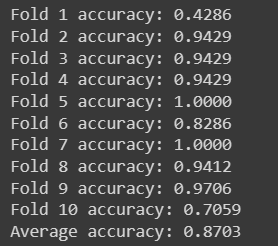
Ransomware Result Analysis

In this research, graph characteristics are extracted from control flow graphs of selected benign and ransomware samples. Machine Learning models were used for classification of files. Below are the Classification results using some renowned classification methods.

Ransomware File: 112

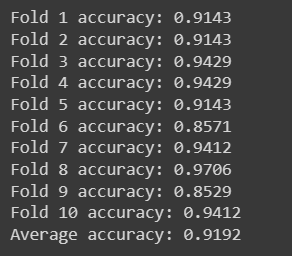
Benign Files: 234

# Random Forest Classifier Using 10 Folds:



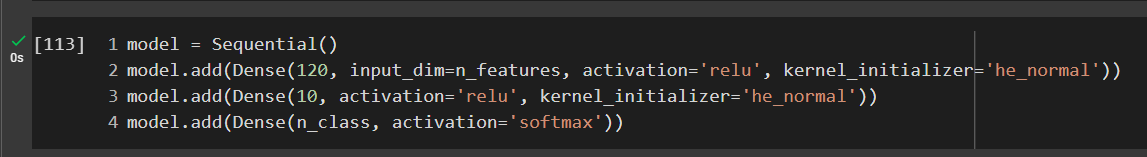
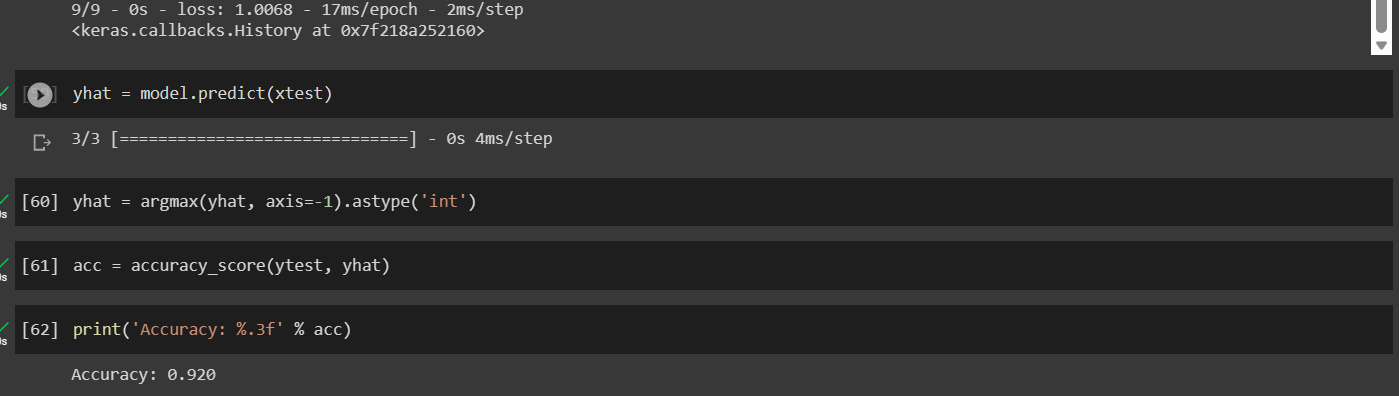
Highest Possible Accuracy Achieved in Radom Forest Classifier is 98.5%, whereas Average is reached around 87%

## SVM Classifier with RBF Kernel for 10 Folds:



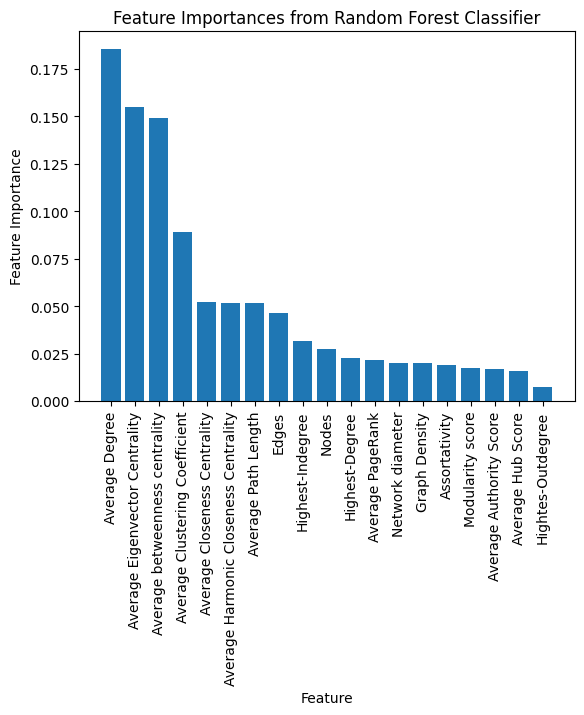
In SVM Classification highest accuracy can be achieved after tuning Kernel as RBF   and regularisation parameter as 1000 and Highest Accuracy Achieved was 92.85% Whereas Average Accuracy was 91.92%

## Neural Network Model with 150 Epochs:

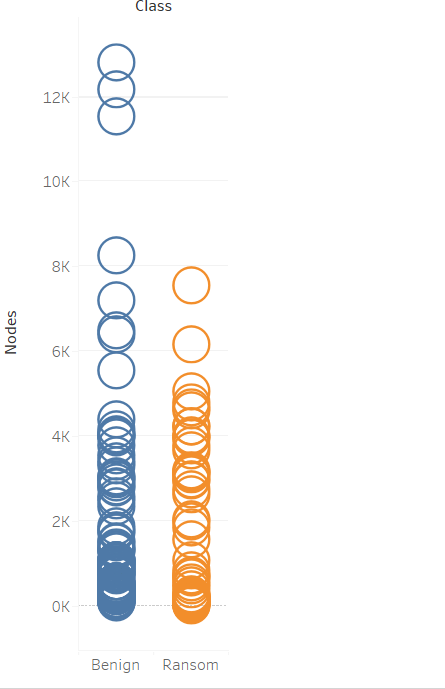
Based on the number of epochs in neural networks the training accuracy and testing accuracy will be changing so according to that we change our model performances by increasing or decreasing epochs (sometimes model may overfit if we increase epochs so, precautions should be taken while fixing number of epochs).

## Important Features Obtained from Random Forest:



Important Features were Extracted from Random Forest Classifier. From the above figure it is clearly stating that Average Degree, Average Eigenvector Centrality, Average Betweenness Centrality, and Average Clustering coefficient are key features to classify a file as Ransomware or Benign. Whereas Highest-Outdegree, Average Hub score, Modularity Score are having less significance in classification.

