCE	Create or tem Process::Windo T1543.003

MBC Objective

MBC Objective	MBC
CRYPTOGRAPHY	Data::AES [C0027.001] t Data::RC4 [C0027.009] ption Key::RC4 KSA [C0028.002] rate Pseudo-random Sequence [C0021]
DATA	ksum::CRC32 [C0032.001] ression Library [C0060] e Data::XOR [C0026.002]
DEFENSE EVASION	d Files or Information::Encodin d Algorithm [E1027.m02] Ob iles or Information::Encr ndard Algorithm [E1027.m05]
DISCOVERY	Code Dis 046.001] File and Direc System Information Discovery [E1082]

Capability

Capability	Namesp
hash data with CRC32	/crc32
encode data using XOR	g/xor
encrypt data using AES (5	yption/aes
reference AES constants (5 ma	encryption/aes
encrypt data using RC4 KSA (3 ma	ion/encryption/rc4
generate random numbers using the D	lation/prng/lcg
extract resource via kernel32 fu	urce
get common file path (3 matc	ile-system
set current directory (3	system
copy file	em/copy
create directory (2	em/create
check if file exists (4	host stem/exists
get file attributes (6 ma	host-in -system/meta
get file size (2 matches)	host-intera le-system/meta
set file attributes	host-interactio /file-system/meta
read file on Windows (3 matches)	host-interaction/file-system/read
write file on Windows (2 matches)	host-interaction/file-system/write
check mutex	host-interaction/mutex

那又要怎麼確定CAPA分析結果是正確的呢？

確認準確率

- 提供了推測的地方
  - $ capa.exe malware.exe -vv
- 不信就自己逆看看

確認準確率

```
encode data using XOR (15 matches)
namespace   data-manipulation/encoding/xor
author      moritz.raabe@mandiant.com
scope       basic block
att&ck      Defense Evasion::Obfuscated Files or Information [T1027]
mbc         Defense Evasion::Obfuscated Files or Information::Encodin
                                    Data::Encode Data::XOR [C0026.002]

basic block @ 0x402D25 in function 0x402A76
```

確認準確率

- 把自己跟前一位做 xor

```
loc_402D25:
mov     ebx, [eax-4]
xor     [eax], ebx
```

```
*v28 ^= *(v28 - 1);
```

確認準確率

- 把自己跟前一位做 xor

```
loc_402D25:
mov     ebx, [eax-4]
xor     [eax], ebx
```

```
*v28 ^= *(v28 - 1);
```

Data::Encode Data::XOR [C0026.002]

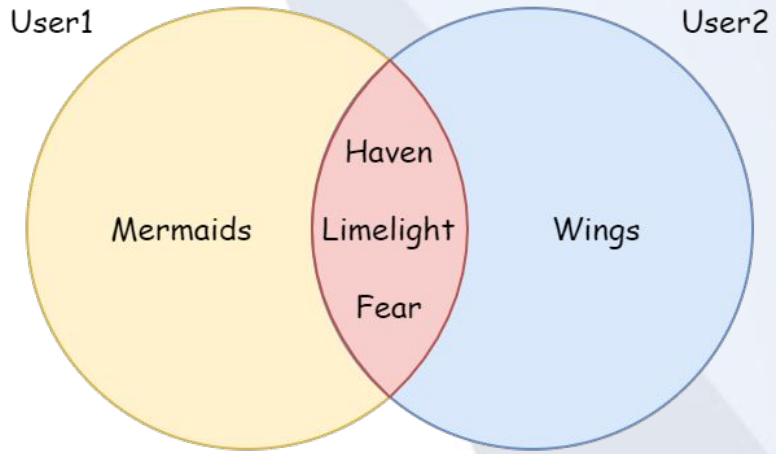
Jaccard Index介紹

- File and Directory Discovery > 'File' 'and' 'Directory' 'Discovery'
- System and Directory Discovery > 'System' 'and' 'Directory' 'Discovery'

交集 = Information, and, Discovery

聯集 = File, System, and, Directory, Discovery

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{5} = 0.6$$



用白話文談數學公式 - Jaccard Index

方法一

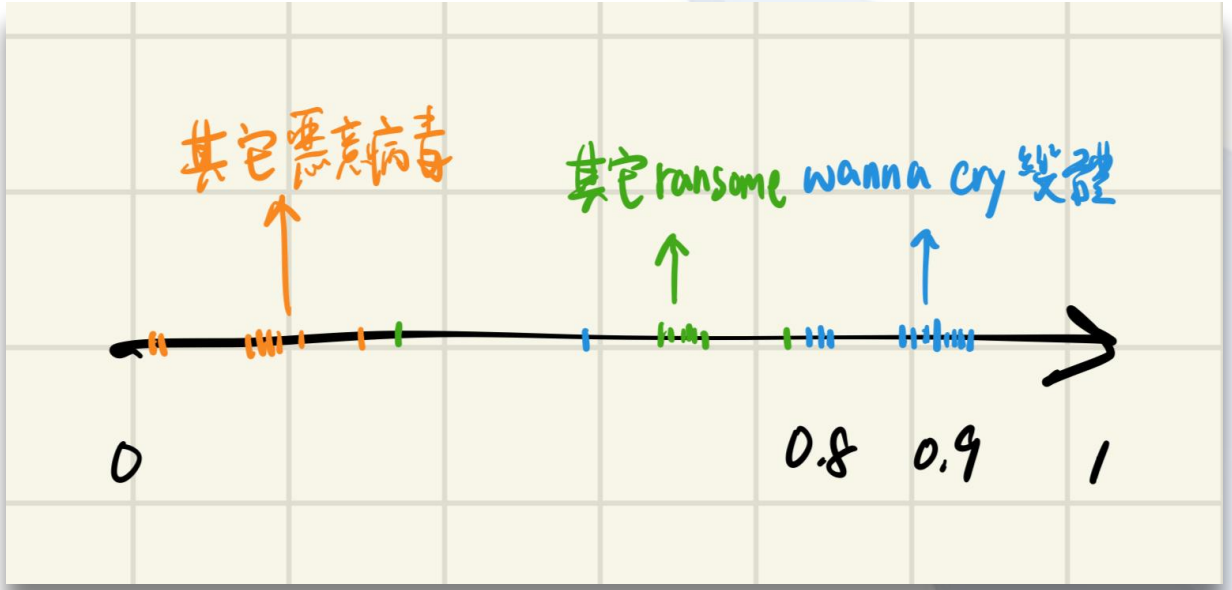
提取CAPA報告特徵

- 從編號下手
- 開一個陣列，紀錄有無該編號

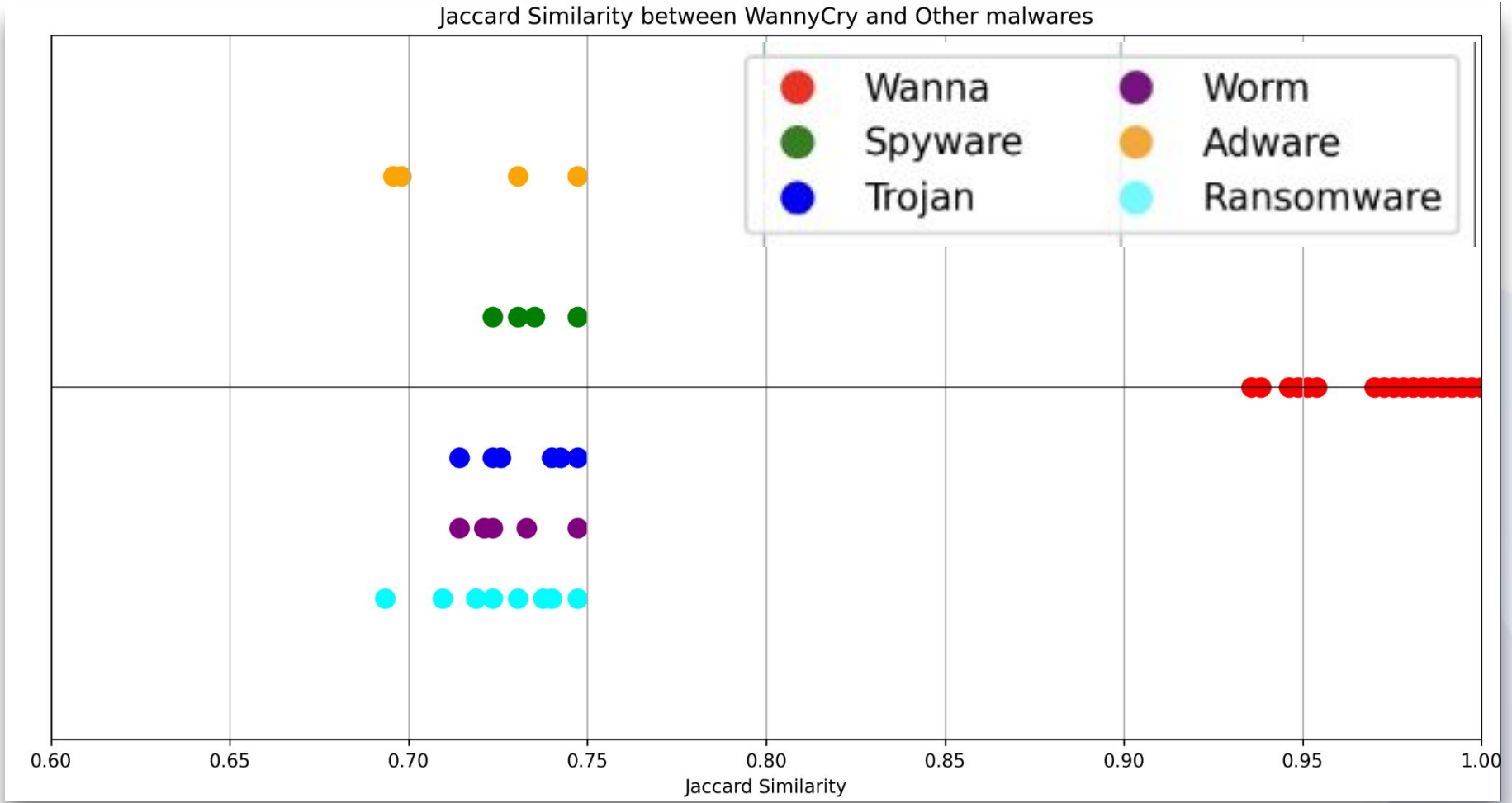
```
wannacry_sample = ['T1001', 'T1003', 'T1005','C0011', 'C0004', 'C0002']
wannacry_vriant = ['T1005','C0011', 'C0004']
```

Jaccard Index比較類別

- WannaCry變體
- 其他勒索軟體
- 其他惡意程式



實驗分析


Jaccard Similarity between WannyCry and Other malwares
Wanna
Spyware
Trojan
Worm
Adware
Ransomware
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
Jaccard Similarity

方法二

使用技術

- K-means
- CountVectorizer
- SentenceTransformer
- PCA

資料前處理

ATT&CK Tactic	ATT&CK Technique
COLLECTION	Input Capture::Keylogging T1056.001
DEFENSE EVASION	Hide Artifacts::Hidden Window T1564.003 Obfuscated Files or Information T1027 Obfuscated Files or Information::Indicator Removal from Tools T1027.005
DISCOVERY	Application Window Discovery T1010 System Information Discovery T1082
EXECUTION	Shared Modules T1129

