Unsupervised Learning and Dimensionality Reduction Report

Unsupervised learning is the subset of machine learning where the models learn patterns from the input without passing on labelled data where the model aims to detect patterns within the data without explicit human intervention on the correct output. Dimensionality reduction is unsupervised learning, where it address challenges like computational complexity, overfitting, and difficulty in visualization by transforming the original high-dimensional data into a lower-dimensional representation. For the Unsupervised Learning Assignment, I choose 2 datasets, Memory-based Malware Artifacts [2]. and Airline Passenger Satisfaction [3].

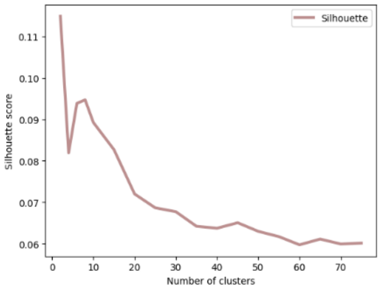
Dataset:

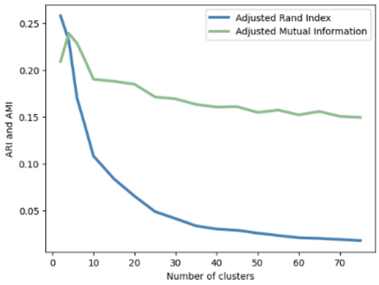
1. Airline Passenger Satisfaction: The objective behind choosing this dataset was to understand how the relationship between feedback from customers and satisfaction results is established. The aim of choosing this dataset is to explain how unsupervised algorithms and dimensionality reduction can find patterns and inform what factors are highly correlated to a satisfied (or dissatisfied) passenger. Features like online boarding, seat comfort, inflight entertainment, and legroom dominate the relationship with satisfied/dissatisfied customers. The dataset has a large number of features, leading to high-dimensional data. K-means and EM can struggle to effectively handle high-dimensional data due to the curse of dimensionality here dimensionality reduction algorithm PCA, ICA, RP, and t-SNE can help solve this issue and extract meaningful data (positive and negative correlation) which can help to state how passengers can be affected based on services, and which factors don’t imply a change in the satisfaction level. Capturing and understanding these complex interactions through dimensionality reduction like RP, can be challenging as if the algorithm misses crucial information, then results might diverge between true labels and y_pred. The dataset contains noisy and incomplete data because satisfaction is quite subjective, such as missing values and inaccuracies in passenger feedback. This helps identify how different algorithms behave and how to be robust enough to handle such data issues and extract meaningful insights despite the presence of noise. This large dataset is computationally heavy to process for unsupervised clustering algorithms but dimensionality-reduced features perform comparatively quickly.

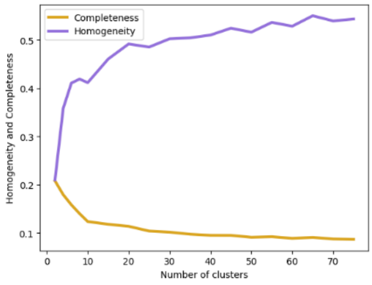
2. Memory-based Malware Artifacts: The main objective behind choosing a dataset was diverse and large feature space, making predicting underlying patterns difficult for unsupervised algorithms. The goal was to build an algorithm to identify benign and malicious processes based on memory behaviour features. Malware Features in the dataset are FeatureType, Malfind, LdrModule, Handles, Process View, and Apihooks which provide detailed information for each process and help determine whether the process is malware or benign. There is no specific definition for benign or malware processes which makes the dataset a good choice for understanding how unsupervised learning K-means and EM can extract structure and predict accurate ground truth labels. While dimensionality reduction PCA, ICA, RP, and t-SNE make the task more difficult because if the relationship among feature set is lost then y-pred might not match with ground truth resulting in poor performance. As stated above the feature set offers diverse perspectives for the model and a strong foundation for dataset analysis. Thus, analysis and identifying similarities or reduce the complexity of the dataset by extracting essential features can help in model improvement and also achieve a better understanding of algorithms in terms of malware detection. Most of the features were numerical which reduced the overhead of dimensionality, memory, and CPU usage.[4]

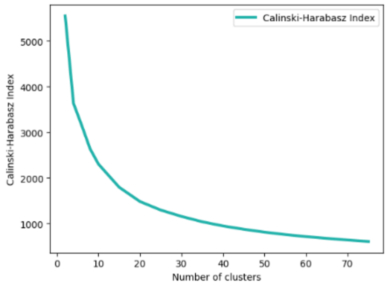
Clustering

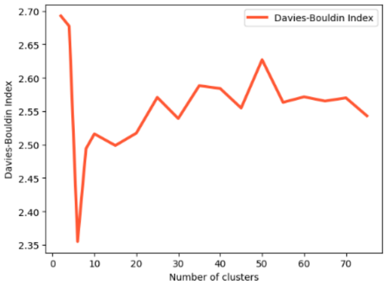
K-Means - Airline Passenger Satisfaction: As defined above Kmeans on the Airline dataset is expected to be decent for the reason that clustering passengers into groups can help gain insights into which features are necessary for satisfaction. However, the inherent subjectivity and multidimensional nature of passenger satisfaction can lead to overlapping clusters which will be challenging for Kmeans to differentiate and predict accurate results. The silhouette score is reducing as the number of clusters increases suggesting that Kmeans was unable to segregate the clusters into required numbers and the terrain of structures was unable to explore the dataset by algorithm. Both ARI and AMI are reducing over time suggesting that the optimal number of clusters for the Airline dataset should be K as 10-15. Com and Hom are contrary where homogeneity is increasing while the completeness decreases slowly indicating that the Kmeans is doing a good job of keeping samples together which are similar, keeping only members of a single class in the cluster. CalHar is in a downward trend indicating that the ratio of between-cluster dispersion to within-cluster dispersion is reducing as No. of clusters increases. While the DaBould saw a sharp decline at the value of K=8, it suddenly increased for further values but we again see a drop at the value K=15 suggesting that average similarity between clusters and their most similar neighbours is low. The performance of K-means on the Airline dataset is doing a decent job, with a degradation in performance as the cluster number increases. This indicates that the Kmeans is struggling to effectively explore the dataset's structures and segregate the data into meaningful clusters. Hyperparameter tuning like n_clusters have already been explored while parameters like n_init and max_iter did not improve performance. Results clearly state that Kmeans can’t identify relationships and patterns between the dataset and ground truth despite data scaling and normalization with all hyperparameter tuning.