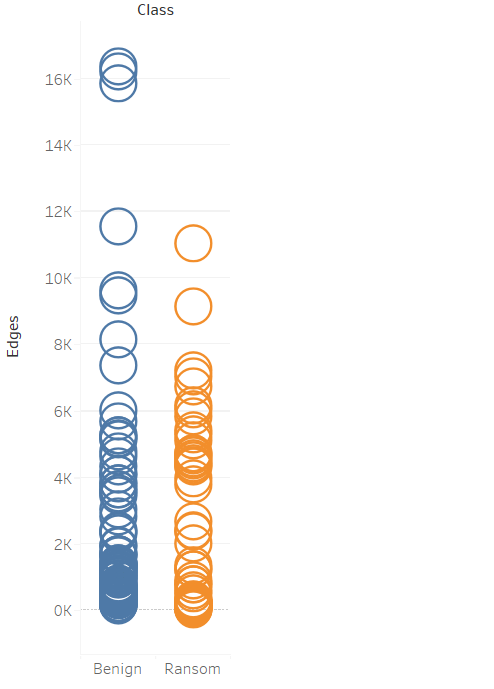
## Nodes:



Ransom: Highest: 7556 Least: 25

Benign: Highest: 12791 Least: 89

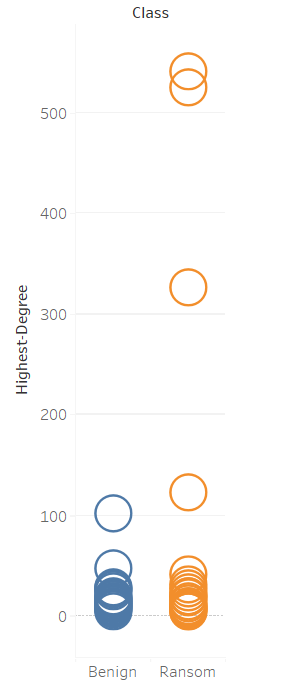
**Edges:**



Ransom: Highest: 11408 Least: 29

Benign: Highest: 16342 Least:127

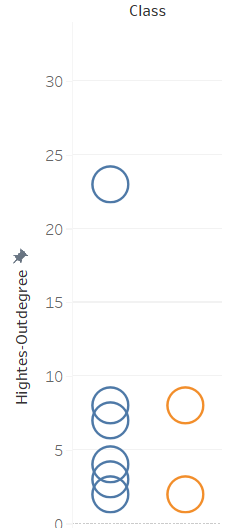
## Highest-Indegree:



Ransom: Highest: 542 Least: 4

Benign: Highest: 101 Least: 5

## Highest-Outdegree

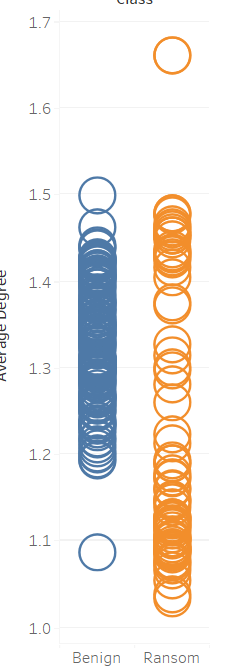


Ransom: Highest: 8 Least: 2

Benign: Highest: 23 Least: 2

## Average Degree:

The average degree of a graph is the average number of edges incident to each node in the graph. It is a measure of the overall connectivity of the graph and can provide insights into the density and structure of the graph.

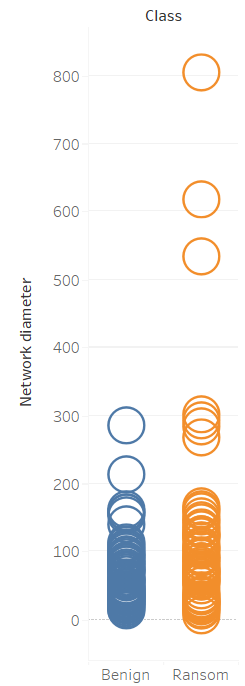


*Ransom: Highest: 1.65 Least:1.03*

Benign: Highest: 1.497 Least: 1.08

## Network Diameter:

The network diameter is the longest shortest path between any two nodes in a graph. In other words, it is the maximum distance between any pair of nodes in the graph.

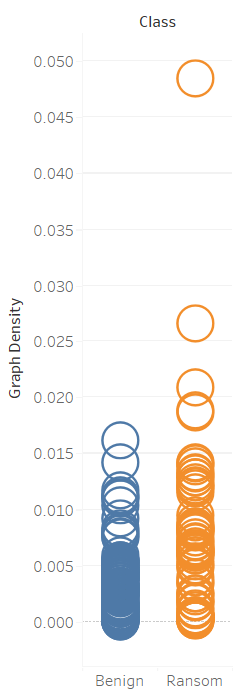


Ransom: Highest: 803 Least: 7

Benign: Highest: 285 Least: 14

# Graph Density:

Graph density is a measure of how many edges are present in a graph compared to the maximum possible number of edges. It is often expressed as a percentage and can range from 0 (no edges) to 1 (all edges).



*Ransom: Highest: 0.04 Least: 0.00019*

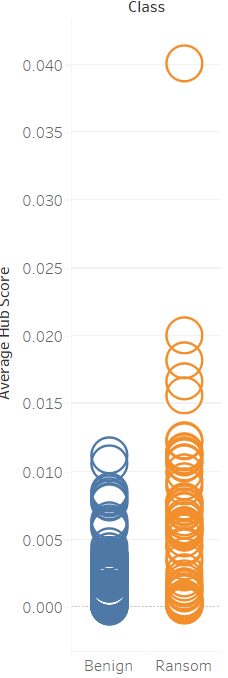
Benign: Highest: 0.016 Least: 0.00010

# HITS (Hyperlink Induced Topic Search)

HITS (Hyperlink-Induced Topic Search) is a link analysis algorithm used in graph analysis to identify important web pages in a network based on their authority and hub scores. In HITS, authority nodes are those that are linked to by many other important nodes, while hub nodes are those that link to many important authority nodes.

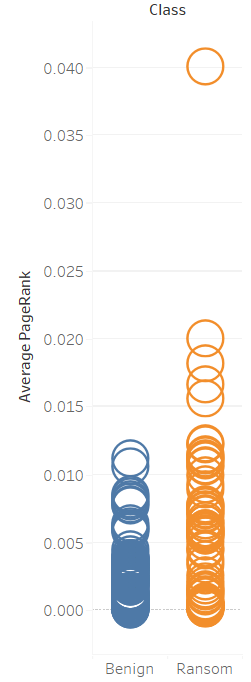
Ransom: Highest: 0.04 Least: 0.00013

Benign: Highest: 0.01124 Least: 0.00008



## PageRank:

PageRank is an algorithm used by Google to rank web pages in its search engine results. It is based on the idea that a web page is important if other important pages link to it. In graph analysis, PageRank is used to measure the importance of a node in a graph based on how many other important nodes link to it.

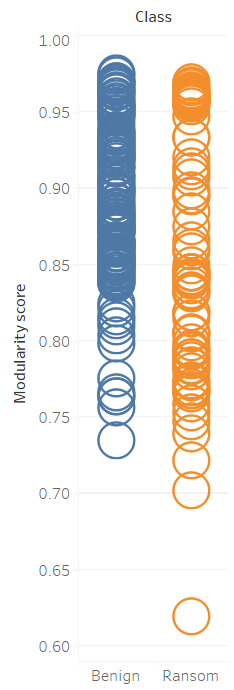


Ransom: Highest: 0.04 Least: 0.00013

Benign: Highest: 0.01124 Least: 0.00008

## Modularity:

Modularity is a measure of how well a graph can be divided into groups or communities. It is based on the idea that nodes within a group are more likely to be connected to each other than to nodes in other groups.

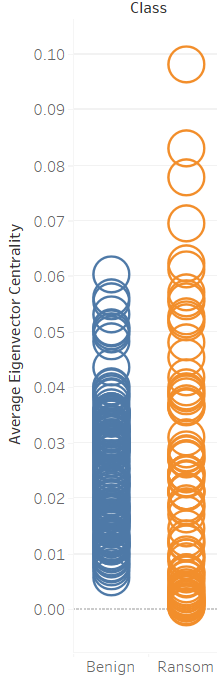


*Ransom: Highest: 0.9690 Least: 0.6195*

Benign: Highest: 0.9748 Least:0.7351

## Eigenvector Centrality:

Eigenvector centrality is a measure of the influence of a node in a graph based on the influence of its neighbors. It is based on the idea that a node is important if it is connected to other important nodes.