MBC Objective	MBC Behavior
ANTI-STATIC ANALYSIS	Executable Code Obfuscation::Argument Obfuscation [B0032.020] Executable Code Obfuscation::Stack Strings [B0032.017]
COLLECTION	Keylogging::Polling [F0002.002]
DATA	Encode Data::XOR [C0026.002]
DEFENSE EVASION	Obfuscated Files or Information::Encoding-Standard Algorithm [E1027.m02]
DISCOVERY	System Information Discovery [E1082]
FILE SYSTEM	Writes File [C0052]
MEMORY	Allocate Memory [C0007]
PROCESS	Allocate Thread Local Storage [C0040] Check Mutex [C0043] Create Mutex [C0042] Set Thread Local Storage Value [C0041] Terminate Process [C0018]

```
all_attack_ids = ['T1001', 'T1003', 'T100
                  'T1027', 'T1029', 'T10
                  'T1052', 'T1053', 'T
                  'T1080', 'T1082', 'T
MBC = ['C0011', 'C0004', 'C0002', 'C0014'
  'C0068', 'C0069', 'C0059', 'C0027', 'C00
  'C0019', 'C0032', 'C0024', 'C0060', 'C00
  'C0016', 'C0045', 'C0046', 'C0016', 'C00
  'C0037', 'C0023', 'C0057', 'C0007', 'C00
```

001_analysis.txt

```
T1010, T1027, T1056, T1082, T1129, T1564

C0026, C0052, C0007, C0040, C0042, C0043, C0041, C0018
```

002_analysis.txt

```
T1027, T1056, T1082, T1129

C0019, C0026, C0046, C0052, C0007, C0040, C0042, C0043, C0017, C0038, C0054, C0041, C0018
```

003_analysis.txt

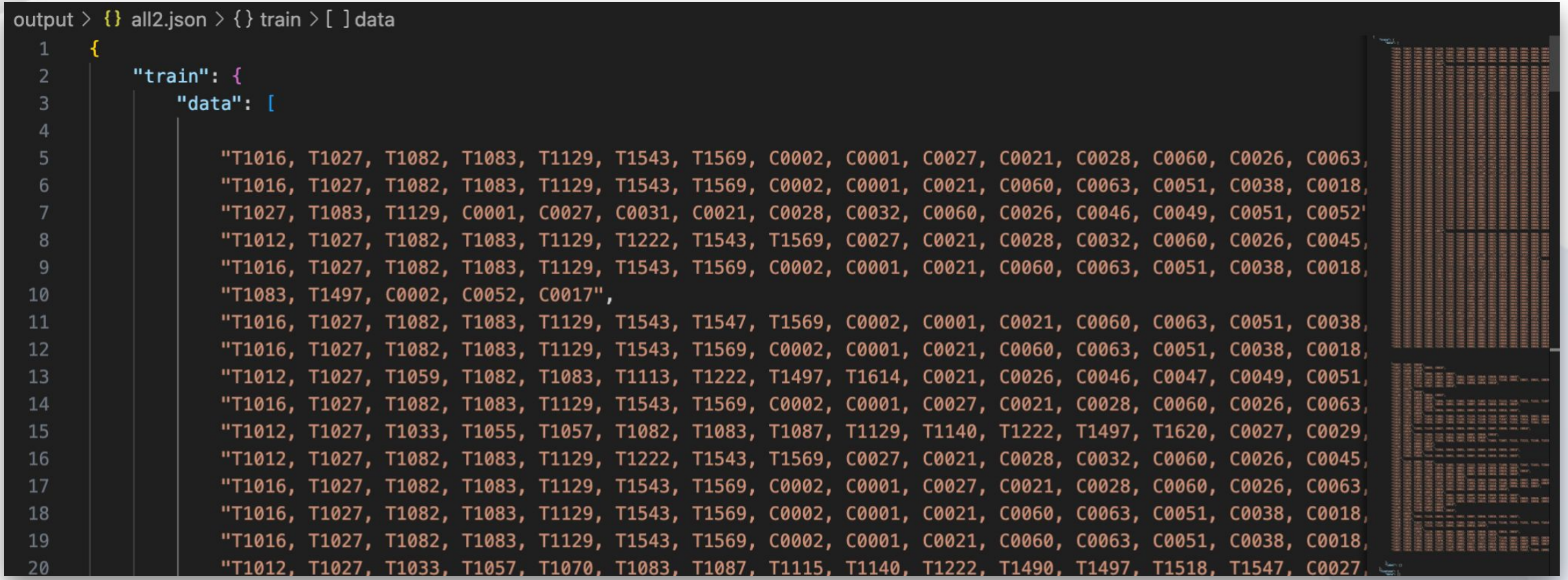
```
T1083, T1129

C0046, C0048, C0047, C0052, C0040, C0017, C0018
```

004_analysis.txt

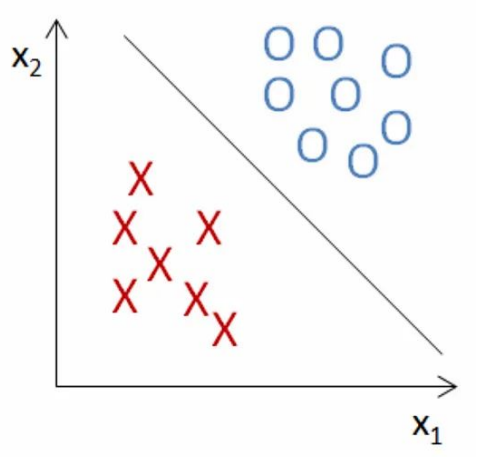
```
T1027, T1033, T1057, T1059, T1070, T1082, T1083, T1112, T1129, T1222, T1490, T1518, T1547,

C0027, C0031, C0029, C0028, C0045, C0047, C0049, C0063, C0051, C0050, C0052, C0007, C0042,
```

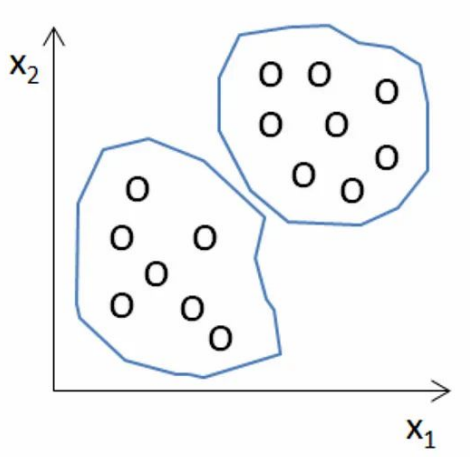

output > {} all2.json > {} train > [ ] data
{
"train": {
"data": [
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0027, C0021, C0028, C0060, C0026, C0063,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038, C0018,
"T1027, T1083, T1129, C0001, C0027, C0031, C0021, C0028, C0032, C0060, C0026, C0046, C0049, C0051, C0052'
"T1012, T1027, T1082, T1083, T1129, T1222, T1543, T1569, C0027, C0021, C0028, C0032, C0060, C0026, C0045,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038, C0018,
"T1083, T1497, C0002, C0052, C0017",
"T1016, T1027, T1082, T1083, T1129, T1543, T1547, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038, C0018,
"T1012, T1027, T1059, T1082, T1083, T1113, T1222, T1497, T1614, C0021, C0026, C0046, C0047, C0049, C0051,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0027, C0021, C0028, C0060, C0026, C0063,
"T1012, T1027, T1033, T1055, T1057, T1082, T1083, T1087, T1129, T1140, T1222, T1497, T1620, C0027, C0029,
"T1012, T1027, T1082, T1083, T1129, T1222, T1543, T1569, C0027, C0021, C0028, C0032, C0060, C0026, C0045,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0027, C0021, C0028, C0060, C0026, C0063,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038, C0018,
"T1016, T1027, T1082, T1083, T1129, T1543, T1569, C0002, C0001, C0021, C0060, C0063, C0051, C0038, C0018,
"T1012, T1027, T1033, T1057, T1070, T1083, T1087, T1115, T1140, T1222, T1490, T1497, T1518, T1547, C0027,