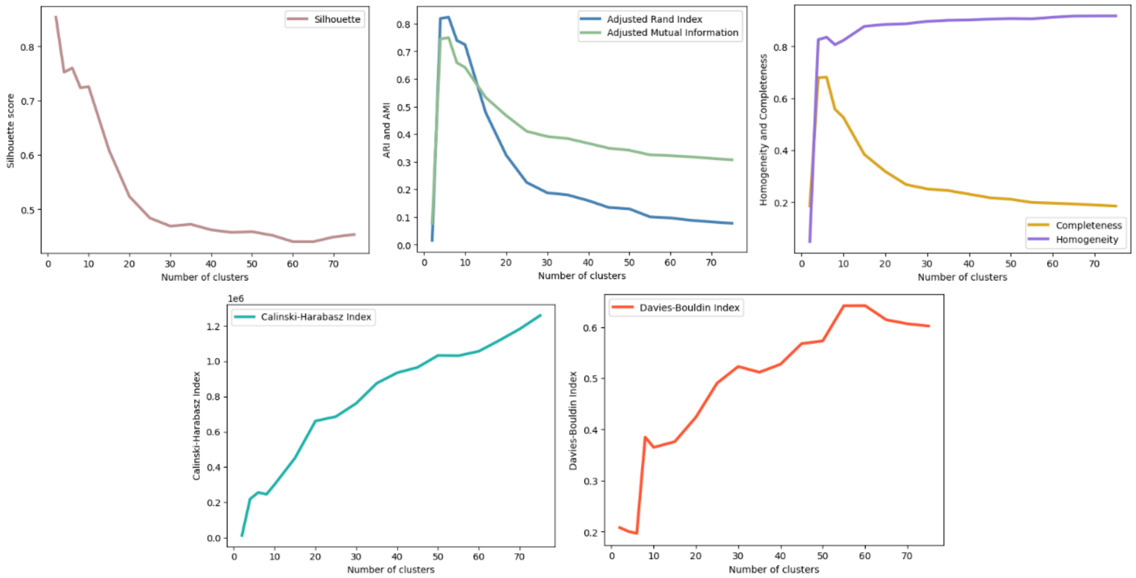




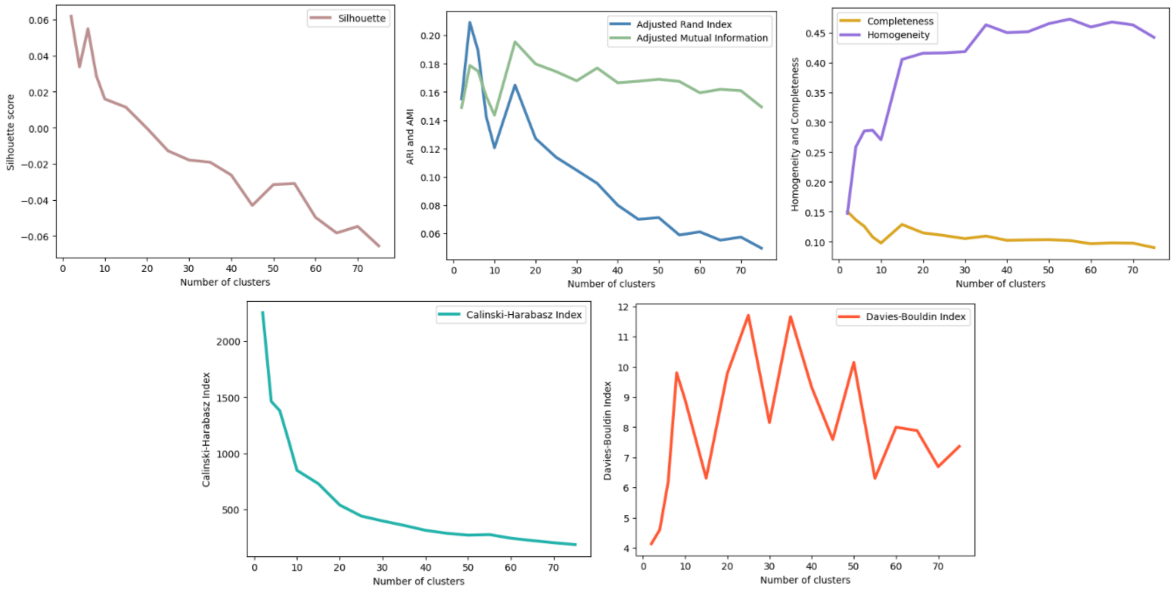




K-Means - Memory-based Malware Artifacts: Kmeans is expected to perform well on the Malware dataset reason for this is unsupervised algorithms are best suited for anomaly detection and since data is scaled and well normalized there should not be any issues in the segregation of data. As per analysis, Kmeans performed well as the silhouette score is in the range of 0.9 which indicates that the clusters are well-separated. ARI and AMI values are high for K as 7-9 but later these values decline for a large number of clusters suggesting perfect labelling between clustering results and ground truth and maximized mutual information. The same is the case for Com and Hom indicating that the ideal number for cluster should be 8 for Malware detection as data is well-separated, scaled and normalized which helps since K is sensitive to the initial cluster centre and scaled dataset. CalHar and DaBoud indices increase as the number of clusters increases hence by using the elbow technique, we get k as 8 which is the well-rounded score for the Malware dataset. Hyperparameter tuning like n_init and max_iter didn’t add value in these scores while consuming more computational time and resources.



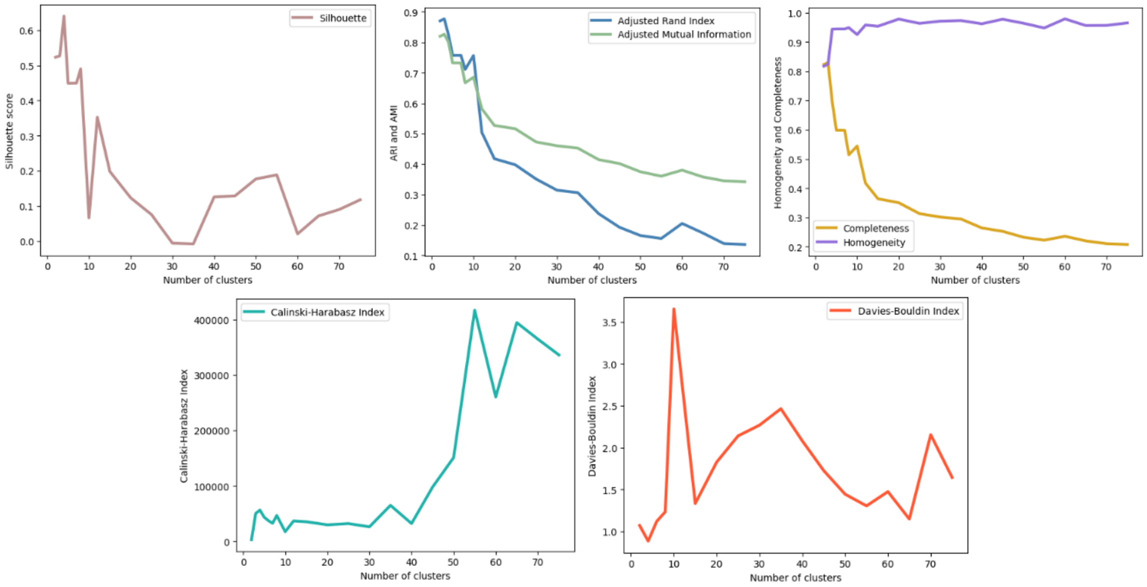
Expectation Maximization - Airline Passenger Satisfaction: Expectation Maximization is an algorithm used for probabilistic modelling and inference in the presence of latent variables. It is particularly useful in scenarios where there is missing or incomplete data, and it aims to estimate the parameters of a statistical model[6]. The initial impression is that EM should perform decently on Airline data because as per the earlier performance of K-means on the dataset was not very impressive since EM can handle missing data by calculating the E-step and M-step for the Gaussian component until convergence is achieved.