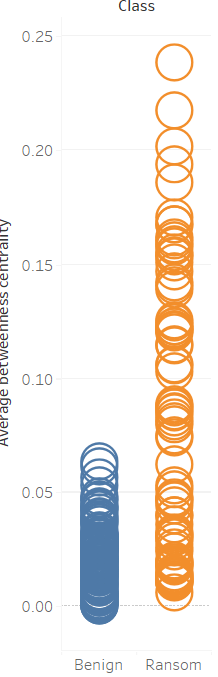


*Ransom: Highest: 0.097 Least: 0.00062*

Benign Highest: 0.0616 Least 0.00585

## Betweenness Centrality

Betweenness centrality is a measure of how often a node lies on the shortest path between other nodes in a graph. It is based on the idea that nodes that are on many shortest paths are important for maintaining communication or flow in the network.



*Ransom: Highest: 0.2381 Least: 0.0061*

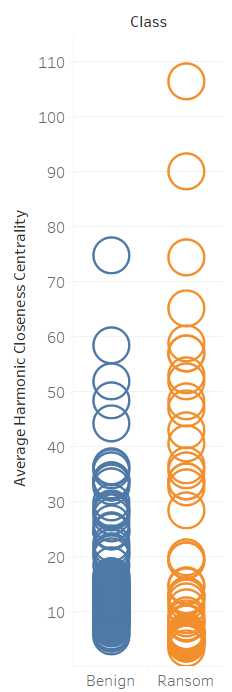
Benign: Highest: 0.0006 Least: 0.0614

## Harmonic Centrality

Harmonic centrality is a variant of closeness centrality that considers the distance between nodes in a graph. It is based on the idea that nodes that are close to many other nodes are more central.

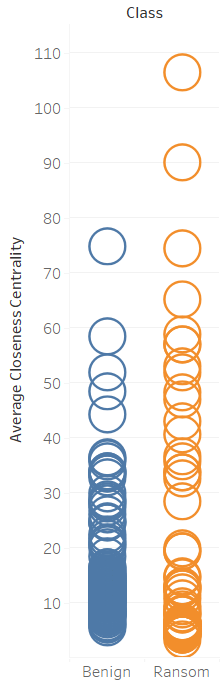
Ransom Highest: 106.6 Least: 3.4

Benign Highest: 5.6 Least: 74.9



## Closeness Centrality

Closeness centrality is a measure of how close a node is to other nodes in a graph. It is based on the idea that nodes that are close to other nodes can spread information or influence more easily.

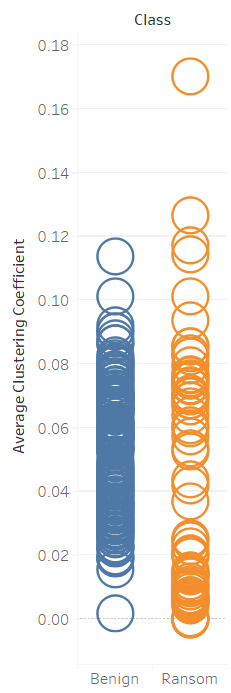


*Ransom: Highest: 106.6 Least: 3.4*

Benign: Highest: 5.6 Least: 74.9

## Clustering Coefficient

Clustering coefficient is a measure of how tightly connected a group of nodes is in a graph. It is based on the idea that nodes that are connected to each other are more likely to be connected to other nodes in the group.

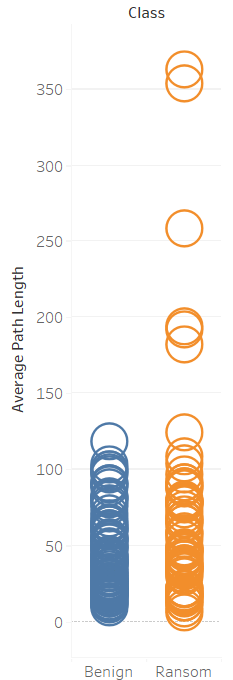


*Ransom Highest:0.1710 Least: 0.00*

Benign Highest: 0.1136 Least: 0.0019

## Average Path Length

The average path length is the average distance between any two nodes in a graph. It is a measure of how easily information or influence can spread through the network.

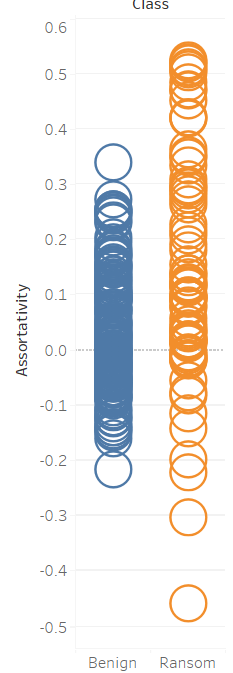


Ransom Highest: 362.4 Least: 6.2

Benign Highest: 118.3 Least: 16.4

## Assortativity

Assortativity is a measure of how nodes in a graph are connected to other nodes with similar characteristics. It is based on the idea that nodes with similar characteristics are more likely to be connected to each other than to nodes with distinctive characteristics.



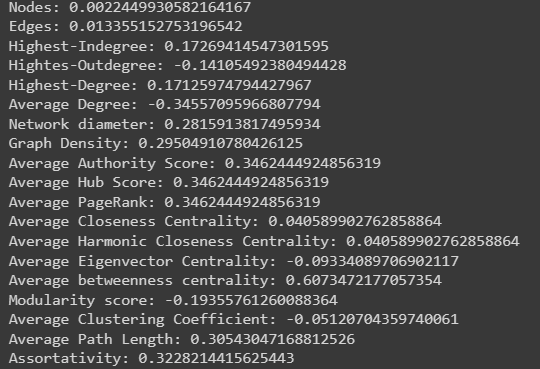
*Ransom Highest: 0.5219 Least: -0.4585*

Benign Highest: 0.3394 Least: -0.2171

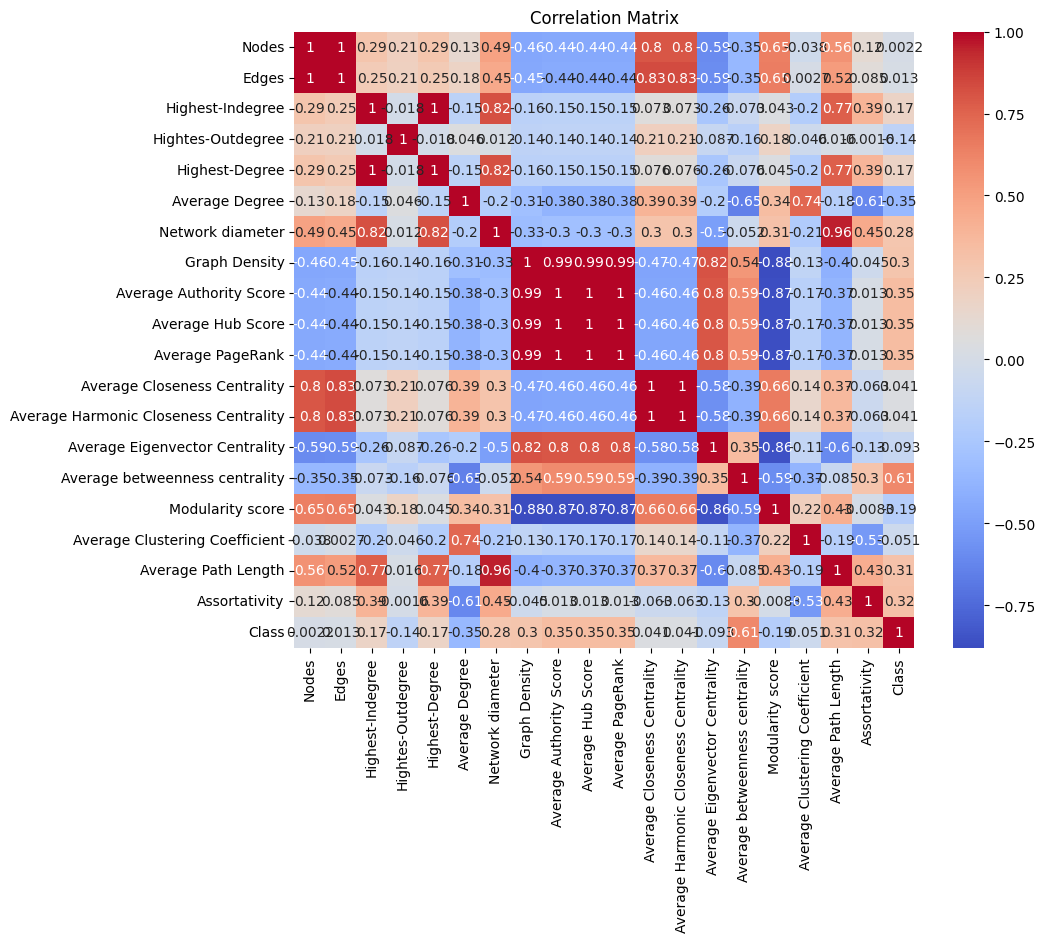
## Correlation Analysis:

Correlation analysis is implemented to figure out if there are any correlations or relationships among the different network features. For example, you can calculate the correlation coefficients (such as Pearson's correlation coefficient) between pairs of features to find any strong positive or negative correlations. This can provide insights into the relationships among the network features and their potential impact on classification accuracy. So, Node, Edges, Highest-Indegree, Highest-Degree, Network diameter, Graph Density, HITS, Average PageRank, Average Closeness Centrality, Average Harmonic Closeness Centrality, Average betweenness centrality, Average betweenness centrality, Assortativity.

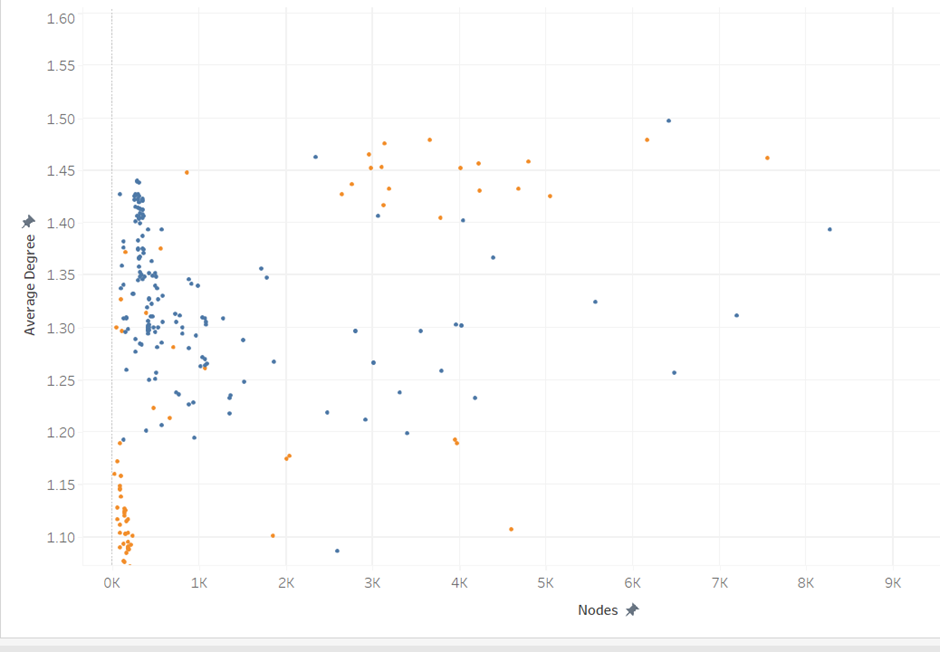
The use of these metrics can yield valuable insights into diverse facets of the network's structure, connectivity, and centrality of nodes. Such insights can facilitate a better understanding of the network's characteristics and behaviour, identification of pivotal nodes or hubs, detection of anomalies or patterns, and even potentially support classification or prediction tasks in the context of your malware classification project. By analysing these metrics in a holistic manner, a comprehensive understanding of the network's properties and interrelationships between various features can be attained, thereby aiding in the analysis of results and interpretation of findings.