K-means介紹

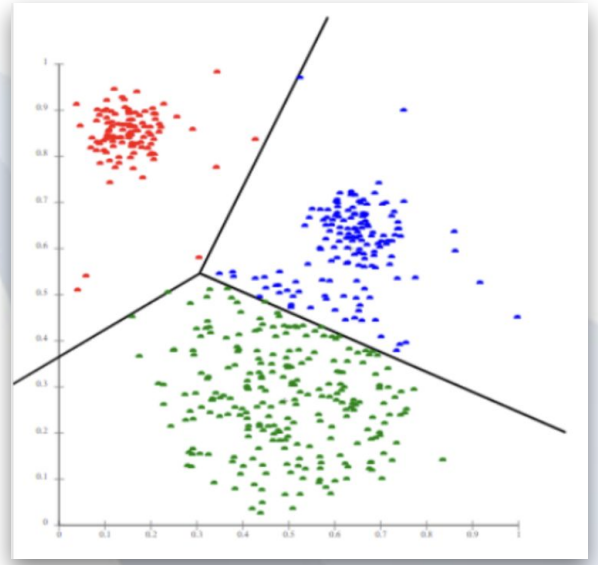
- 分群
- 非監督式學習



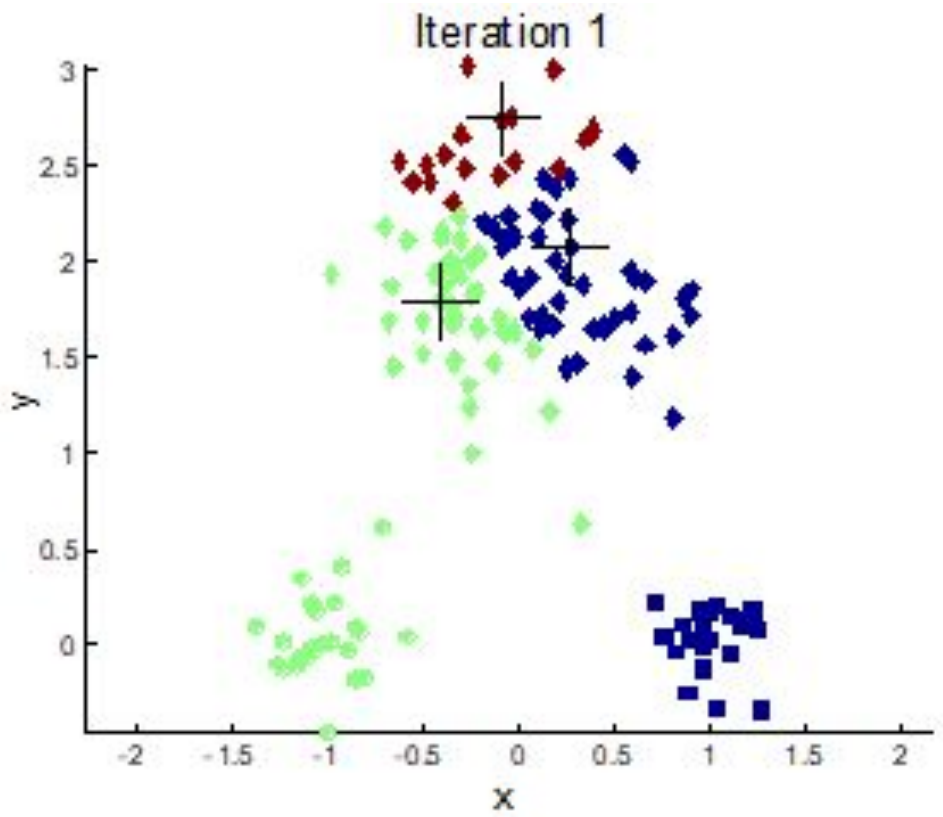
監督式學習



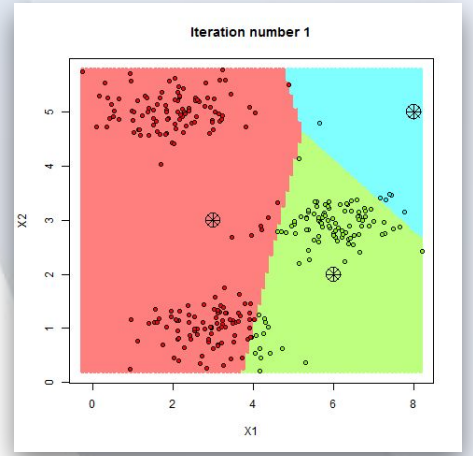
非監督式學習



K-means介紹



設k=3



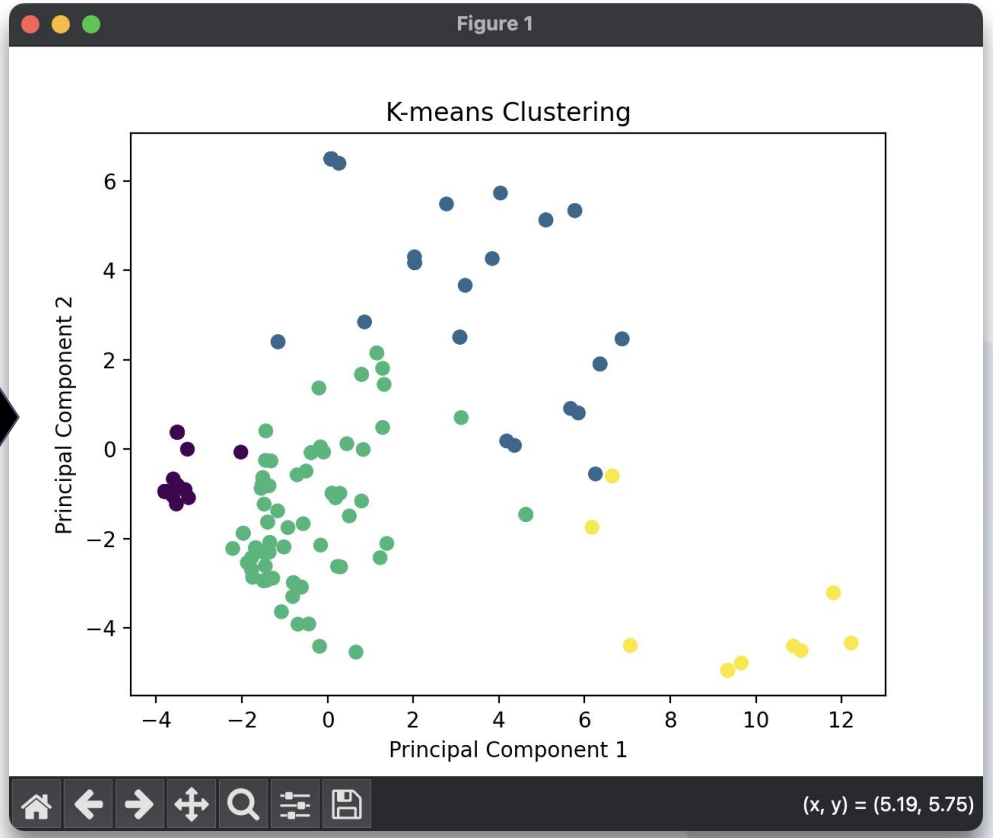
CountVectorizer

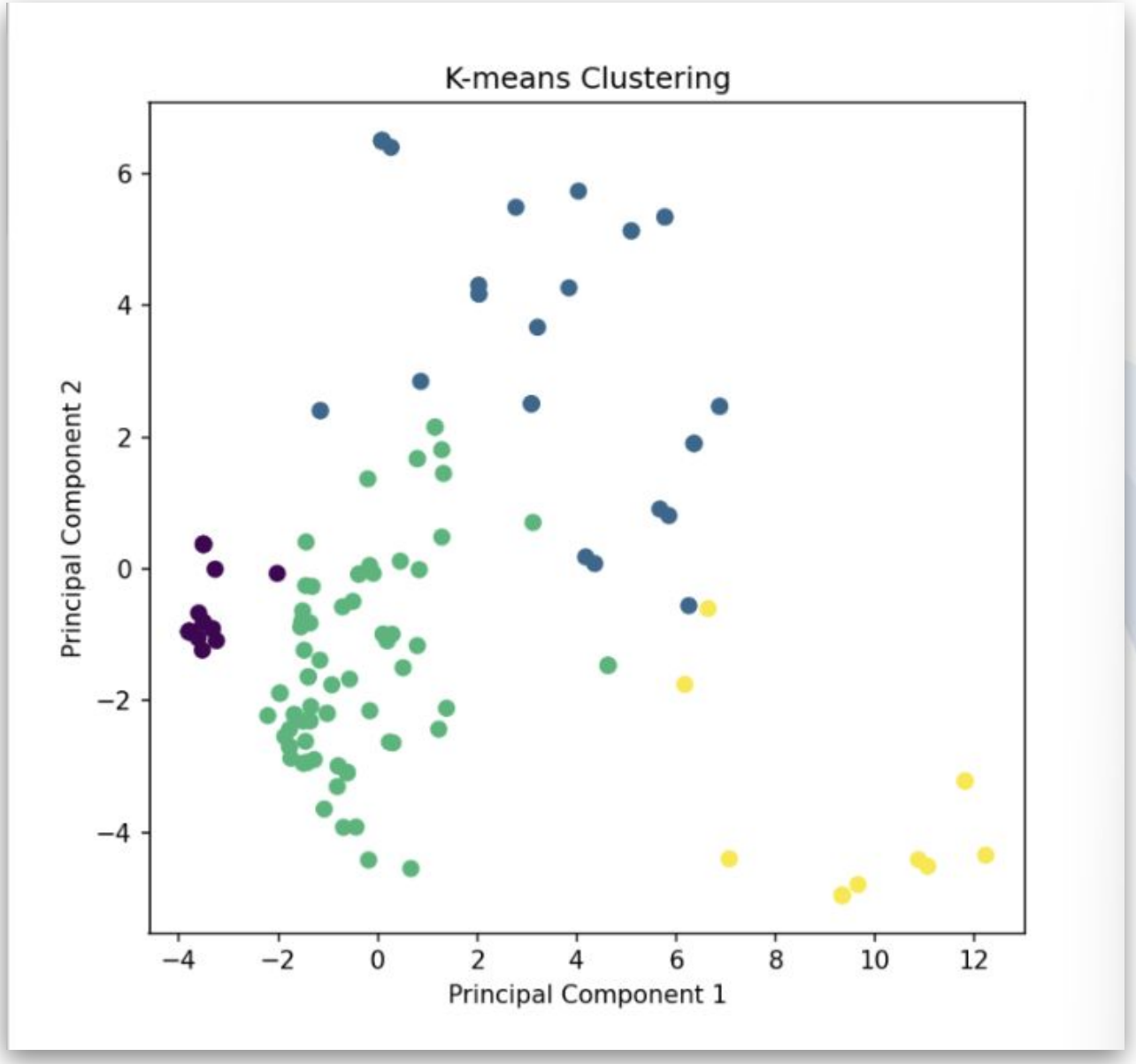
- 提取文本特徵
- 文本轉換為詞袋模型（ Bag-of-Words Model）
- 將文本分割為單詞(或詞組)後，統計每個單詞在文本中出現的次數，並用這些次數構建一個向量。

output > {} all2.json > {} train > [ ] data

```
{
    "train": {
        "data": [

            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1027, T1083, T1129, C0001, C0027, C0031,
            "T1012, T1027, T1082, T1083, T1129, T1222,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1083, T1497, C0002, C0052, C0017",
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1012, T1027, T1059, T1082, T1083, T1113,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1012, T1027, T1033, T1055, T1057, T1082,
            "T1012, T1027, T1082, T1083, T1129, T1222,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1016, T1027, T1082, T1083, T1129, T1543,
            "T1012, T1027, T1033, T1057, T1070, T1083,
```




K-means Clustering
Principal Component 2
Principal Component 1
6
4
2
0
−2
−4
−4
−2
0
2
4
6
8
10
12

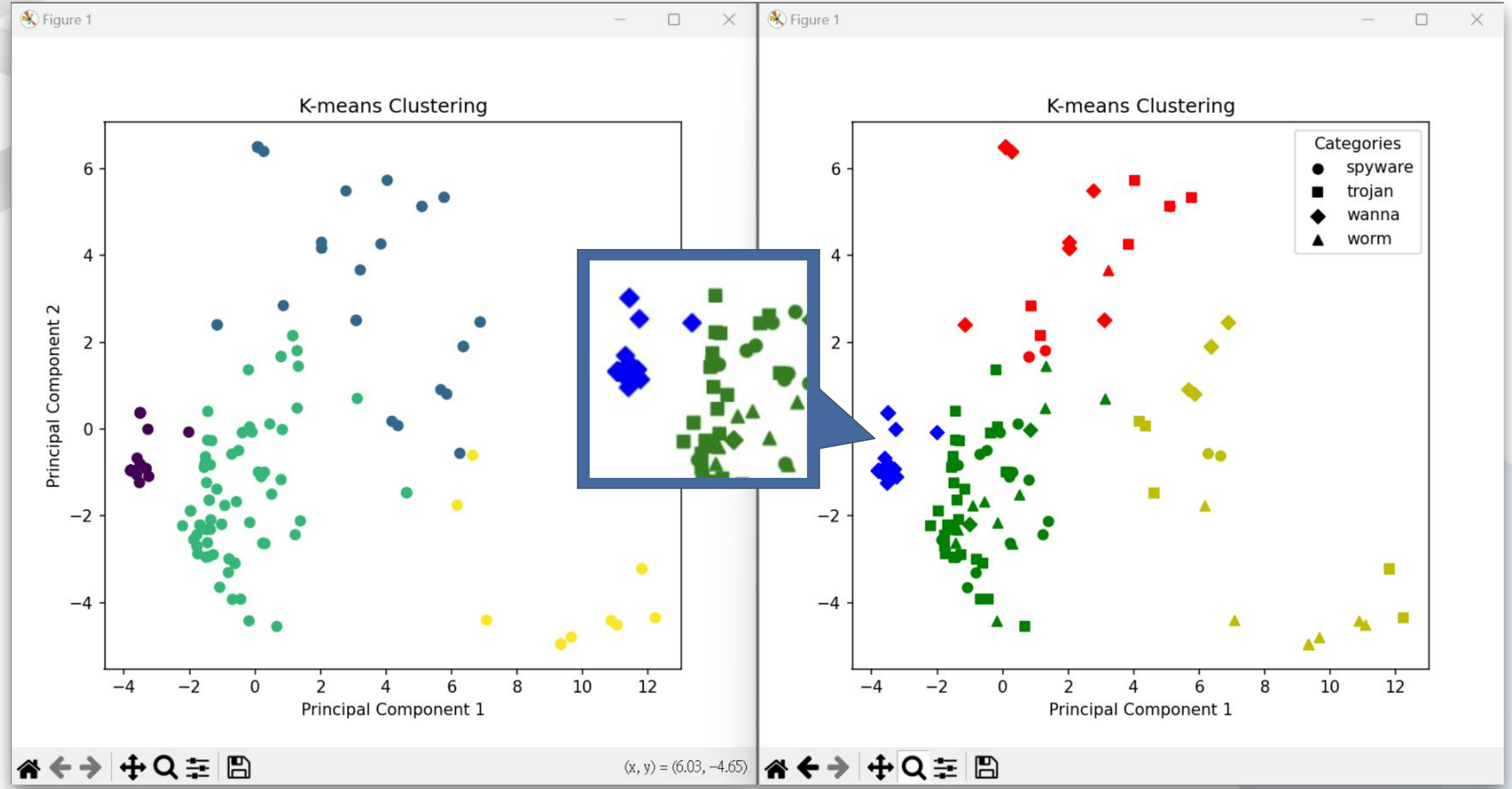


Figure 1
K-means Clustering
Principal Component 2
Principal Component 1
(x, y) = (6.03, −4.65)
Figure 1
K-means Clustering
Categories
spyware
trojan
wanna
worm
Principal Component 1