The silhouette score for EM on the Airline dataset is worse than Kmeans despite probabilistic Gaussian modelling specialised in missing data. The reason for this is that E-step and M-step are calculating wrong probabilities and assigning data points to the wrong clusters. ARI and AMI are decent for k values in the range of 1-10 indicating that small clusters can perfectly label datapoints between EM results and ground truth while maximizing mutual information. The Com and Hom graphs are contrary to each other where the Hom increases with the increase in no. of clusters while Com is inversely proportionate to cluster numbers pointing that the class labels are preserved and the decrease is marginal with an increase in cluster numbers. The homogeneity score shows that EM is unable to segregate clusters into only single class members for small cluster numbers and for this reason silhouette, ARI and AMI scores are less for EM. CalHar score is high initially indicating that the range of k values from 5-10 has healthy inter and intra dispersion ratio. DaBould index is unstable where we are aiming for lower values which hold for k values in the range of 0-5. Hence from all the analysis k value of 5 will be the ideal value considering all scores and indices. EM results are the opposite of expectation as EM calculates all Gaussian probabilities and evaluates E-step and M-step for clustering data points, the patterns and structure of the Airline dataset were not completely accurate, and EM predictions are not robust and reliable.

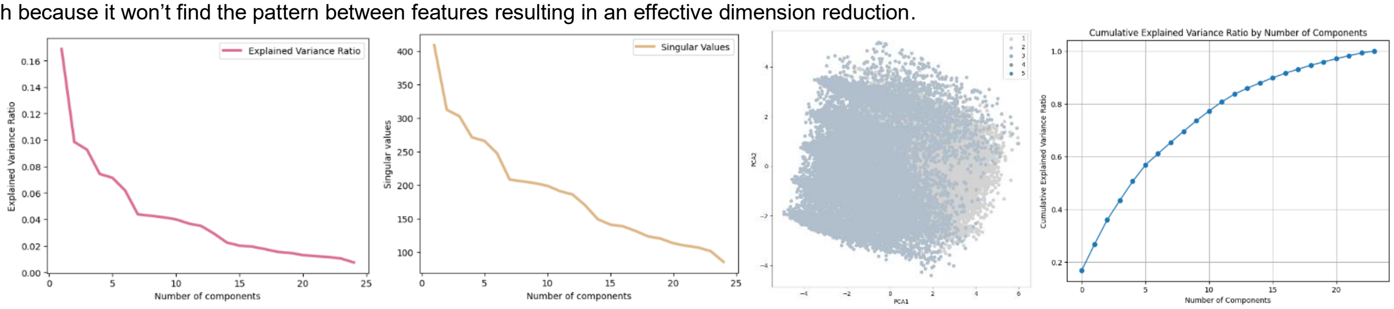
Expectation Maximization - Memory-based Malware Artifacts: As per initial expectations EM should ideally perform decently on the Malware dataset reason for this is that Kmeans has performed well setting benchmarks while EM considers Gaussian probabilities shall perform decently where E-step and M-step for Gaussian mixture model might boost performance making EM robust algorithm. As silhouette score below displays an increase in value for k=5 and then a drop indicating that EM was unable to separate clusters properly. For K=30-35 indicates that the clusters were overlapping indicating poor performance. The ARI and AMI scores are high for k in the range of 2-7 showing that cluster results and true labels are almost. The Hom and Com are contradictory as per expectation and prior observations where Hom scores are almost similar as the number of clusters increases pointing out that clusters contain single class members only. While the Com score reduces over time explaining the reason for the drop in ARI, AMI and silhouette score because data points of the same class are not assigned to the same cluster. CalHar index is almost flat for k=40 but there is a drastic increase for larger values of k, reason for this is lower values is because the features in datasets which can be benign is large for a small number of clusters hence intra and inter-distance might not be large. Ideally, the DaBould score has to be lower which is true for k=5 indicating the compact and well-segregated clusters. EM performed poorly on Malware dataset with is surprising despite using more computational resources and dataset having high covariance.

General Comparison between EM and Kmeans: The performance of Kmeans is better compared to EM because K assume the number of clusters as a hyperparameter and starts by randomly selecting K data points as cluster centres, and then iteratively updates the centroids based on the similarity between the data points and the centroids. While EM computes expected values indicated by responsibilities calculated in E-step, and in M-step the algorithm updates the model's parameters to maximise the likelihood of the observed data considering latent variables. For these complex steps in EM, it is unable to surpass Kmeans both int terms of computational time and resources, and this can be observed in the Airline and Malware dataset.



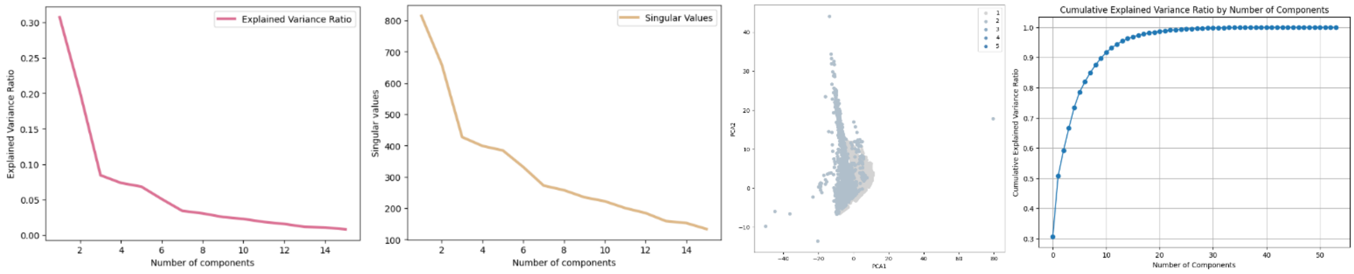
Dimensionality Reduction

PCA Airline: Airline dataset features don't have covariance among each other as the satisfaction score is highly independent hence initial expectation from PCA is not much because it won't find the pattern between features resulting in an effective dimension reduction.

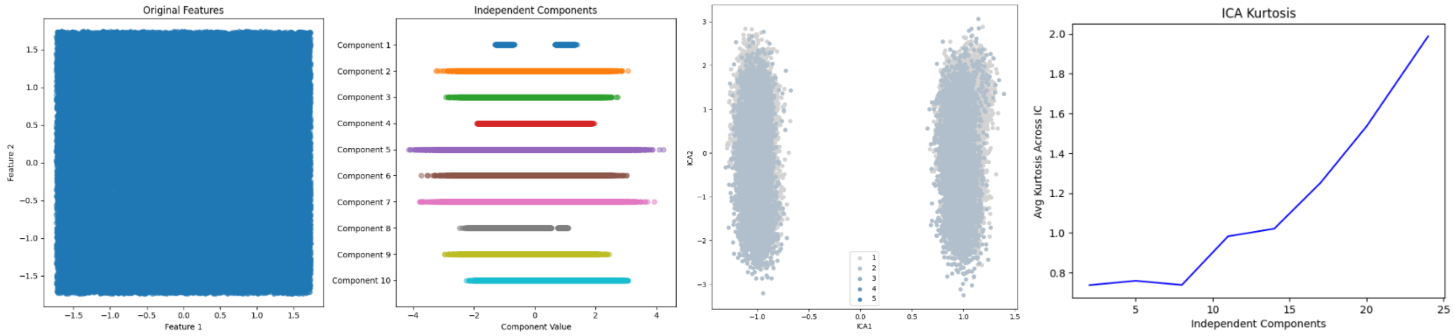


As per the above graphs explained variance ratio reduced as the number of components increased indicating that initial components are explaining most of the variance in linear equation form by PCA in datapoints. Singular values are the square roots of the eigenvalues of the covariance matrix which has to be maximized. Here Singular values also point out that smaller components have captured most of the covariance. PCA2 vs PCA1 graphs displays that still there is overlap between clusters. Samples with labels as satisfied and not satisfied should be well segregated for better performance of the clustering algorithm on the Airline dataset. At least we can say that we can see clear differentiated data points of left and right indicating the preference of customers. Ideally, components with a value of 4 have decent performance retaining 310% of the total explained variance ratio.
Hyperparameters like 95% variance or Sparse PCA were tried to check if we can improve performance but these methods didn't contribute to performance enhancements.

