**NOTE: Execution of the files provided below or any files that contain malware must be executed in a controlled and regulated Sandbox environment to prevent the systems and networks from security breaches.**

1. Specific libraries were installed for each to simplify executing the two separate Python scripts. The following libraries have been established for the **Extract\_Features.py** script, which is accountable for gathering network data and extracting features: **pandas** for manipulating information, **pyshark** for network packet analysis, **defaultdict** for managing default dictionary values, **matplotlib.pyplot** for visualisation, and **subprocess** to execute shell commands.
2. The **Dempster\_Shafer.py** script, on the other hand, requires only two libraries: **defaultdict** for essential dictionary functions and **matplotlib.pyplot** for displaying and visualising the outcomes. This script was created to apply the Dempster-Shafer rule for file classification.
3. As the primary IDE for executing specialised scripts developed to run malicious and benign files, Visual Studio Code of **Python version 3.11.4 (64-bit)** was used in this research setting.
4. I created a controlled **sandbox environment** for safe file execution as part of my research on malware analysis focused on enhancing system and network security. While legitimate files were collected from numerous locations, malware samples were obtained from a public repository on GitHub https://github.com/mstfknn/malware-sample-library. The executable files chosen for analysis include both malicious ones (such as **CmdManager.exe, DarkTequila.exe, GoziBankerISFB.exe, botcmd.exe, and OctopusDelphi.exe**) and legitimate ones (such as **AnyDesk.exe, disk-drill-win.exe, Thunderbird Setup 115.1.1.exe, setup-lightshot.exe, and ZoomInstallerFull.exe**).
5. **Extract\_Features.py**
6. Executable files are used as inputs when running the Extract\_Features.py script.
7. Network traffic is captured by Wireshark for 10 seconds for each file.
8. Statistics and collected data are recorded in a spreadsheet at a predetermined path.
9. The result is a graph that displays the bytes transmitted and received between source and destination IPs.
10. Close the graph to continue recording network traffic for the following file.
11. The procedure is repeated for each executable file that is left.
12. **Dempster\_Shafer.py**
13. Assign belief values to the evidence collected from Extract\_Features.py.
14. Calculates belief values, plausibility values, and uncertainty values.
15. Apply the Dempster-Shafer combination rule to the derived values to classify the file, whether malicious or benign.
16. The result is a graph displaying the belief and plausibility values.
17. Close the graphs to check the outputs for belief and plausibility values for benign, belief and plausibility values for malicious, and uncertainty values.
18. The procedure is repeated for each executable file that is left.