SentenceTransformer

- 用於句子和段落 Embedding
- 基於BERT及其變體(如 RoBERTa、DistilBERT 等)的預訓練模型
- 生成之Embedding能夠捕捉到句子的語義和語境

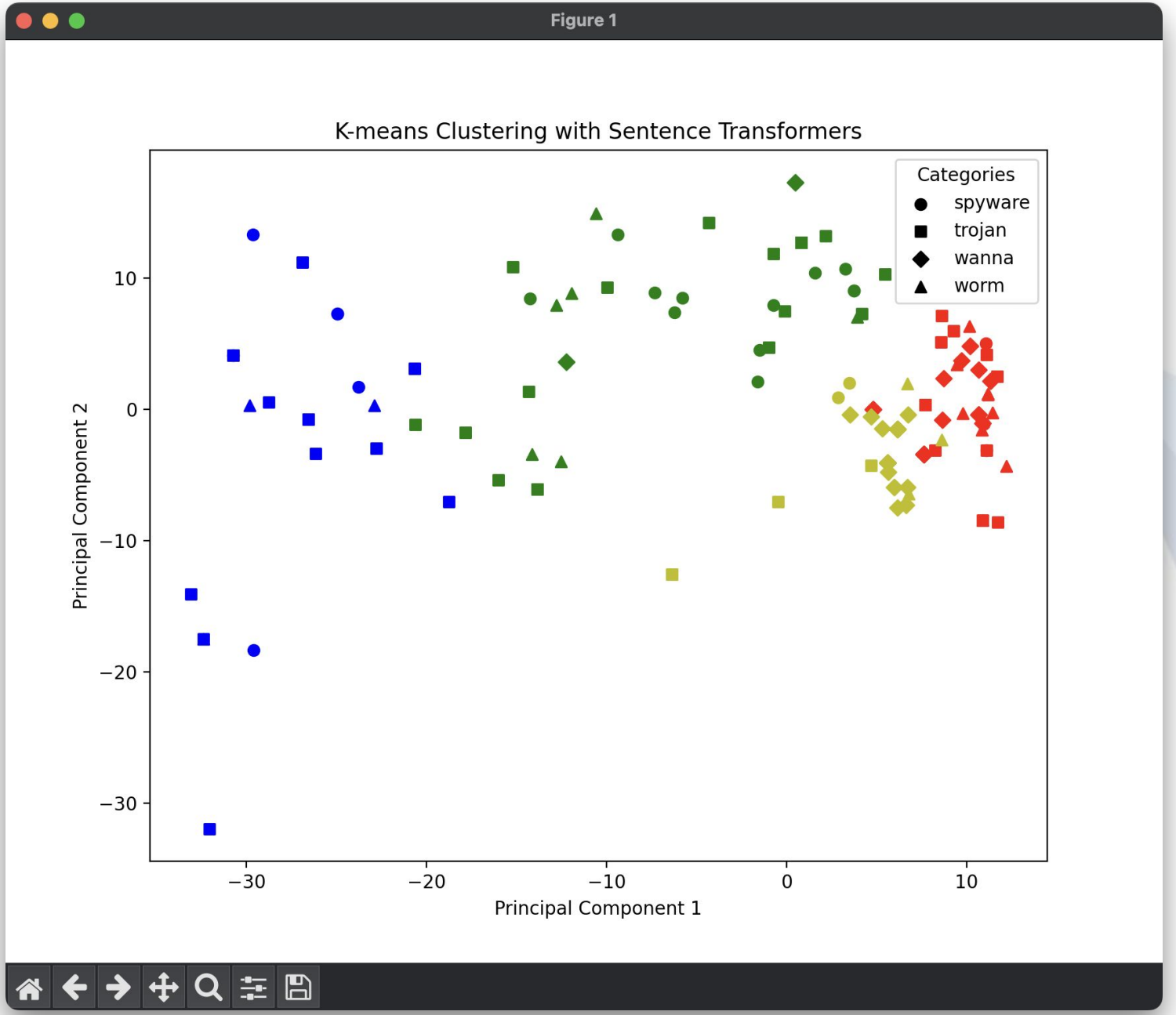


Figure 1
K-means Clustering with Sentence Transformers
Categories
spyware
trojan
wanna
worm
Principal Component 2
Principal Component 1
10
0
−10
−20
−30
−30
−20
−10
0
10

...

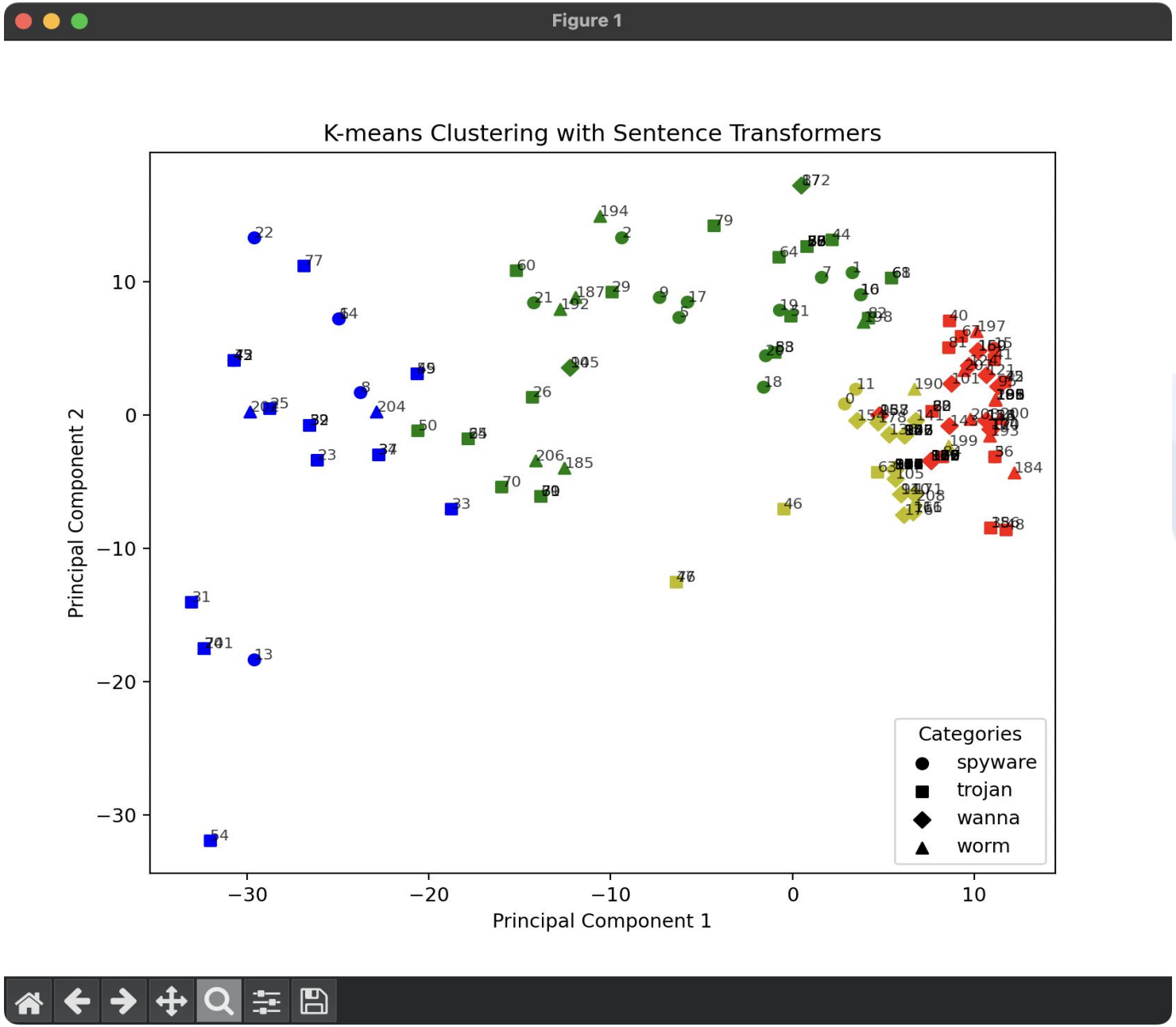


Figure 1
K-means Clustering with Sentence Transformers
Principal Component 2
Principal Component 1
10
0
−10
−20
−30
−30
−20
−10
0
10
Categories
spyware
trojan
wanna
worm