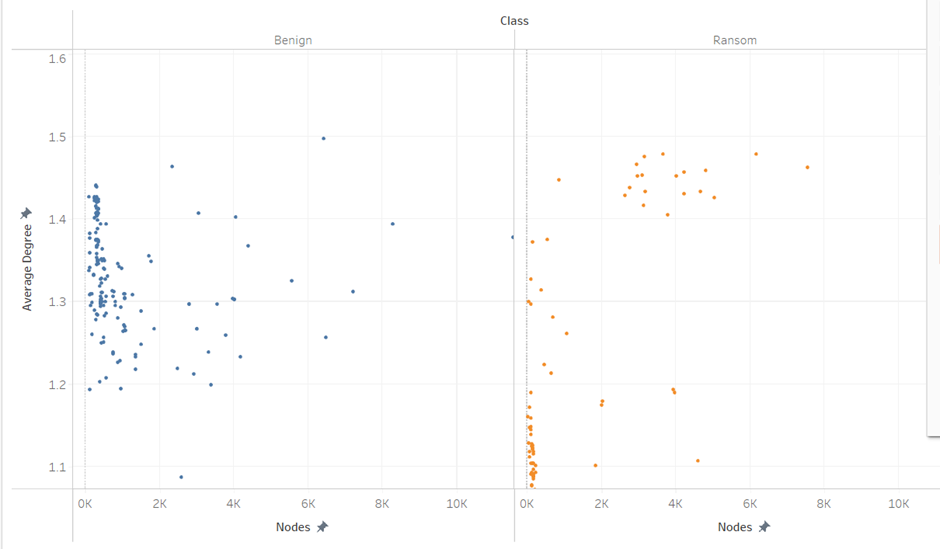


# Correlation Matrix:



**Average Degree vs Nodes**

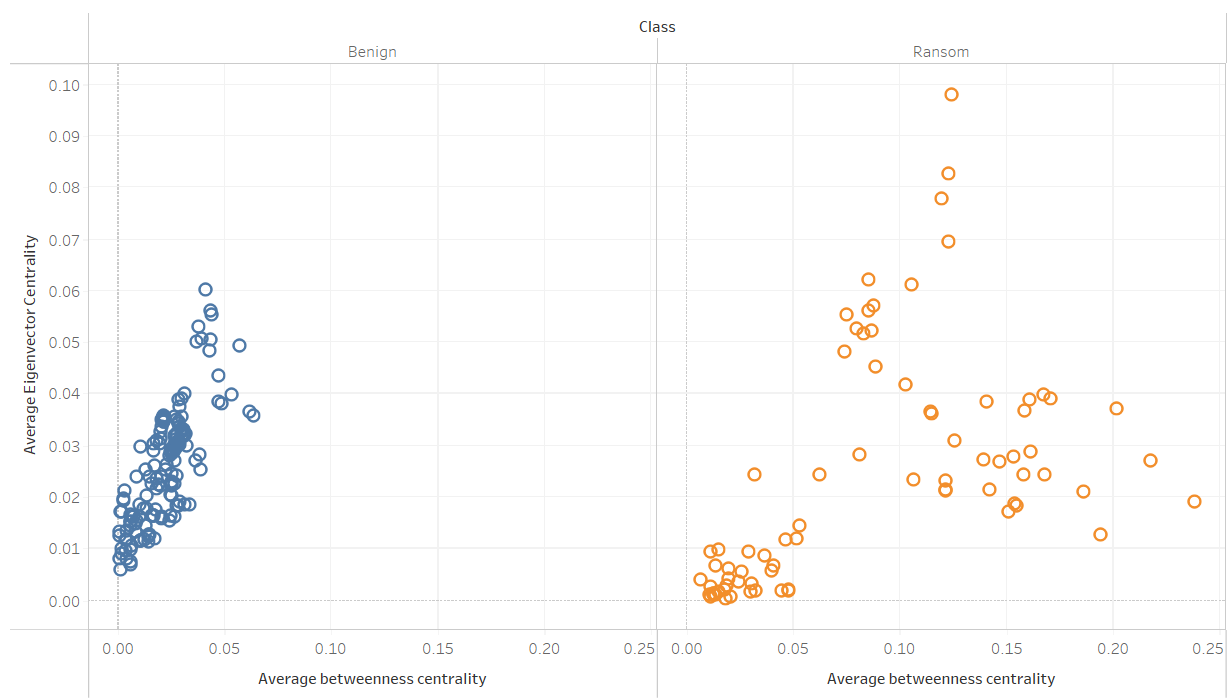




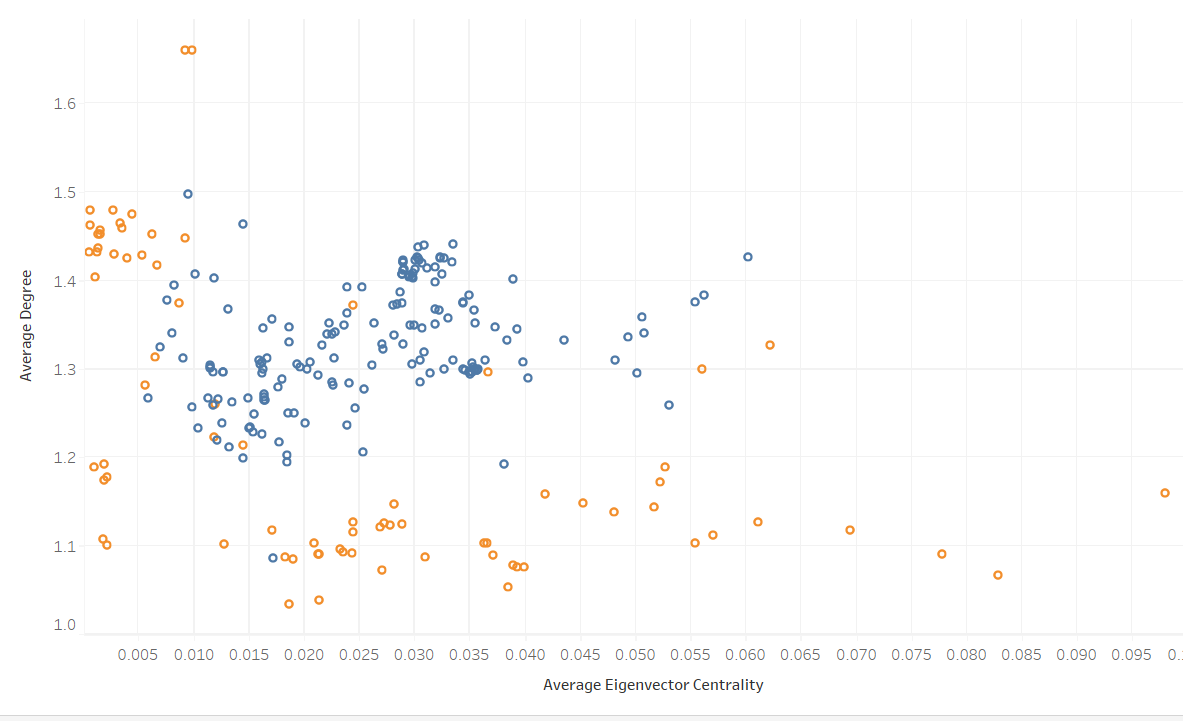
In the above plots of comparing Ransomware and benign, It is clearly observable that in case of benign files, as no of nodes increases average degree remains constant (i.e. almost between 1.25 to 1.45 approx.) where as in Ransomware files average degree is varying on increment of nodes (i.e. almost below 1.25 and above 1.45 approx.)

### Average Eigenvector Centrality Vs Average Betweenness Centrality

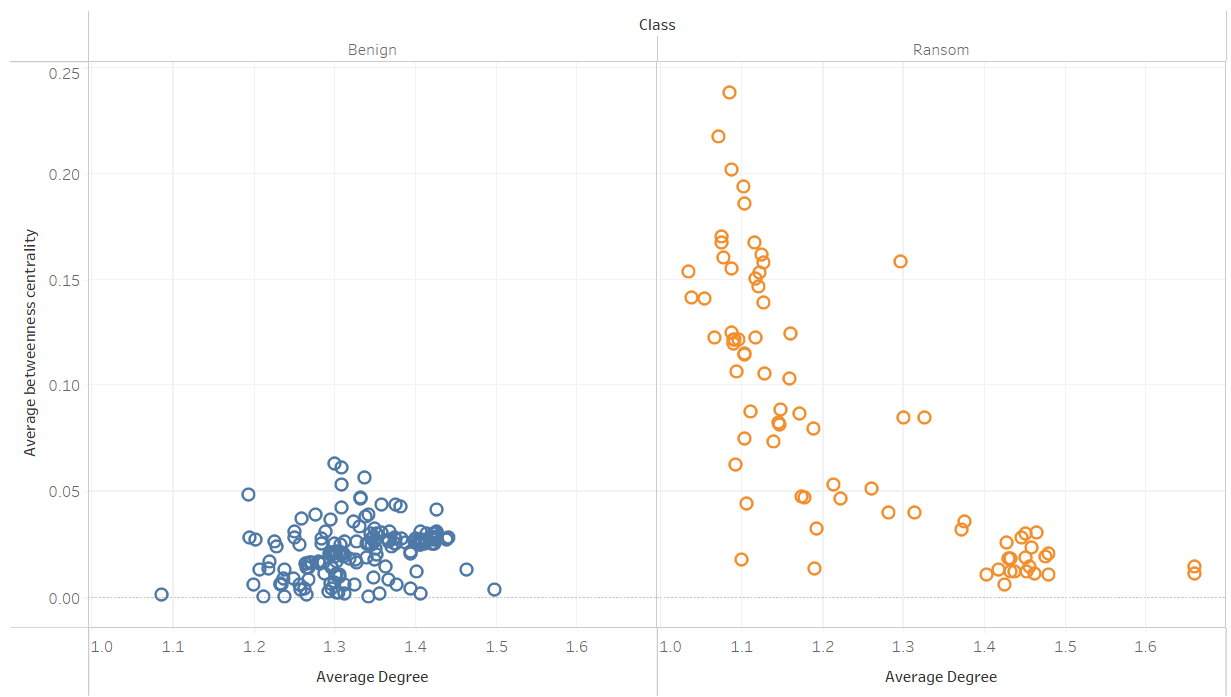
Nodes with high eigen vector centrality tend to be connected to other highly influential nodes in the network, and their removal can have a significant impact on the overall connectivity of the network. On the other hand, nodes with high betweenness centrality tend to lie on many shortest paths between other nodes, and their removal can have a significant impact on the flow of information or resources through the network.