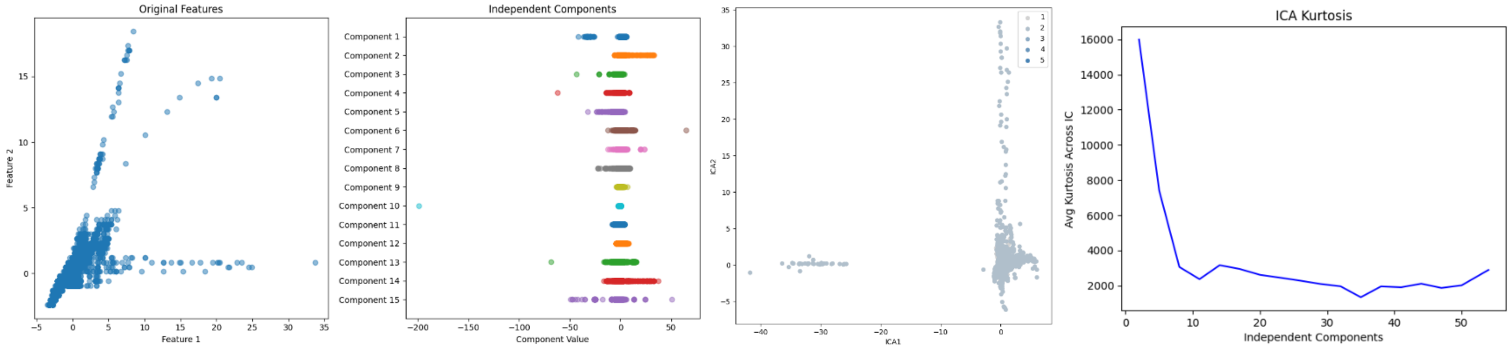
PCA Malware: PCA has to perform exceptionally on the Malware dataset because the dimensions of features are high and data points are correlated suggesting that the reduction in features will be quite high retaining most of the information forming a good starting point for the clustering algorithm. As the below graphs explain the explained variance ratio is high for the initial small number of components suggesting that most of the variance in data has been captured by the model. While PCA2 vs PCA1 cluster displays there is still overlap between clusters but we can see the boundaries of the 2 clusters. There is misclassification which can reduce the Hom, and Com scores but if 3 components retained 410% of variance from the dataset then PCA Malware can be a decent starting point for clustering. The cumulative explained variance graph points to 10 components by applying the Elbow point, this will be a good exercise to check the performance of clustering algorithms with k as 3 and 10. Sparse PCA did enhance the performance but for the higher number of components which can be considered.



ICA Airline: The airline dataset should perform exceptionally high because there is not much interrelation among the features which leads to high statistical independence of the estimated components. Feature 2 Vs Feature 1 graph is entirely covered with component 1 indicating that the first component itself has captured maximum variance in the data where this component is a linear combination of the original features. In ICA2 vs ICA1, we see 2 clusters are ground truth and ICA predictions where the datapoints are still overlapping. The expectation was that the intra-cluster segregation should also have been clearer increasing the CalHar scores when applying clustering algorithms but still, ICA will perform better as compared to PCA with clustering algorithms. ICA Kurtosis graph displays that when we have a higher number of independent components then the kurtosis score is higher but also a point to consider is this higher number of components won't capture variance as compared to that by component 1. ICA did perform as per expectation where the independent components with low variance have good grip and form a strong base for the reduction in dimensions serving as a starting point for clustering.

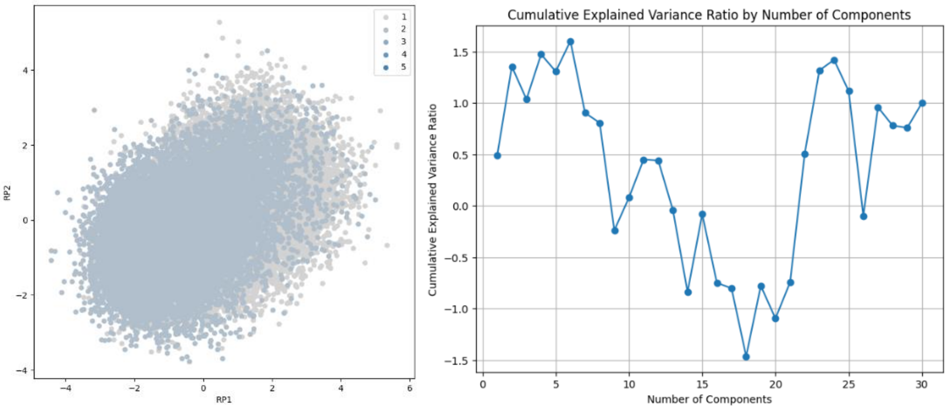


ICA Malware: Since Malware has performed well for PCA there is not much scope for ICA on Malware dataset reason for this is dataset features have high covariance among each other indicating less independence which is the basis of ICA computations.

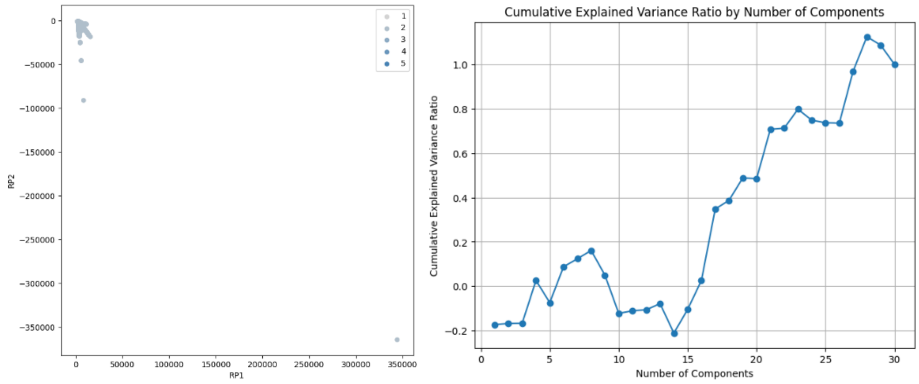


The graph of Feature 2 Vs Feature 1 points out that ICA has poor performance the ICA2 vs ICA1 is not well-segregated which can reduce the Hom, Com and CalHar scores drastically for the inter and infra cluster dispersion. Logically this makes sense because each process for Malware should ideally be in covariance with other factors like thread number, and file pointers, and if we see any abnormal values then mark that process as Malware. ICA considering independent variables won't contribute much for the reason that these features can't stand independently making ICA a bad choice for reducing dimensions by equating components which are linear equations of original features. The kurtosis graph indicates the high value for smaller independent components explaining those can be used to improve the accuracy on top of conventional clustering algorithms.