...QQ

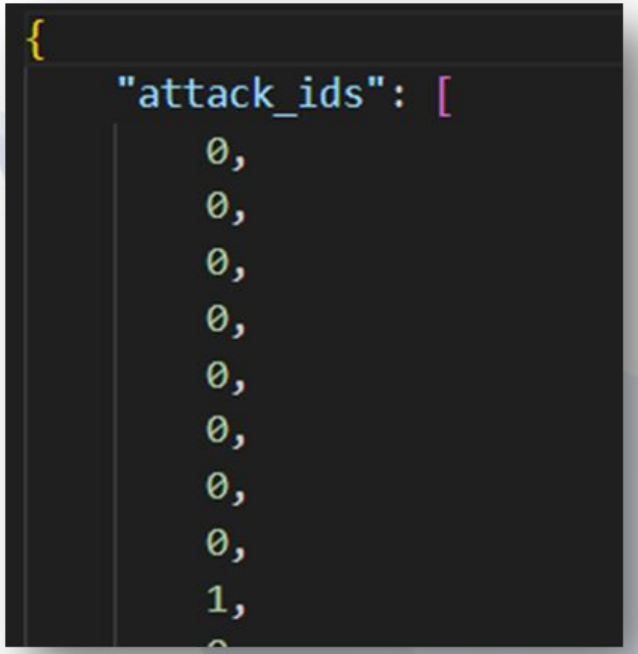
方法三

最終採納的方式

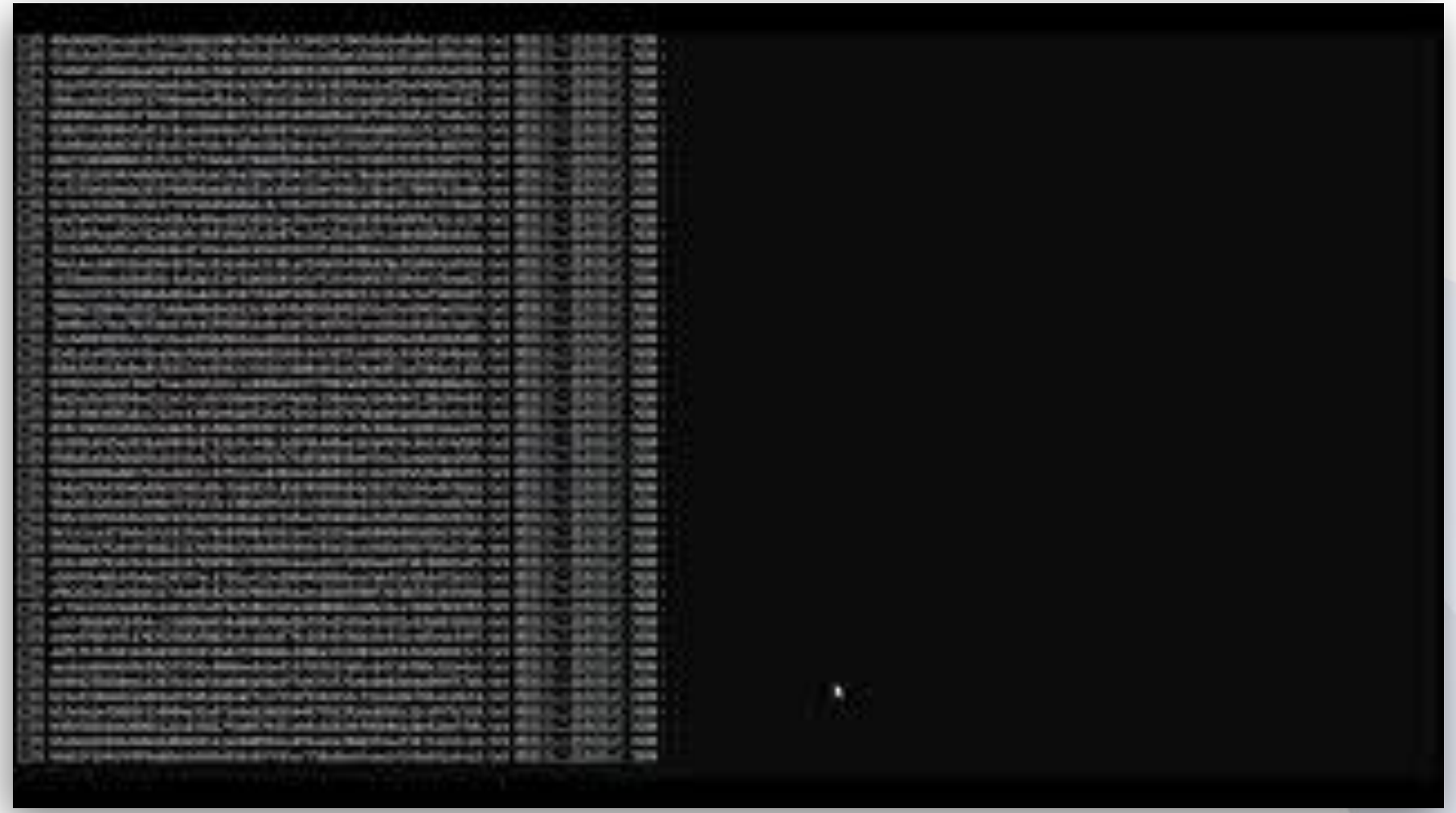
1. 改變資料前處理方式。

2. 使用UMAP將相似度矩陣降維。

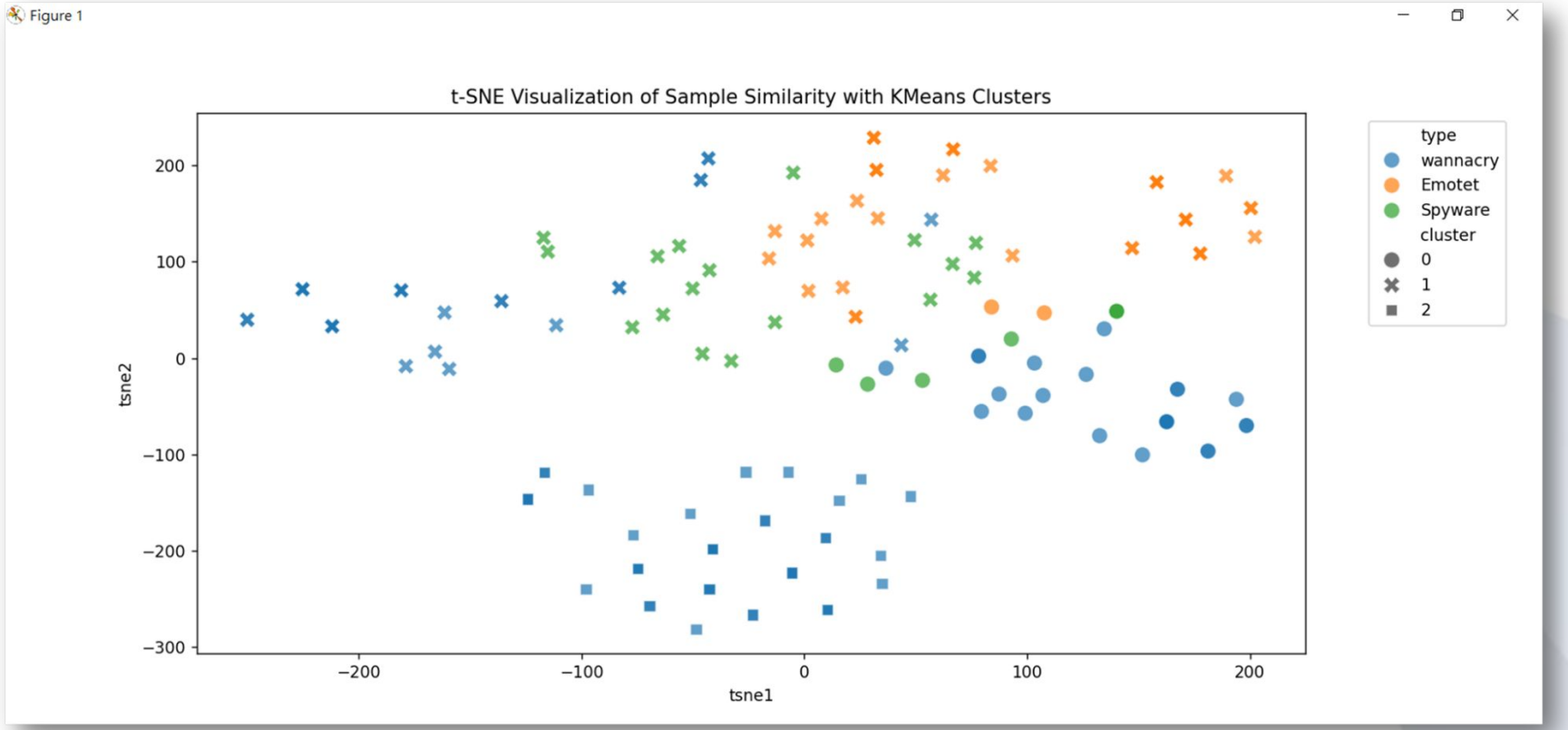
3. 繪製UMAP聚類圖，並標註特定樣本。



提取和分析

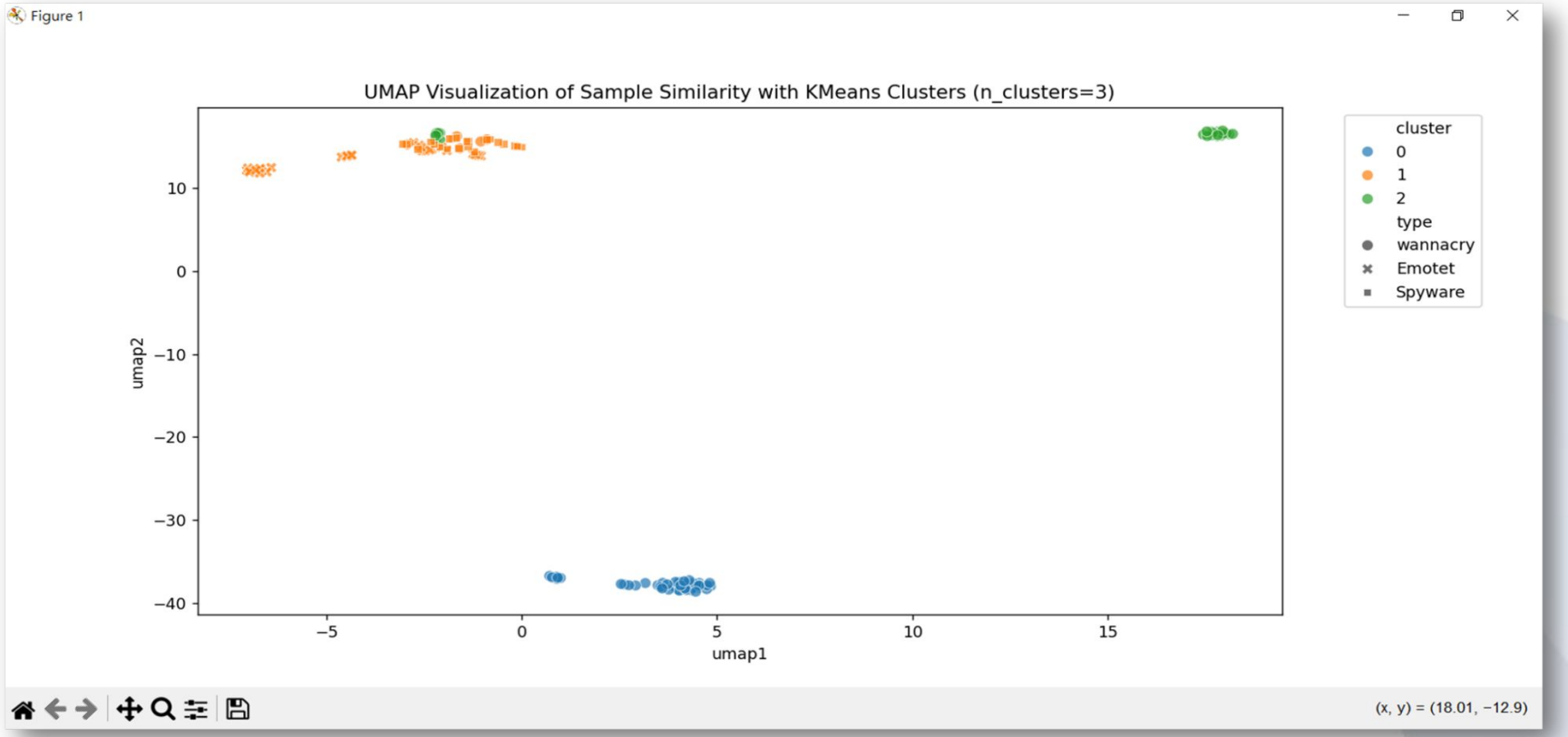


K-means繪圖



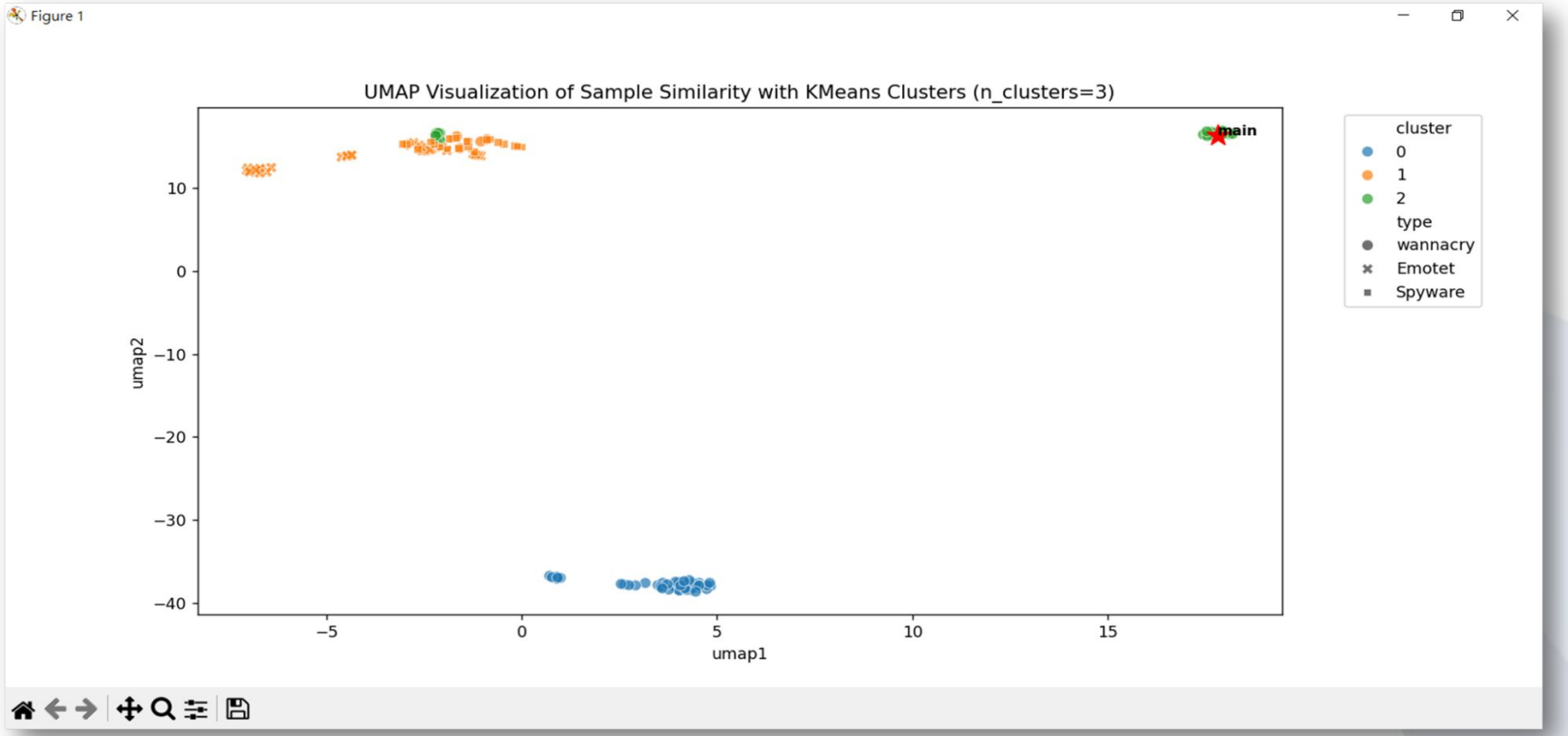
使用t-SNE降維畫出的圖

K-means繪圖



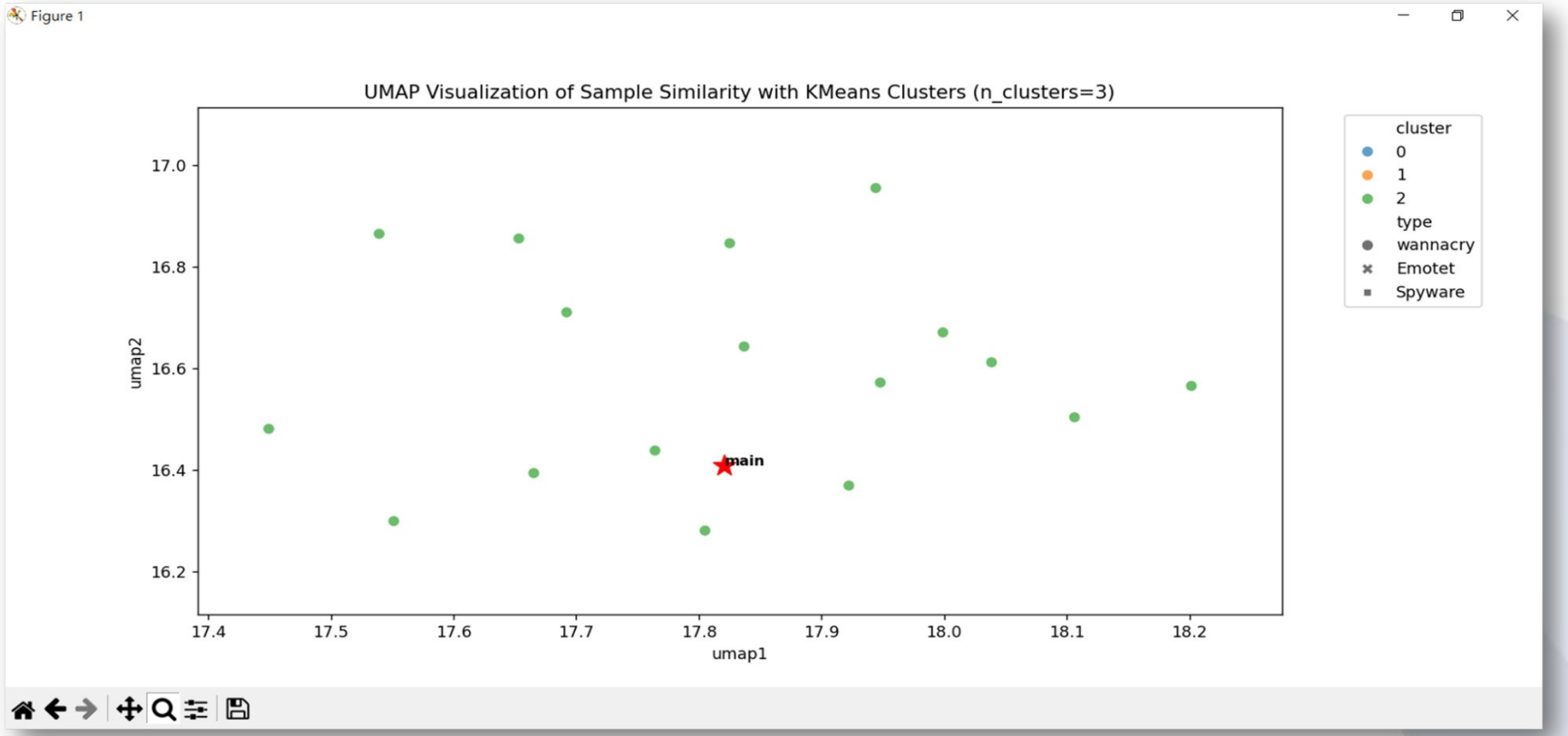
使用UMAP降維畫出的圖

標記



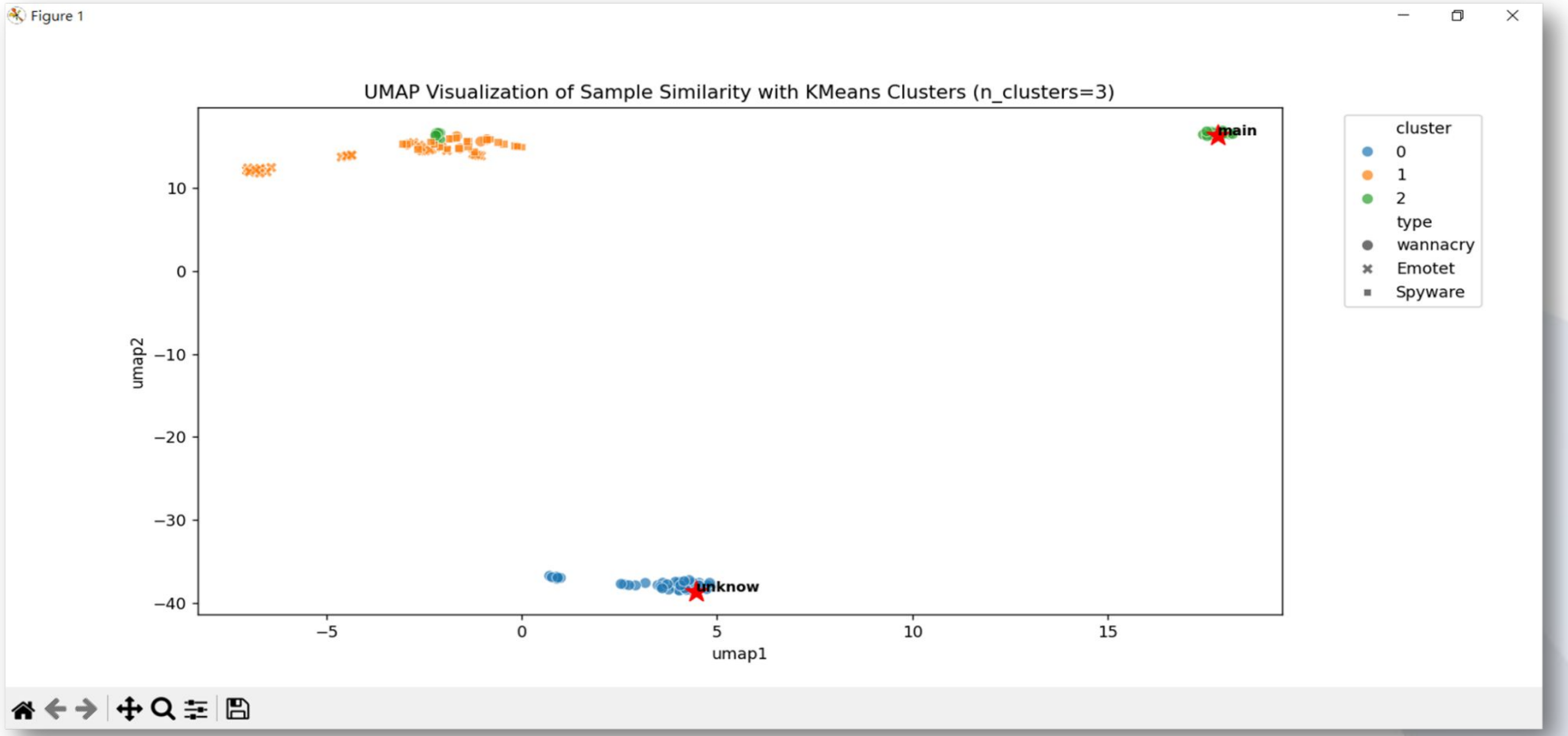
標記Wanncry原珠位置

標記



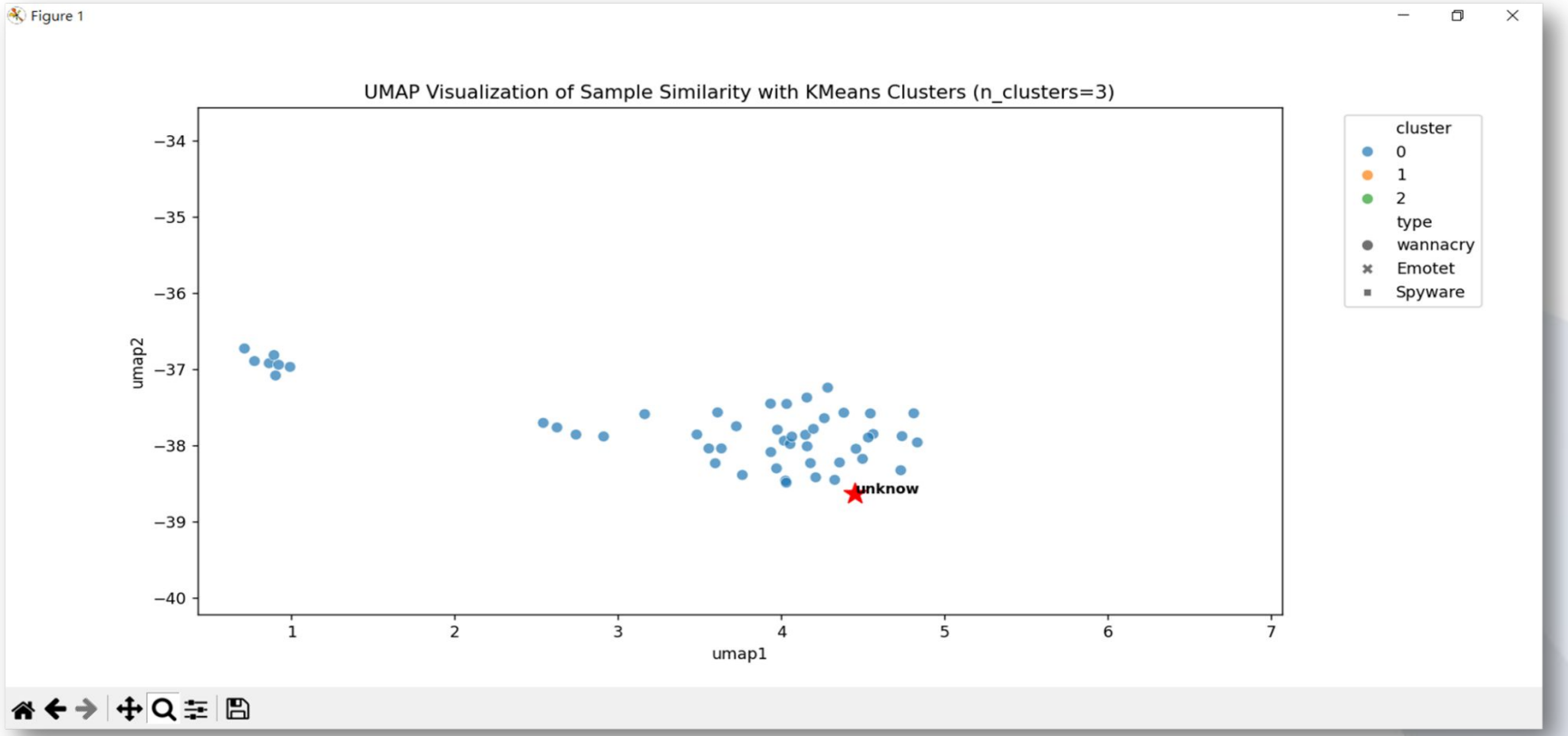
標記Wanncry原珠位置

成果展示



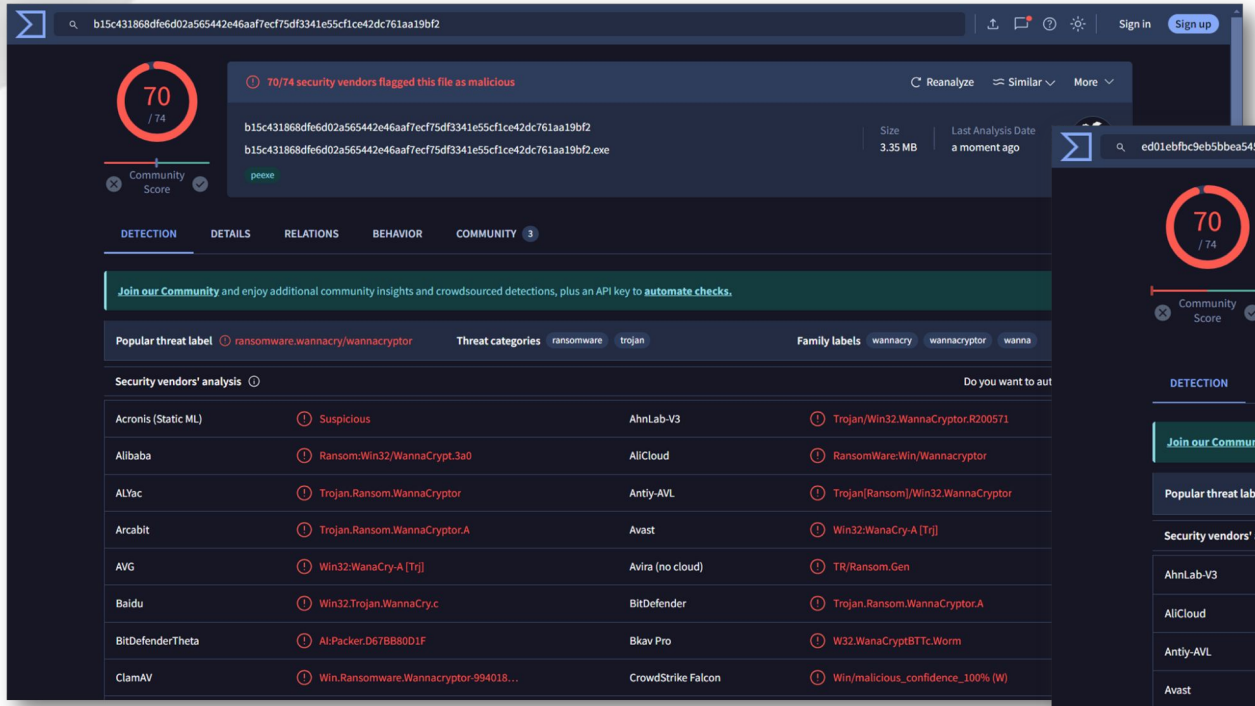
測試放入一筆未知種類的檔案

成果展示

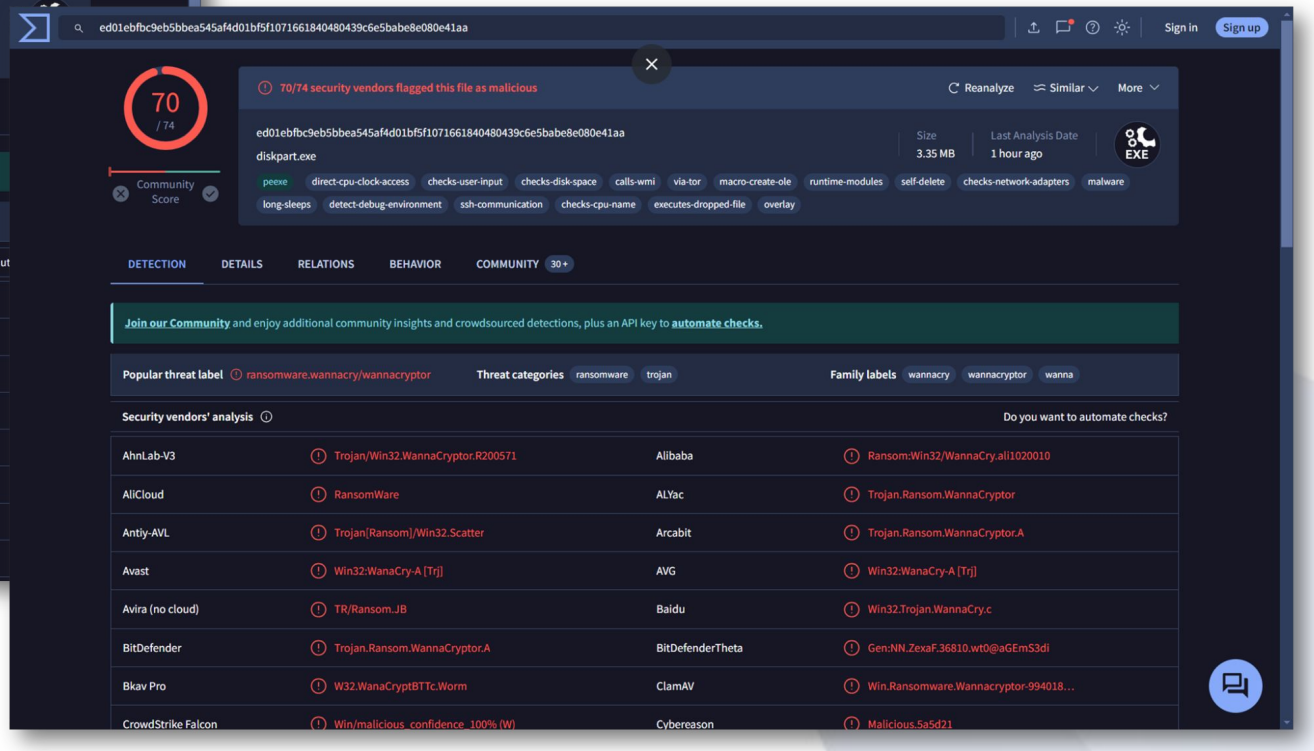


測試放入一筆未知種類的檔案

準嗎？


b15c431868dfe6d02a565442e46aaf7ecf75df3341e55cf1ce42dc761aa19bf2
Sign in
Sign up
70
/ 74
Community Score
70/74 security vendors flagged this file as malicious
Reanalyze
Similar
More
b15c431868dfe6d02a565442e46aaf7ecf75df3341e55cf1ce42dc761aa19bf2
b15c431868dfe6d02a565442e46aaf7ecf75df3341e55cf1ce42dc761aa19bf2.exe
peexe
Size
3.35 MB
Last Analysis Date
a moment ago
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 3
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Popular threat label ransomware.wannacry/wannacryptor
