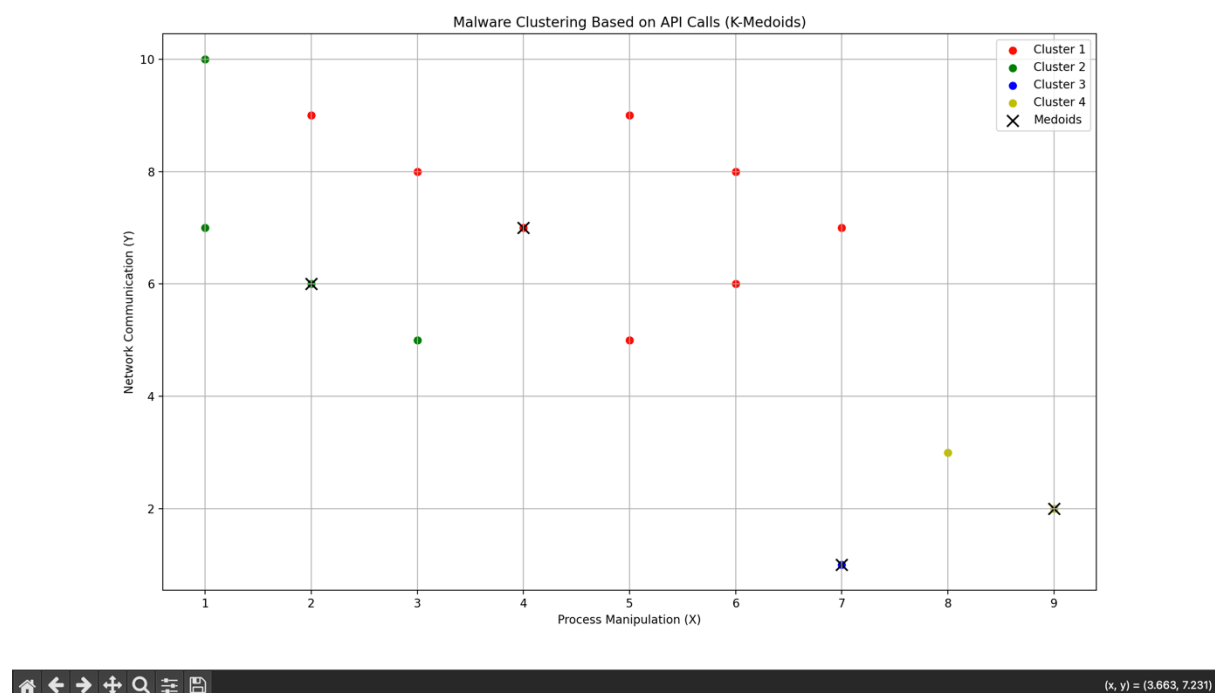
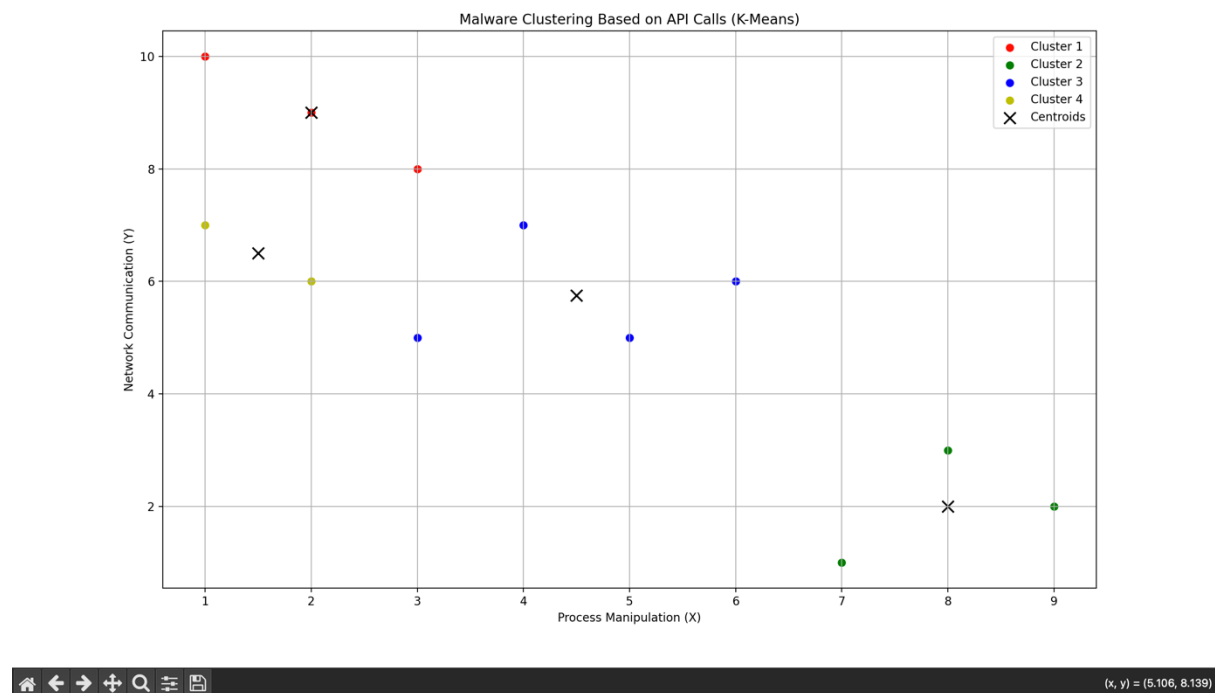