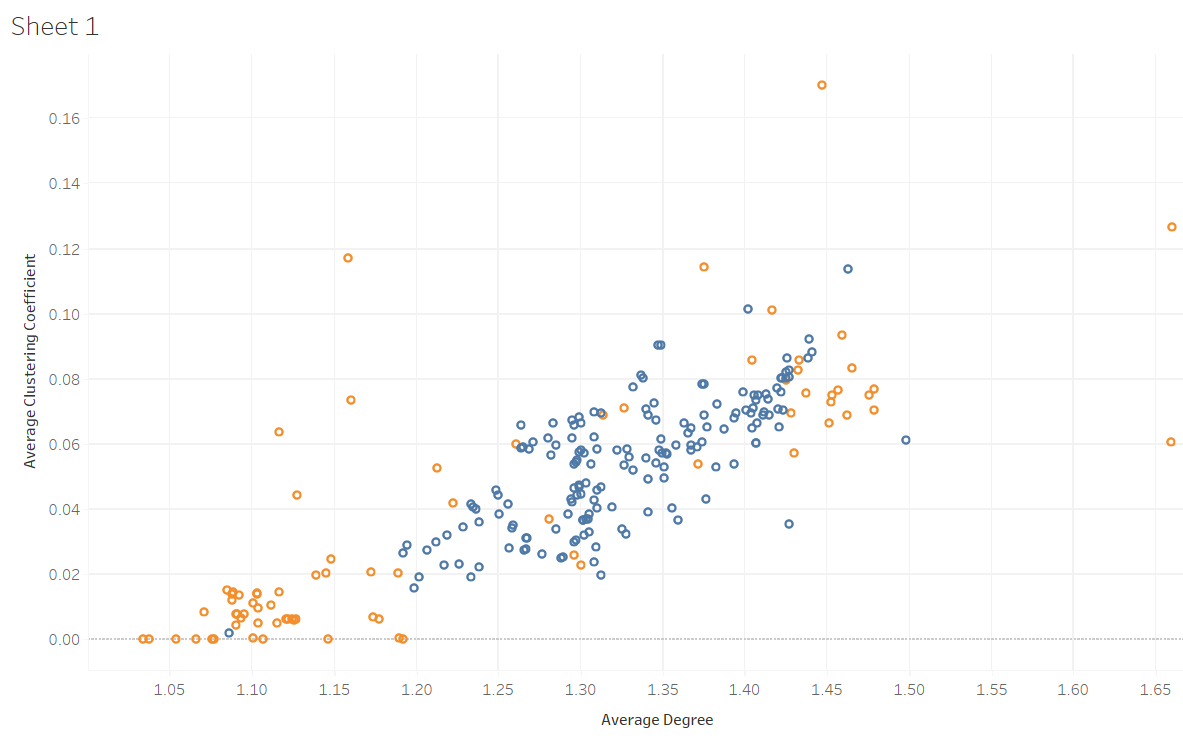
**Average Degree vs Average Eigenvector Centrality**



## Average Degree VS Average Betweenness Centrality



**Average Degree vs Average clustering coefficient**



# Ransomware (API Calls):

* CryptEncrypt
* CryptDecrypt
* CreateFileA/W
* WriteFile
* ReadFile
* DeleteFileA/W
* MoveFileA/W
* GetWindowsDirectoryA/W
* GetSystemDirectoryA/W
* GetTempPathA/W
* RegOpenKeyExA/W
* RegQueryValueExA/W
* RegSetValueExA/W
* InternetOpenA/W
* InternetConnectA/W
* HttpOpenRequestA/W
* HttpSendRequestA/W
* VirtualAlloc
* VirtualProtect
* LoadLibraryA/W
* GetProcAddress

## Behaviour of Ransomware:

Average Degree is an important feature to measure connectivity, so it plays an crucial role in detecting ransomware and its effect. The average degree of the control flow graph can be used to analyse the complexity and connectivity of the ransomware code. A high average degree in the graph indicates that there are many control flow transfers between basic blocks, which can make the graph more difficult to analyse and may increase the risk of vulnerabilities that can be exploited by the ransomware.

By contrast, a low average degree in the graph indicates that there are relatively few control flow transfers between basic blocks, which can make the graph easier to analyse and may decrease the risk of vulnerabilities that can be exploited by the ransomware.

Ransomware encrypts the files of an infected system using system calls. In this scenario average degree is used to detect complexity and depth of file encryption process. As the highest average degree, it performs the typical operations related to system calls like using multiple encryption methods to encrypt. Average degree also infers about spread and propagation of Ransomware. Highest average degree means graph is more connected so propagation through system will be fast, comparatively lower average degree will be for sparse and less connected graph, spread through the system will be slower.

Almost Every Ransomware target on file encryption while some of them also loads Trojans for data exploitation.

**Common Steps :**

* **Executable code extraction.** *Cybercriminals often use binary packers to hinder the malicious code from reverse-engineered by malware analysts. A packer is a tool that compresses, encrypts, and modifies a malicious file’s format. Sometimes packers can be used for legitimate ends, for example, to protect a program against cracking or copying.*
* **Creates RWX memory.** *There is a security trick with memory regions that allows an attacker to fill a buffer with a shellcode and then execute it. Filling a buffer with shellcode is not a big deal, it is just data. The problem arises when the attacker can control the instruction pointer (EIP), usually by corrupting a function’s stack frame using a stack-based buffer overflow, and then changing the flow of execution by assigning this pointer to the address of the shellcode.*

**Ransom:Win32/****Locky.A:**

(0fb1874d13ce559b8df1f40fabe67d1559a653d2dc998dbc558fd917a9382c50)

* **The binary contains encrypted or compressed data.** *In this case, encryption is a way of hiding virus’ code from antiviruses and virus’ analysts.*
* Behaviour consistent with a dropper attempting to download the next stage.;
* Exhibits behaviour characteristic of Locky ransomware;
* **Anomalous binary characteristics.** *This is a way of hiding virus’ code from antiviruses and virus’ analysts.*
* Ciphering the papers located on the victim’s disk drive — so the target can no longer utilize the data;
* **Preventing routine accessibility to the victim’s workstation.** *This is the typical behaviour of a virus called locker. It blocks access to the computer until the victim pays the ransom.*

*API CALLS:*

* AddFontResourceW
* LoadBitmapA
* UnrealizeObject
* GetModuleHandleW
* EndPath
* GetTickCount
* GetParent
* CloseFigure
* CreateCompatibleDC
* GetFileAttributesW
* LoadIconW
* GdiFlush
* EndPath
* SendMessageW
* GetEnhMetaFileW
* CloseFigure
* GetParent
* LoadLibraryA
* RegQueryValueExA
* GetModuleHandleW

The following functions and API calls from the list provided are commonly used by ransomware and trojans to encrypt files and lock systems:

GetFileAttributesW: This function is used to retrieve information about a file or directory, such as its attributes or security settings. Ransomware and trojans often use this function to locate and encrypt specific files on the system.

CreateCompatibleDC: This function is used to create a device context (DC) that is compatible with the specified device. Ransomware and trojans can use this function to create a DC for a bitmap image and then use other functions like LoadBitmapA to load a bitmap that will be used as the ransom note or lock screen.

EndPath and CloseFigure: These functions are used to define and close a path in a graphics device context. Ransomware and trojans may use these functions to draw the lock screen or ransom note on the user's desktop.

LoadLibraryA: This function is used to load a dynamic link library (DLL) into the process address space. Ransomware and trojans can use this function to load a DLL containing encryption or decryption routines.