RP Airline: Randomized Projections use the randomly generated matrix to convert high-dimensional features to low-dimensional representations of the data and since this brute force method might sometimes work on well-defined clusters there are not many expectations considering previous results. RP1 vs RP2 graphs display the clusters are not separated while we can see overlapping boundaries which shows that the randomly generated matrix is not efficient enough to differentiate between two clusters the reason for this is RP was unable to understand the relationship and draw patterns from data. Explained Variance ratio by the number of components should be strictly increasing graph but for RP we are observing an irregular pattern due to the randomly generated matrix being unable to handle the increased components and this leads to a fall in the cumulative variance which ideally should have to be maximum as components increase. To find the optimal number of RP for an explanation of the Airline dataset, the Johnson-Lindenstrauss lower bound could be used. But for datasets with relatively few samples, this formula can result in a high number of projections, often in the thousands. Hence will stick to the same number that has been propagated by PCA for the Airline dataset.

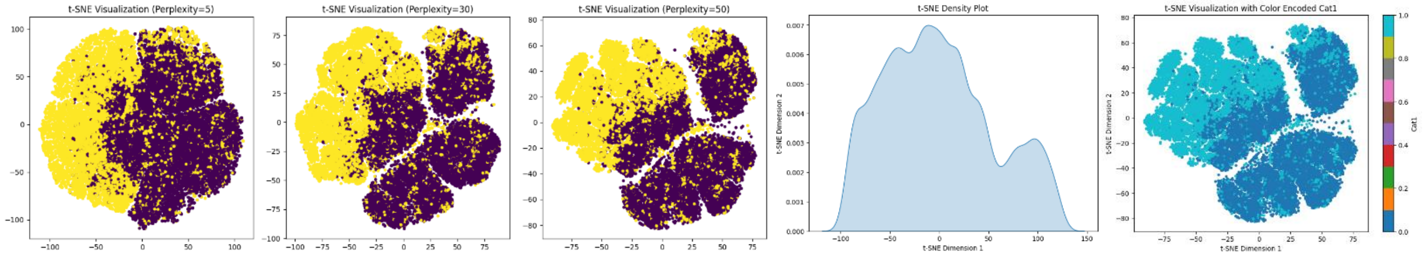


RP Malware: RP for Malware can perform better since the dataset is well scaled and normalized which can help randomized matrix to convert high dimensions to lower without much effort and can perform task effectively.



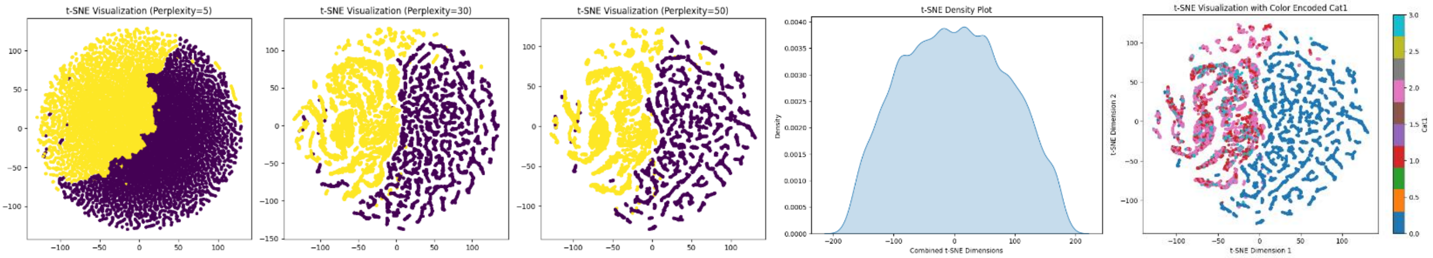
The performance of RP is poor, the cluster only contains a single member class but most of them are highly biased on the upper left corner indicating that the inter-cluster distance is overlapped. The reason for this issue is randomness used for lowering dimensions. Cumulative explained variance is increasing for a large number of components which is ideally correct and this suggests that hyperparameter tuning like Gaussian distribution rather than uniform distribution can boost performance. Error tolerance can help control the accuracy of the approximations, lowering EPS can lead to accurate results but it becomes computationally expensive. RP was expected to perform well on the Malware dataset but the randomized matrix could not extract patterns from the features that identify benign or Malware which leaves room for hyperparameter tuning.

tSNE Airline: tSNE calculates the pairwise similarities between data points using the Gaussian kernel and then normalizes to obtain conditional probabilities to get a similarity matrix. This makes tSNE a robust algorithm for applying to rigid datasets. Decent performance expectations have been set considering previous results.



tSNE visualizations, we can see a clear distinction between 2 clusters though for lower values of perplexity, the overlap between clusters is high and some data points have been misclassified which can reduce the HomCom score. But as we increase perplexity there are clear boundaries established among clusters even though the Homogeneity score is low this can be explained as the satisfaction feedback is subjective and it is difficult to extract exact patterns from the terrain and for this reason, many algorithms have failed but tSNE can draw relationships among features. tSNE density also displays the denser region has higher and more concentrated points while there is a sharp drop in a range of 50-60 indicating overlap between data points.

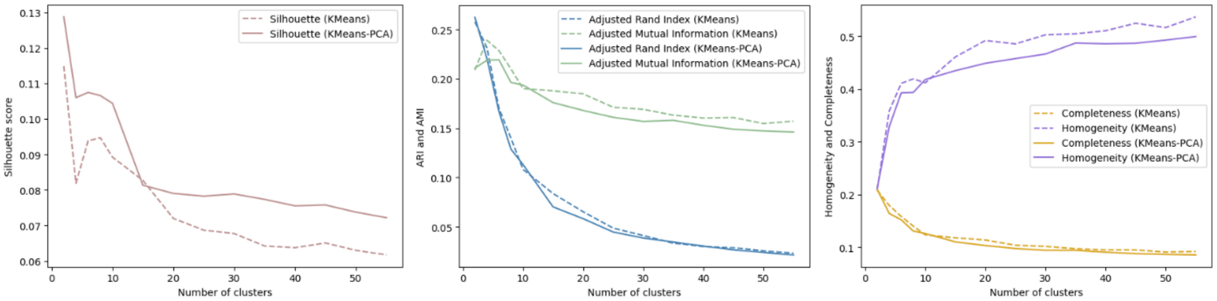
tSNE Malware: As per the above discussion tSNE is quite a robust algorithm and Malware is a well-normalized dataset hence this exercise should lead to the best performance. This initial expectation has been fulfilled and can be observed from the below graphs where perplexity with value 5 has a clear distinct boundary for clusters even though there is still misclassification and overlap visible. This drawback has been reduced with an increase in perplexity where the density of data points is reduced which is good for the dataset. Still, we can observe some misclassified points didn't get corrected which will reduce the Hom score when applied with cluster algorithms. tSNE density plot displays that the space between inter-cluster is quite low pointing to the denser cluster which is true because logically malware processes will have almost similar characteristics which leads to getting classified as Malware. This indicates that the tSNE has correctly identified the structures among features of the dataset.