Threat categories ransomware trojan
Family labels wannacry wannacryptor wanna
Security vendors' analysis
Do you want to aut
Acronis (Static ML) Suspicious
AhnLab-V3 Trojan/Win32.WannaCryptor.R200571
Alibaba Ransom:Win32/WannaCrypt.3a0
AliCloud RansomWare:Win/Wannacryptor
ALYac Trojan.Ransom.WannaCryptor
Antiy-AVL Trojan[Ransom]/Win32.WannaCryptor
Arcabit Trojan.Ransom.WannaCryptor.A
Avast Win32:WanaCry-A [Trj]
AVG Win32:WanaCry-A [Trj]
Avira (no cloud) TR/Ransom.Gen
Baidu Win32.Trojan.WannaCry.c
BitDefender Trojan.Ransom.WannaCryptor.A
BitDefenderTheta AI:Packer.D67BB80D1F
Bkav Pro W32.WanaCryptBTTc.Worm
ClamAV Win.Ransomware.Wannacryptor-994018...
CrowdStrike Falcon Win/malicious_confidence_100% (W)

未知檔案


ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
Sign in
Sign up
70
/ 74
Community Score
70/74 security vendors flagged this file as malicious
Reanalyze
Similar
More
ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
diskpart.exe
Size
3.35 MB
Last Analysis Date
1 hour ago
EXE
peexe
direct-cpu-clock-access
checks-user-input
checks-disk-space
calls-wmi
via-tor
macro-create-ole
runtime-modules
self-delete
checks-network-adapters
malware
long-sleeps