k means is generally faster since calculating means in relatively faster

k medoids is outlier resistant algorithm, meaning if there's a lot of noise it's better to use k medoids rather than k means

k means can only be used for numerical data, while k medoids can be used for both numerical and categorical data.



In my case, both K means and K medoids clustering works fairly well, but as per pictures both algorithms made some mistakes (even though in different parts of graph).

In my case, K means would work better if I made the malware characteristic more numerical, giving all the APIs that they use weight and converting them into numerical values (but since this isn't a major project, there is no need to reverse engineer 15 malware samples and catch all the API calls they make). It would also be a wiser choice if I was working with a large set of different malware, since K means clustering algorithm is more efficient.

K medoids would work better if the set of malware I'm trying to cluster would be non-standard, meaning, if the malware wouldn't follow the average pattern of its family. This would happen because K medoids clustering is resistant to noise so it wouldn't be affected by the malware's irregular API usage.

In my case, I didn't really have a lot of malware samples which use the mix of APIs of different kinds of malware. Most of the samples are pretty well separated, so there is no real need to use K medoids.

K-means clustering algorithm was more efficient in my case with the data given by me.

I am implementing different clustering algorithms in python. I am clustering different malware based on their degree of use of process manipulation(X axis) and networking(Y axis) APIs. please give me the data of some real world malware based on this. scale the degree of usage of the distinct APIs on the scale 1-10,(for example ransomware doesn't do much communication and does a lot of process manipulation, thus example of ransomware coordinates could be [9,2]) give 3 examples of 5 different types of malware: ransomware, backdoor, trojan, spyware, worm.



Here's the data for clustering real-world malware based on their use of Process Manipulation (X-axis) and Networking (Y-axis) APIs:

python

Copy code

```
[
  [9, 2], # M1 (Ransomware)
  [8, 3], # M2 (Ransomware)
  [7, 1], # M3 (Ransomware)
  [2, 9], # M4 (Backdoor)
  [1, 10], # M5 (Backdoor)
  [3, 8], # M6 (Backdoor)
  [6, 6], # M7 (Trojan)
  [5, 5], # M8 (Trojan)
  [4, 7], # M9 (Trojan)
  [3, 5], # M10 (Spyware)
  [2, 6], # M11 (Spyware)
```



Message ChatGPT



ChatGPT can make mistakes. Check important info.

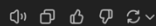
axis) and Networking (Y-axis) APIs:

python

Copy code

```
[
  [9, 2], # M1 (Ransomware)
  [8, 3], # M2 (Ransomware)
  [7, 1], # M3 (Ransomware)
  [2, 9], # M4 (Backdoor)
  [1, 10], # M5 (Backdoor)
  [3, 8], # M6 (Backdoor)
  [6, 6], # M7 (Trojan)
  [5, 5], # M8 (Trojan)
  [4, 7], # M9 (Trojan)
  [3, 5], # M10 (Spyware)
  [2, 6], # M11 (Spyware)
  [1, 7], # M12 (Spyware)
  [6, 8], # M13 (Worm)
  [5, 9], # M14 (Worm)
  [7, 7], # M15 (Worm)
]
```

Each malware type has 3 examples.



Message ChatGPT



ChatGPT can make mistakes. Check important info.