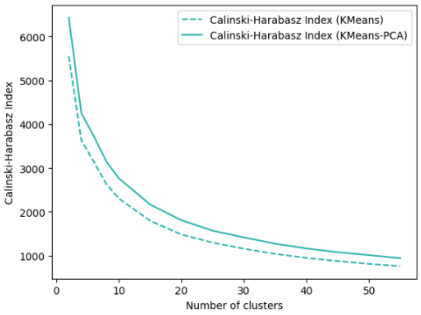


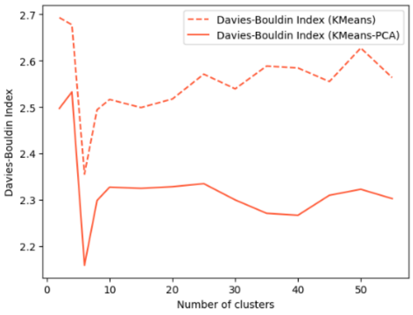
Comparison and Conclusion: Dimensionality reduction like PCA, ICA, RP and tSNE are good algorithms to apply for higher dimensional datasets (like Airline), provided features among datasets are highly correlated which yield better results like Malware dataset. If the base demand of an algorithm is to have a dataset that has high covariance, then we shouldn't use a low covariant dataset like in the case of ICA-Airline examples where results are too immature for applying a clustering algorithm. Also, RP is not very effective because of random matrix calculations which can be a big no-no for using it with clustering algorithms. PCA seems to be a safe choice for DR but still requires the dataset to have covariance among feature sets. The airline dataset itself is a very challenging dataset where tSNE was able to perform exceptionally in forming distinct clusters and improve performance, this depicts the ability of the tSNE algorithm to reduce dimensions and retain most of the relationships among low variant datasets. Hence it was great learning to apply all these algorithms on diverse datasets which resulted in interesting results.

Dimensionality Reduction with Clustering Algorithms.

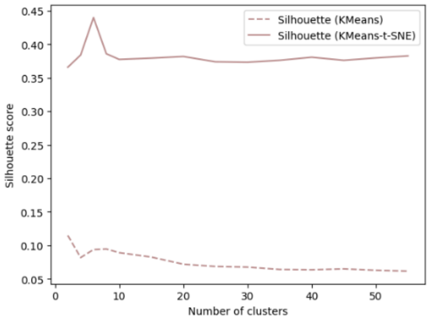
PCA-Kmeans Airline: PCA performed well on the airline dataset and Kmeans had a good hold over the terrain of the features suggesting that we can expect good performance by PCA-Kmeans. As per the below silhouette score, we can observe there is a difference in the performance initially which gradually reduces as cluster numbers increase this trend is common from previous observations. Silhouette score for K=15 still holds for PCA-Kmeans as the score is almost flat for larger values of K explaining that the clusters are well-separated with clear boundaries for smaller values. ARI and AMI scores coincide with each other leaving no room for improvement because there is not much mutual information that can influence the satisfaction score for an airline. ComHom is also mostly similar to previous Kmeans with dimension reduction, Hom score is less for PCA-Kmeans vs Kmeans and the reason for this is clusters were overlapping with few misclassifications for large value of K and this can be explained for Silhouette score and AMI, ARI scores as well. There is a slight increase in CalHar score indicating that the dispersion ratio for within-cluster and between-cluster has been improved because of dimension reduction PCA. At the same time, there is a drastic difference in the DaBould index pointing out that the clusters are compact and well-separated for a small number of clusters. Thus PCA-Kmeans did a decent job in finding the terrain structure among features for the airline dataset where we can observe differences in Silhouette and DaBould index improving. Hyperparameters like n_clusters and n_components have been optimised to get the best results.

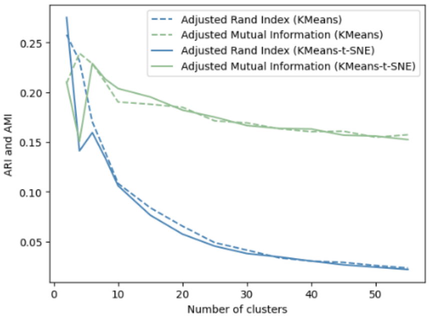


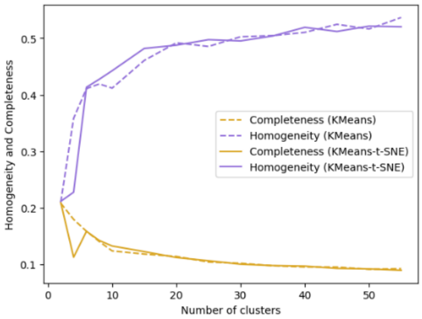


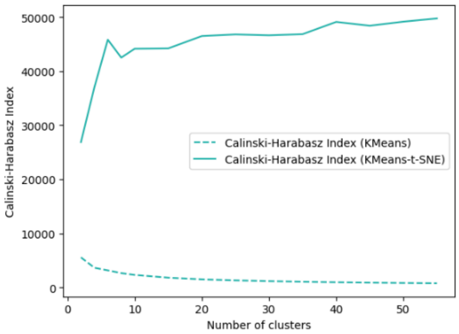


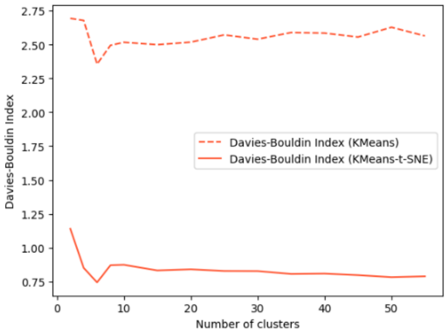
tSNE-Kmeans Airline: The performance of tSNE was exceptional on dimensionality reduction while Kmeans has also performed well in the past hence there are hopes for the tSNE-Kmeans algorithm to perform best. Silhouette score has improved by 400% which is remarkable and the reason for this is the use of Gaussian kernel for calculating similarity matrix by iteratively assigning data points and updating centroids of clusters. The ARI, AMI, Hom and Com scores are almost the same because dimensionality reduction didn't affect the mutual information when reducing dimensions while also the distribution of data points to a single class has been maintained to that of higher dimensional space. CalHar score did increase by 300% while DaBould increased by 2 times suggesting that the variance increased when applying tSNE which was the main criterion while reducing dimensions and the similarity between data points increased because tSNE calculates the pairwise similarities for similarity matrix calculations. Hyperparameters like perplexity, n_components and n_clusters were set to their optimal values to achieve this superb performance.



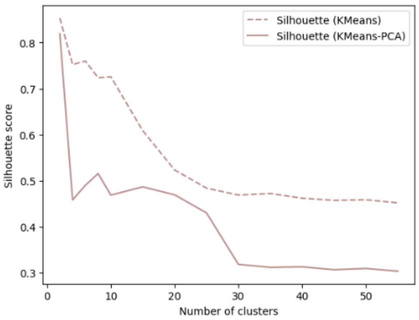


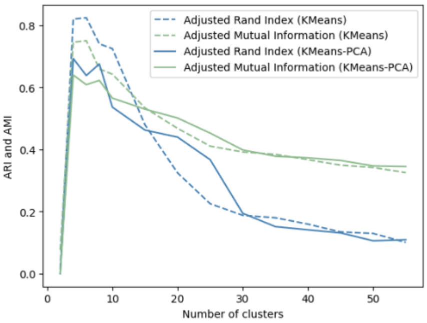


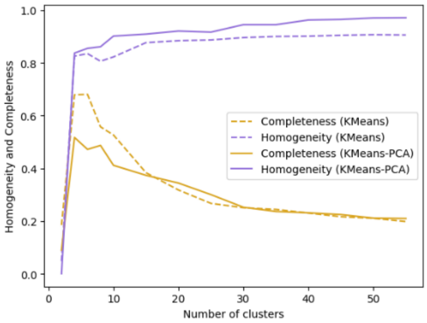


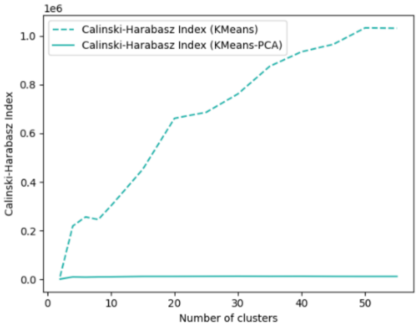


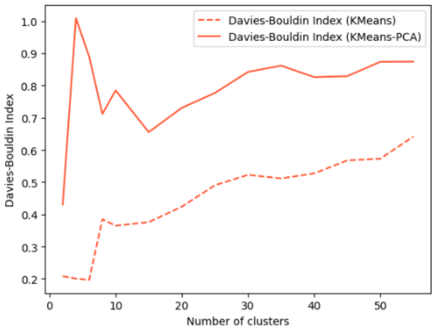
PCA-Kmeans Malware: The malware dataset was well normalized and the performance of Kmeans and PCA was great individually on the airline dataset was great hence bar has been raised for PCA-Kmeans on the malware dataset.