RegQueryValueExA: This function is used to query the value of a specified registry key. Ransomware and trojans may use this function to obtain information about the system, such as the location of user files or configuration settings.

SendMessageW: This function is used to send a message to a window or control in the user interface. Ransomware and trojans may use this function to display a lock screen or ransom note to the user.

**Ransom:Win32/****Gandcrab.E!MTB:**

(d6976f52f163bf30ce31fd3ff9fb542ec0b15bcae0d9daa09b80c37d828f63e8)

* **Reads data out of its own binary image.** *The trick that allows the malware to read data out of your computer’s memory.*
* Everything you run, type, or click on your computer goes through the memory. This includes passwords, bank account numbers, emails, and other confidential information. With this vulnerability, there is the potential for a malicious program to read that data.
* Performs some HTTP requests;
* **The binary likely contains encrypted or compressed data.** *In this case, encryption is a way of hiding virus’ code from antiviruses and virus’ analysts.*
* Uses Windows utilities for basic functionality;
* **Attempts to repeatedly call a single API many times in order to delay analysis time.** *This significantly complicates the work of the virus analyser. Typical malware tactics!*
* **Installs itself for autorun at Windows startup.** There is simple tactic using the Windows startup folder located at:  
  C:\Users\[user-name]\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup. Shortcut links (.lnk extension) placed in this folder will cause Windows to launch the application each time [user-name] logs into Windows.

#### Ransom:Win32/Crowti.A

(f51ab79090550856f5dff1f62ea90587542fb44cc9945851902318463ecc78dd)

* **Executable code extraction.** *Cybercriminals often use binary packers to hinder the malicious code from reverse-engineered by malware analysts. A packer is a tool that compresses, encrypts, and modifies a malicious file’s format. Sometimes packers can be used for legitimate ends, for example, to protect a program against cracking or copying.*
* Injection (inter-process);
* Injection (Process Hollowing);
* Injection with CreateRemoteThread in a remote process;
* **Creates RWX memory.** *There is a security trick with memory regions that allows an attacker to fill a buffer with a shellcode and then execute it. Filling a buffer with shellcode is not a big deal, it’s just data. The problem arises when the attacker is able to control the instruction pointer (EIP), usually by corrupting a function’s stack frame using a stack-based buffer overflow, and then changing the flow of execution by assigning this pointer to the address of the shellcode.*
* Mimics the system’s user agent string for its own requests;
* Looks up the external IP address;
* Uses Windows utilities for basic functionality;
* Executed a process and injected code into it, probably while unpacking;
* Deletes its original binary from disk;
* Attempts to delete volume shadow copies;
* Attempts to stop active services;
* Modifies boot configuration settings;
* **Installs itself for autorun at Windows startup.**
* There is simple tactic using the Windows startup folder located at:  
  C:\Users\[user-name]\AppData\Roaming\Microsoft\Windows\StartMenu\Programs\Startup Shortcut links (.lnk extension) placed in this folder will cause Windows to launch the application each time [user-name] logs into Windows.
* The registry run keys perform the same action, and can be located in different locations:
  + HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
  + HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run
  + HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
  + HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
* **Creates a hidden or system file.** *The malware adds the hidden attribute to every file and folder on your system, so it appears as if everything has been deleted from your hard drive.*
* **Attempts to modify proxy settings.** *This trick used for inject malware into connection between browser and server;*
* Creates a copy of itself;
* **Attempts to disable System Restore.** [*System Restore*](https://howtofix.guide/system-restore-point-windows-10/) *function – allows you to revert the computer’s state (system files, applications, and system settings) to that of a previous point in time, which can be used to recover after a virus attack.*
* **Anomalous binary characteristics.** *This is a way of hiding virus’ code from antiviruses and virus’ analysts.*
* Uses suspicious command line tools or Windows utilities;
* Ciphering the files situated on the victim’s hard disk drive — so the victim can no more use the data;
* **Preventing regular access to the target’s workstation.** *This is the typical behaviour of a virus called locker. It blocks access to the computer until the victim pays the ransom.*
* Counter({'GetStartupInfoW': 2,
* 'HeapSetInformation': 1,
* 'GetCommandLineA': 1,
* 'lstrlenA': 1,
* 'SetLocaleInfoW': 1,
* 'CreateWindowExW': 1,
* 'FileTimeToDosDateTime': 1,
* 'TlsAlloc': 2,
* 'GetUpdateRect': 1,
* 'DrawEdge': 1,
* 'GetWindowWord': 1,
* 'TransparentBlt': 1,
* 'FillRect': 1,
* 'WaitMessage': 1,
* 'GetTapeParameters': 1,
* 'GdiFlush': 1,
* 'ActivateKeyboardLayout': 1,
* 'GetCPInfo': 3,
* 'GetConsoleFontSize': 1,
* 'GetStringTypeA': 1,
* 'SetWindowContextHelpId': 1,
* 'GetWindowContextHelpId': 1,
* 'FindAtomW': 1,
* 'GetTextAlign': 1,
* 'WindowFromPoint': 1,
* 'str.kernel32.dll': 1,
* 'LoadLibraryW': 2,
* 'SetWindowTextW': 1,
* 'DlgDirSelectExA': 1,
* 'CreateDesktopW': 1,
* 'GetProcAddress': 4,
* 'SetCommBreak': 1,
* 'GetKeyNameTextA': 1,
* 'EnumPropsW': 1,
* 'CreateJobObjectA': 1,
* 'GetTextMetricsA': 1,
* 'SetMetaFileBitsEx': 1,
* 'CreateFontIndirectExW': 1,
* 'VirtualProtect': 1,
* 'GetTextExtentExPointA': 1,
* 'CreateColorSpaceA': 1,
* 'GetModuleFileNameW': 1,
* 'GetStdHandle': 2,
* 'WriteFile': 1,
* 'ExitProcess': 1,
* 'str.mscoree.dll': 1,
* 'GetModuleHandleW': 3,
* 'HeapCreate': 1,
* 'str.KERNEL32.DLL': 2,
* 'TlsSetValue': 2,
* 'TlsGetValue': 2,
* 'TlsFree': 2,
* 'EncodePointer': 7,
* 'DecodePointer': 11,
* 'GetCurrentThreadId': 3,
* 'DeleteCriticalSection': 1,
* 'InitializeCriticalSectionAndSpinCount': 4,
* 'Sleep': 3,
* 'InterlockedIncrement': 4,
* 'EnterCriticalSection': 1,
* 'LeaveCriticalSection': 1,
* 'GetFileType': 2,
* 'SetHandleCount': 1,
* 'GetEnvironmentStringsW': 1,
* 'WideCharToMultiByte': 2,
* 'FreeEnvironmentStringsW': 2,
* 'GetModuleFileNameA': 1,
* 'GetSystemTimeAsFileTime': 1,
* 'GetCurrentProcessId': 1,
* 'GetTickCount': 1,
* 'QueryPerformanceCounter': 1,
* 'GetLastError': 4,
* 'SetLastError': 1,
* 'GetCurrentProcess': 2,
* 'TerminateProcess': 2,
* 'IsDebuggerPresent': 2,
* 'SetUnhandledExceptionFilter': 2,
* 'UnhandledExceptionFilter': 2,
* 'str.USER32.DLL': 1,
* 'InterlockedDecrement': 4,
* 'HeapFree': 1,
* 'RtlUnwind': 2,
* 'HeapSize': 1,
* 'MultiByteToWideChar': 2,
* 'GetStringTypeW': 1,
* 'LCMapStringW': 1,
* 'GetOEMCP': 1,
* 'GetACP': 1,
* 'IsValidCodePage': 1,
* 'HeapAlloc': 2,
* 'HeapReAlloc': 1,
* 'ReplyMessage': 1})

Thread Local Storage (TLS) is a mechanism that allows each thread of a process to have its own data, distinct from data belonging to other threads. TLS functions, like TlsAlloc, TlsGetValue, and TlsSetValue, are used to allocate and access the TLS memory for a specific thread.