detect-debug-environment
ssh-communication
checks-cpu-name
executes-dropped-file
overlay
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY 30+
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Popular threat label ransomware.wannacry/wannacryptor
Threat categories ransomware trojan
Family labels wannacry wannacryptor wanna
Security vendors' analysis
Do you want to automate checks?
AhnLab-V3 Trojan/Win32.WannaCryptor.R200571
Alibaba Ransom:Win32/WannaCry.ali1020010
AliCloud RansomWare
ALYac Trojan.Ransom.WannaCryptor
Antiy-AVL Trojan[Ransom]/Win32.Scatter
Arcabit Trojan.Ransom.WannaCryptor.A
Avast Win32:WanaCry-A [Trj]
AVG Win32:WanaCry-A [Trj]
Avira (no cloud) TR/Ransom..JB
Baidu Win32.Trojan.WannaCry.c
BitDefender Trojan.Ransom.WannaCryptor.A
BitDefenderTheta Gen:NN.ZexaF.36810.wt0@aGEmS3di
Bkav Pro W32.WanaCryptBTTc.Worm
ClamAV Win.Ransomware.Wannacryptor-994018...
CrowdStrike Falcon Win/malicious_confidence_100% (W)
Cybereason Malicious.5a5d21

原珠

結論

透過Capa以及Jaccard
自動分析報告、提取特徵並計算相似度，
有效協助研究人員快速識別和分類
未知病毒近似於何種現有樣本，
以此來加快、簡化惡意軟體分析速度。

參考資料

- CAPA - [連結]
- MITRE ATT&CK® - [連結]
- MBC - [連結]
- MalwareBazza - [連結]
- Jaccard Index介紹 - [連結]

THANKS !