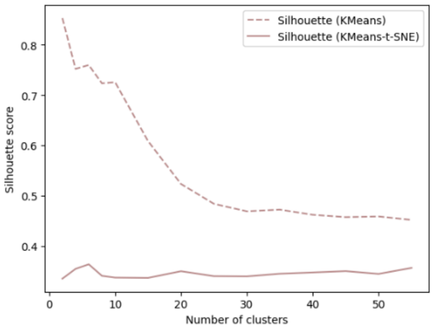


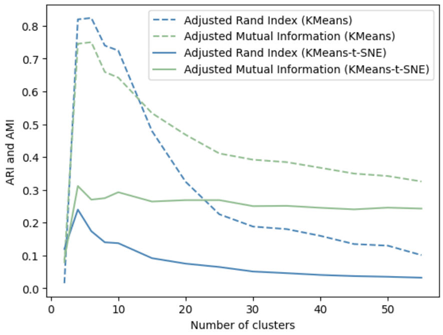


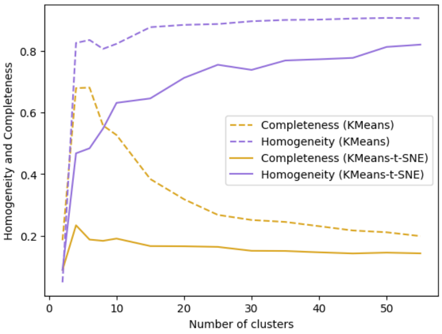
The Silhouette score for PCA-Kmeans is worse than vanilla Kmeans reason for this is the clusters from dimensionality reduced PCA had overlapping and clusters were dense displaying poor inter and intra separation and when this cluttered information is passed to Kmeans the performance dropped which can be seen in Silhouette graph. Though AMI, ARI, Com, and Hom indices more or less are similar suggesting that PCA kept mutual information intact and class labels were mapped similarly to those of higher dimensions to lower dimensions. CalHar score dropped to zero pointing that there is no variance among the data points from

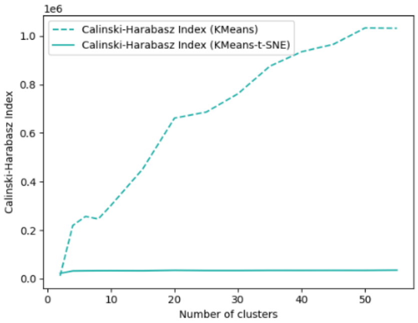
PCA-Kmeans clusters. DaBould Score increase which should ideally have to decrease suggesting that the similarity between clusters is reduced which means PCA did change the similarity when reducing dimensions resulting in poor performance. PCA-Kmeans suggests that if the dataset is well-defined (scaled and normalised properly) then stacking a complex algorithm will result in degraded results because the algorithm missed the relationship of structures and complexity between features of the Malware dataset.

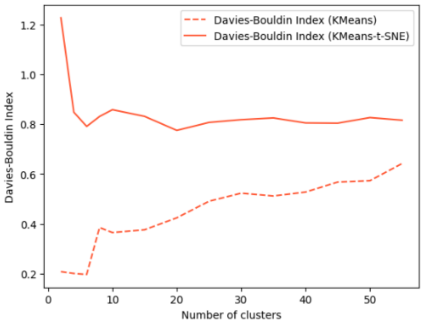
tSNE-Kmeans Malware: Though tSNE and Kmeans performance is great individually there are not many expectations from tSNE-Kmeans as per previous observations PCA-Kmeans results were not satisfying. As we can see Silhouette score for tSNE-Kmeans reduced by 50% because a highly complex dataset makes it difficult to capture the underlying structure for tSNE because the features are not independent and are not identically distributed. For this reason, tSNE with Kmeans performance has degraded even though individually they perform well. AMI and ARI indices have also been reduced this time explaining that the ground truth and clustering results don’t match while there is strong disagreement between both values. Hom and Com results have been proportionally reduced suggesting that the clusters are not well-separated and not preserving class labels. At the same time, the CalHar score is zero same as PCA-Kmeans with zero variance among data points. DaBould index increased by 6 times for initial values reason for this is the highly complex modelled Malware dataset where tSNE and Kmeans both are unable to capture complexity by debugging the structure required to classify malware and benign data points. We can firmly state that the tSNE-Kmeans algorithm is not robust enough to handle complex models with noisy data which leads to poor performance.











Conclusion: The results are contrary based on the dataset and DR (dimensionality reduction) algorithms like PCA-Kmeans for the Airline dataset were decent enough as not much gain in the silhouette score has been observed but if we see tSNE-Kmeans for the same dataset then can see a great difference between vanilla Kmeans and combined algorithm. This explains that if the DR algorithm has a good hold over the dataset, then it can improve performance provided the dataset should comply with all requirements for DR while the algorithm should also be able to extract patterns from feature sets. In the case of the Airline dataset, PCA was unable to extract meaningful information and hence PCA-Kmeans algorithm was not very effective because PCA requires a covariant dataset while tSNE did a great job in pairwise similarity mapping which resulted in identifying an underlying relationship that resulted in boosted performance in combination with Kmeans. On the contrary, we can see that PCA-Kmeans and tSNE-Kmeans algorithms have poor performance on the Malware dataset despite having high covariant data which states that when trying to overcomplicate the problem, results will degrade. The reason for this low performance was that PCA and tSNE were unable to extract complex modelling feature sets of the Malware dataset and hence structural information was lost which later didn’t pass on to clustering algorithms which resulted in bad performance. Hence results of DR in combination with clustering depend on the nature of the dataset and the ability of the DR algorithm to understand structural patterns.