Ransomware and trojan can use TLS to store information specific to a thread, like encryption keys or data related to the malware's behavior. This information can be kept separate from the rest of the process, making it harder to detect and analyze by security software.

* HeapSetInformation: Ransomware and Trojans can use this API to modify the heap settings, which can cause the system to allocate more memory than it is capable of handling. This can lead to a system crash or make the system unresponsive, allowing the malware to perform its malicious activities without being detected by antivirus software.
* CreateJobObjectA: This API can be used by malware to create a new job object, which can be used to control and limit the resources available to processes running on the system. By limiting the resources, the malware can slow down the system and prevent other processes from running, making it difficult for the user to detect and remove the malware.
* TlsAlloc: This API can be used by malware to allocate memory for Thread Local Storage (TLS). TLS is a mechanism that allows each thread of a process to have its own private storage space. Malware can use this API to allocate a large amount of memory for TLS, which can cause the system to slow down or crash.
* GetCommandLineA: This API can be used by ransomware to determine the command line arguments used to launch a process. This can be useful for identifying important files and directories on the system that can be targeted for encryption.
* SetLocaleInfoW: This API can be used by ransomware to change the locale settings on the system. By doing so, the malware can change the language used by the system and make it difficult for the user to understand the ransom demands or instructions provided by the malware.
* CreateWindowExW: This API can be used by ransomware to create a new window on the system. The window can be used to display a ransom message or other instructions to the user.

Ransomware may use the EncodePointer and DecodePointer API calls to help obfuscate their code and evade detection. These functions can be used to transform a pointer into a new pointer value that is not immediately recognizable by static analysis tools. This can make it more difficult for security software to detect the presence of the malware and to identify its behavior. The DecodePointer function is used to reverse the transformation and retrieve the original pointer value when needed.  
  
**Average Betweenness centrality:**

Average Betweenness centrality is number of shortest paths between all pairs of nodes in a graph that pass through a particular node so it can be used in ransomware to find the important parts in a graph and in the ransomware code that are responsible for spread of ransomware through the network further more and may affect additional systems in the network.

If betweenness centrality is high then number of edges that pass through these nodes will be high which indicates number of calls through a function. If it is low then ransomware effect on the system will be less.

So, by finding these important nodes that have high betweenness helps to develop mitigation strategies that may help to reduce spread of the ransomware and its effect.

**Average closeness centrality:**

Closeness centrality states how the nodes are closely interconnected in the graph, So nodes with high closeness centrality potentially suggest that they play a crucial role in functioning of ransomware and the removal of that nodes could halt the entire program. This could be due to the presence of multiple decision points or loops in the code, which allows the ransomware to take different execution paths based on the system environment or user behaviour

On the other side less closeness centrality will decrease the pace of ransomware spreading. This could be due to the absence of decision points or loops in the code, which limits the number of possible execution paths

**Average clustering coefficient:**

High average clustering coefficient in ransomware says there the clusters that are tightly grouped together will be very high which indicates that some code blocks are highly interconnected.

So, in these clusters each one represents a particular functionality of the ransomware program for instance file encryption (how the file should be encrypted and keys that are used to encrypt the system data) and the commands how it can be pass through network from system to system.

If number of clusters is high then ransomware program is very hard to remove from system and uses higher level of encryption standards and in the contradictory the program will be easier to remove when there are low clusters.

**Network Diameter:**

Network Diameter is the shortest distance between any two distant nodes in a graph

So large network diameter results in widely dispersed graph with which ransomware may take longer time for the malware to spread throughout the network and it also indicates that malware has complex and branching execution paths.

On the other hand ransomware with low network diameter clearly states that graph is closely interconnected and could be able to spread quickly thorough the network and similarly it will be having simpler and more straightforward execution paths.

**Average Eigenvector Centrality: (Node level)**

Average Eigenvector centrality is a measure of centrality in graph theory that is used to identify nodes in a graph that have connections to other highly connected nodes and can say what nodes are highly influential in the network.

So High Average Eigenvector centrality could be due to presence of code(instructions) in ransomware will actively enables propagation to other connected nodes which have high impact on the behaviour of the ransomware.

And conversely less Average Eigenvector centrality could possibly infer that it has limited potential to spread and may only be able to infect a small number of nodes because nodes with high eigen centrality are not strongly connected to other important nodes in the network.

**Average Harmonic Mean Centrality:**

In graph theory, "Average Harmonic Mean Centrality" is not a well recognised or often used centrality metric. As a result, there is no clear use or meaning for Average Harmonic Mean Centrality in the context of a malware Control Flow Graph (CFG).

**Average Path Length:(Graph level)**

Finding the shortest route between each pair of nodes in the graph and averaging those path lengths is how you determine the average path length. The route with the fewest edges is the shortest one between two nodes. This is known as the average path length.

Path length provides the information about the connectivity and structure of the network so, if the network has small average network, then it signifies that the control flow is very extensively interconnected which indicates that the ransomware has well defined and reliable propagation mechanisms, which permitting it to quickly traverse through different parts of the program and potentially infect large number of the systems.

On the contrary hand, the ransomware CFG's longer average path length points to a more dispersed and fractured control flow. This would suggest that the ransomware is more complicated and diverse, with several branches and performance-affecting decision points. Large average route lengths may also be a hint of code evasion or encryption strategies used to thwart analysis and discovery.

**Assortativity: (Node level)**

Assortativity value(r) ranges from –1 to 1 if the value is positive which 0 to 1 (0<r<1) this indicates that nodes are connected to each other containing identical attributes. if it is negative from (-1<r<0) then nodes with dissimilar attributes are more likely to be connected in the network.

When it comes to function types or system calls, for example, assortativity analysis could be performed to see if the ransomware exhibits assortative mixing (homophily), which indicates that particular groupings of functions are more likely to be related with one another. This might suggest the existence of modular or specialised ransomware components that carry out related functions or have a similar design.

However, disassortative mixing (also known as heterophily) in function types or system calls may indicate that the ransomware is made up of various, heterogeneous components that interact with various system components. This would suggest that the malware is more complicated or polymorphic than previously thought, with numerous components controlling different features and behaviours.

**Average Page Rank:**

The fundamental tenet of PageRank is that a node's significance is based on the quantity of inbound connections it gets. A node's relevance or centrality in the graph is determined by its PageRank score.

The control flow of the ransomware is likely to be heavily dependent on nodes with high PageRank scores in the CFG. These nodes could stand in for crucial operations, system calls, or turning points that affect how the virus behaves or spreads. The nodes with a greater average influence over the whole CFG can be found by looking at the PageRank of the network.

In summary, the average PageRank of a ransomware's network can give insight into the crucial steps and activities involved in the encryption process, help in understanding the ransomware's behaviour and potential vulnerabilities. Additionally, it can be used to find potential flaws in the code that could be used to stop the process of encrypting data.

**Modularity Score:**

Modularity is a numerical measure of the strength of the community structure inside a graph, modularity assesses how easily a graph may be divided into communities or clusters in the network.

In the Ransomware high modularity can indicate distinct functionalities, such as encryption, propagation and making it harder to detect the ransomware.

Functionalities can be identified: Ransomware often comprises of several parts or modules that work together to execute the operation. It is feasible to locate groups or communities of nodes that represent various functionality by doing modularity analysis on the network.