Capa、巴卡阿卡、Jaccard、咪卡巴卡、嗝(๑>ᴗ<)ﻭ

Capa、巴卡阿卡、Jaccard、咪卡巴卡、嗝(๑>ᴗ<)ﻭ

心得分享 - 錯誤問題計畫過程

希望我們的經驗能給你們一些幫助和提示。

以下我們會分享一些

1.做專題過程中，嘗試後卻失敗，從失敗中的經驗，所學的知識！

2.做專題過程中有趣的部分分享！

該先做的事

1. 了解組員
   - 一開始大家要互相了解一下熟悉的領域和知識有哪些
   - 可以從有人的專長開始延伸，這樣成果會比較有深度

2.DC好棒 ->

3.~~跟助教打好關係~~

遇過的問題

- 想不到題目（最可怕的事）
- 想到題目結果都不會做
- 抓惡意程式遇到的障礙
- Capa報告資訊太複雜
- Jaccard index切分詞要如何選擇
- Capa區分不了變體
- AI問題 (萊姆)
- 格式轉換問題＋合併程式碼（萊姆)

想不到題目

我們組是B組中倒數第二晚想出題目的，第二天晚上兩位助教陪著我們到十二點一點。我們前前後後提了二十多種吧。

會遇到的問題(被打槍的種種原因)：

- 有人做過一樣的功能 /專案
- 用Rule-base就可以了
- 太難了 - 助教覺得我們做不出來 /我們自己覺得做不出來
- 要做惡意軟體相關，但沒有人會逆向

~~——不想做AI~~

~~如果大家都跟這領域不熟真的可以直接請助教幫忙生題目~~

抓惡意程式遇到的障礙

手動下在速度太慢，還要一筆一筆拿去分析，乾脆寫個程式自動下載後自動使用 Capa分析。

但是!!

我發現程式也是一筆一筆下載再一筆一筆分析，這樣不知道要跑到民國幾年。

最後使用異步處理、多執行續，加快整個下載分析的過程。

Capa報告資訊太複雜

從Capa所提取出的報告中有很多資訊，一開始認真看了很久 MBC 跟 MITRE ATT&CK框架，想來想去感覺每個類別都有它的重要性很難去做去捨

所以最後就直接全部採納了 ✅

Jaccard index切分詞要如何選擇

一開始用文字且以每個空格作為切分詞來做比較，發現出來了效果不好，分類出來的差距不明顯，但當我們提起出標籤並且改用豆好做為切分詞後效果明顯的升了很多

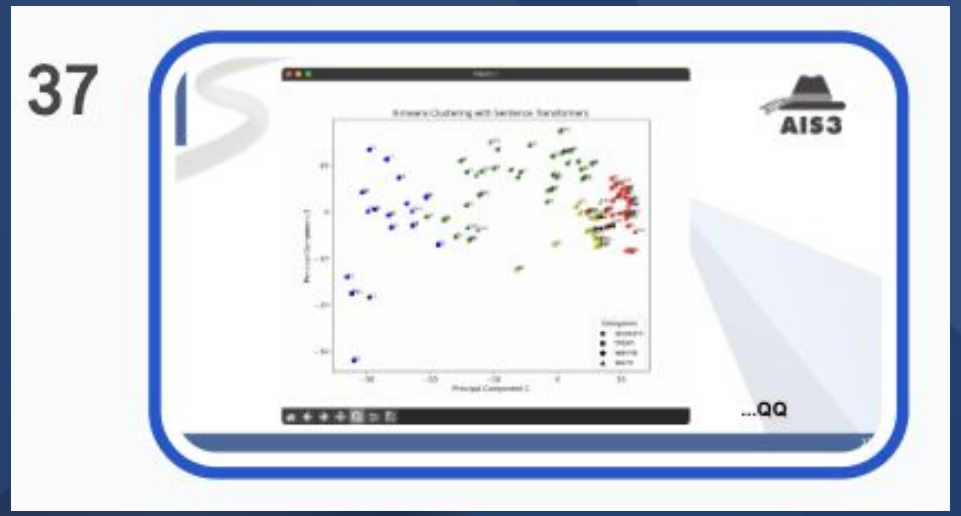
AI - Dataset - 格式轉換問題＋合併程式碼

在要將資料喂進去前要先將資料做好前處理，我們在進行資料轉換的過程中，最終是零零散散的寫了五支程式在跑 (非常難過)。這樣會導致許多問題，像是每次都要改一堆引入文件目錄，還有每個環節都可能出錯卡住。總之後來花了一個多小時，把他們都整合再一起了。

生出capa報告.txt	__genCapa.py(?)
用array[0or1]表示有無之所有Tag	__trans.py(?)
把兩個array併成一個	__check_format.py
a.轉成array[0orTag] .txt	__mergeAll.py
b.轉成array[Tag] .txt	__extract.py
將所有.txt合併成一份json	__toJSON.py

capa.rar ->一堆capa.txt
best.py ->一個json

AI - 跑出來不是我們想要的



就是一個難過的故事。

助教說我們要去調可能要搞一個禮拜，於是就是直接放棄方法二，直接轉成用方法三了。

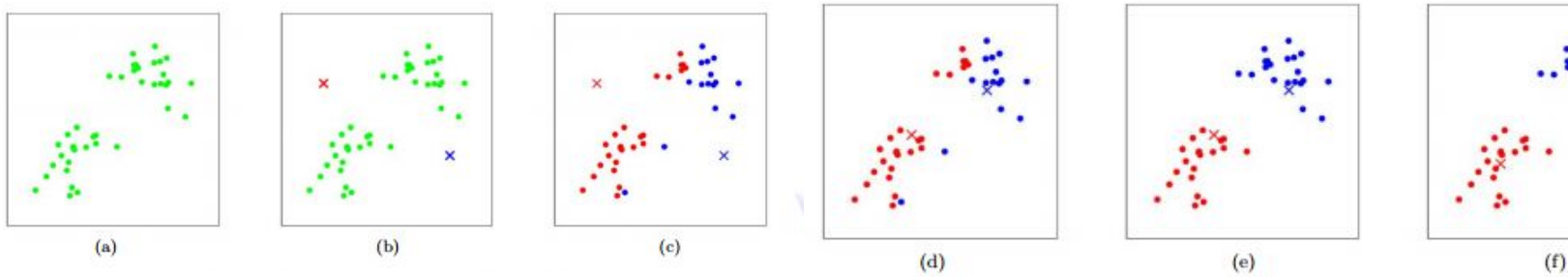
K-means繪圖

一開始用文字且以每個空格作為切分詞來做比較，發現出來了效果不好，分類出來的差距不明顯，但當我們提起出標籤並且改用豆好做為切分詞後效果明顯的升了很多

有趣的部分　（這一part是不是有點諷刺）

- 前面再跑capa腳本分析148個變體時，直接跑了兩節課剛好跑完
- 跟隊友玩
- 一群人塞在一個地方一起爆肝到凌晨連續五天
- 玩了兩次大老二 XD
- 第四天通宵學了一下惡意軟體的編譯　結果最後沒用到 🤡 但其實也沒學會就是了 🤡🤡🤡

Temp



TODO(萊姆)

- 報一遍
- 流程圖

流程圖

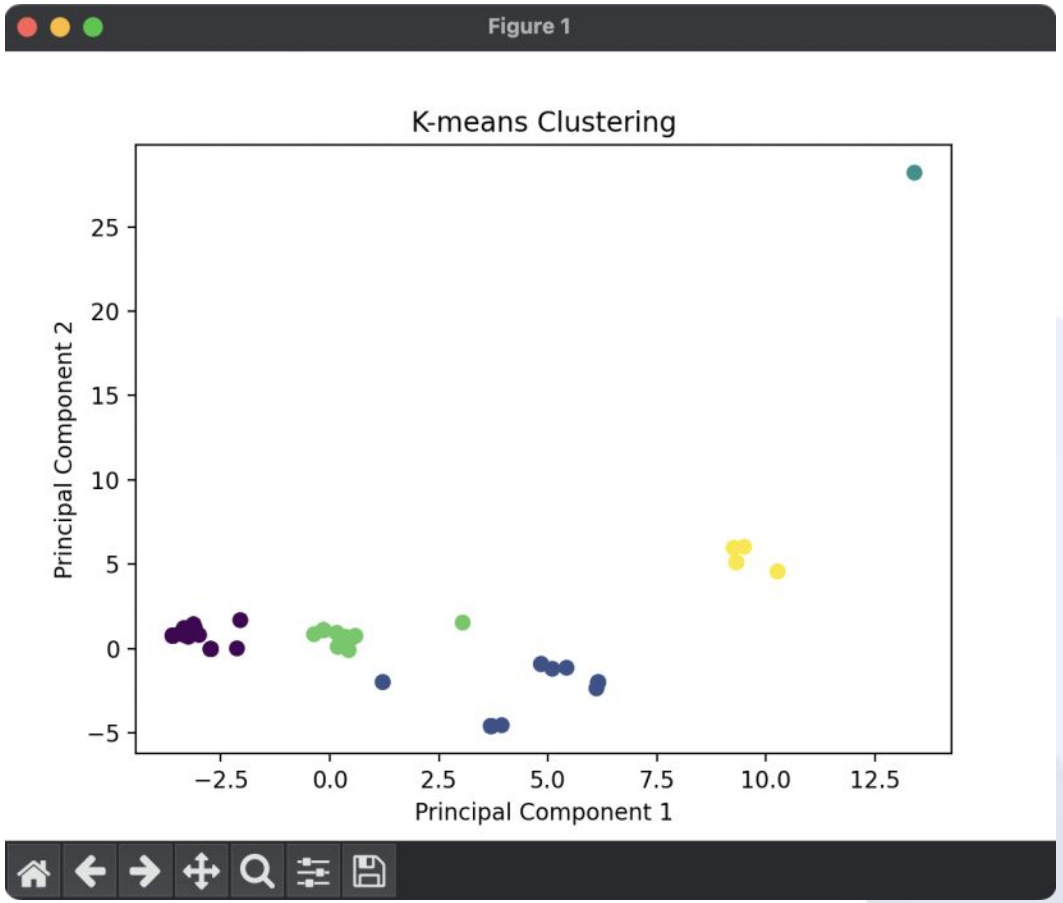
用sample了解我們做了什麼事、一張圖　一張文字

人不要一直換、格式要統一、

展望？

K-means 分群

1.分WannaCry變體



001.txt 以「文字編輯」打開

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0][0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0]
```

002.txt 以「文字編輯」打開

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0][0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0]
```

10th AIS3

目錄