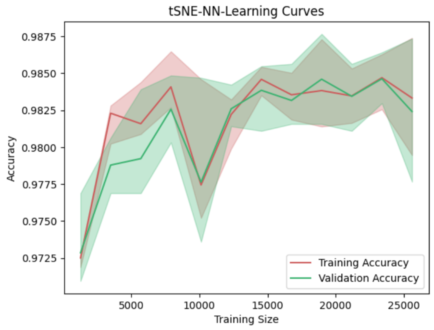
Dimensionality Reduction with Neural Network

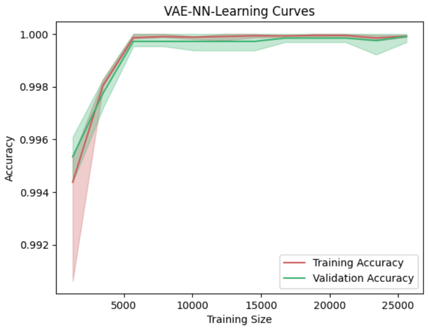
tSNE-NN and VAE-NN on Malware dataset:

Earlier in Assignment 1, I applied a hidden size of 128, and learning rate of .001, and an activation function as relu which yielded results with an accuracy of 99.9875%. Hence considering these as ideal parameters because of hyperparameter tuning and cross-validation already applied in Assignment 1 and I ran tSNE and VAE algorithms. Also, used Adam as an optimizer to apply adaptive learning rate, momentum, and weight decay on NN for better performance.[2] tSNE with NN on the Malware dataset yielded 98.51% accuracy the reason for such results was because of the dimensionality reduction applied by tSNE to understand the underlying relationship between features. This is counter-intuitive to the results we get from tSNE-Kmeans on the Malware dataset which points out that tSNE-NN is the nonlinear algorithm that can capture minute details of the features which leads to better performance. tSNE-NN is less sensitive to noisy data which results in an accuracy score while the complex terrain of the Malware dataset was able to be cracked by the combined algorithm. tSNE-Kmeans algorithm has a higher computational cost than tSNE-NN because it requires computing the similarity between all pairs of data points and then clustering them using Kmeans. While tSNE-NN has a lower computational cost as it only requires computing the similarity between the data points and the nearest neighbours. Hence tSNE-NN is hands-down the best solution with great computational resources and high accuracy.

VAE has an encoder network that takes input data and maps it to a lower-dimensional latent space that contains FCN while the decoder network takes samples from the latent space and reconstructs the original data. VAEs are trained to capture the underlying structure of the data, they can handle highly correlated features by learning to decompose them into meaningful latent variables. When this architecture is combined with NN the algorithm has superb performance with an accuracy of 99.85% which is 13 basis points higher than tSNE-NN architecture. VAE uses generative modelling which is used in LLM as a base for Transformer architecture which can generate new samples that are the same as that of the training dataset. This can be done by extracting patterns from features of the dataset which results in new samples that resemble malicious processes that yield great performance in the validation dataset too. VAE-NN is cheap in terms of computational cost because the lightweight FCN can compute lower dimensional latent variables and map them into Gaussian distribution as compared to the pairwise similarity mapping matrix of tSNE. Dimensionality Algorithms yield expected results which makes them good choice to pair up with NN to get the results provided we are using a Malware dataset because this observation can change in different data settings.

tSNE-NN-Accuracy = 98.51%
VAE-NN-Accuracy = 99.85%





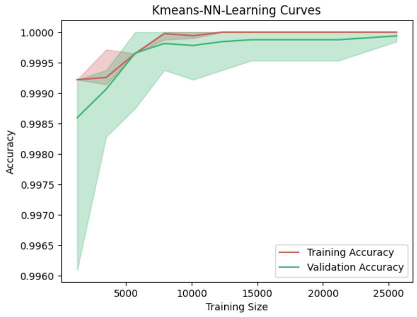
Clustering Algorithms with Neural Network

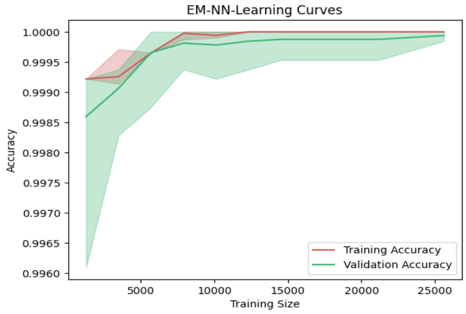
Kmeans-NN and EM-NN on Malware dataset:

Initial Expectations from Kmeans in combination with NN are high since the Dimensionality Reduction algorithms applied had decent results with NN. Kmeans should also be able to capture the complexity of the model and understand the underlying pattern to get meaningful results. As we can observe the learning curve we can see the final Accuracy score is 99.9875% which is similar to vanilla back_prop NN when applied to the malware dataset this means there is no significant additional in the performance from the Kmeans clustering algorithm. We expected that when Kmeans reduced the dimensions of the dataset it would understand the relationships that would have been propagated to the NN in terms of input and when this input has been processed by MLPClassifier, it would understand and predict accurate results. But since all these are missing, we are getting results the same as that of the backdrop suggesting that the Kmeans were unable to add any value to the score or reduce predicting ability.

Again, as per performance from Kmeans-NN, there are no explicit expectations from EM-NN. We can observe the EM-NN resulted in the same learning curve graph as that of Kmeans-NN which means that both clustering algorithms do not add any value in NN. As per computational time, EM calculates the E-steps and M-steps for reducing the dimensions of the dataset into low-dimensional space. But also point to consider is that when reducing dimensions EM and Kmeans didn't change the topology of data points which can also be observed from ARI, AMI, Hom and Com graphs and for that reason the accuracy of combined algorithms EM-NN didn't change which is a good point to notice about clustering algorithms.

Kmeans-NN-Accuracy = 99.9875%
EM-NN-Accuracy = 99.9875 %





Conclusion:

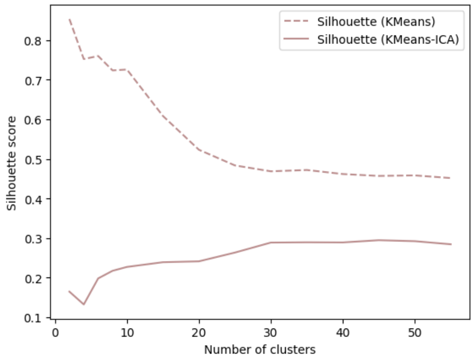
All in all, Clustering and dimensionality reduction algorithms are not improving accuracy on the other side they are just consuming computational resources and the results of feature sets are simply passed to MLPClassifier which doesn't contain any value addition hence we don't see a significant rise in accuracy when applying combinational algorithms.

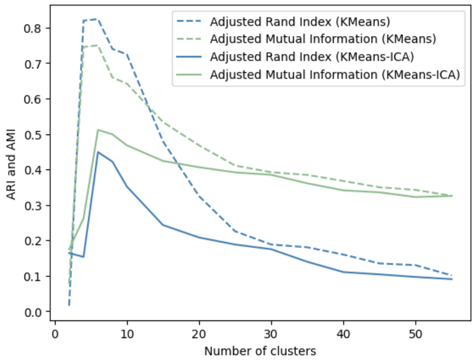
References

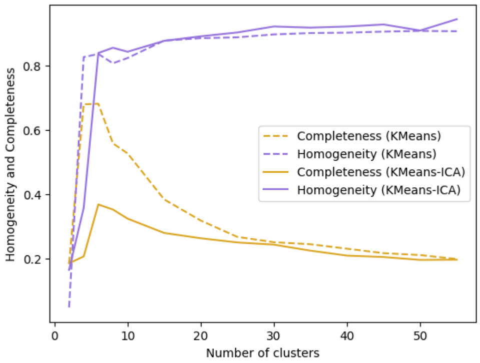
[1].https://cloud.google.com/discover/what-is-supervised-learning#:~:text=Supervised%20learning%20is%20a%20category,the%20input%20and%20the%20outputs.
[2]. https://www.kaggle.com/datasets/jlcole/cic-malmem-2022/data
[3]. https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction
[4]. Assignment 1 writeup.
[5]. Assignment 2 writeup.
[6]. https://proceedings.neurips.cc/paper_files/paper/2003/file/8208974663db80265e9bfe7b222dcb18-Paper.pdf

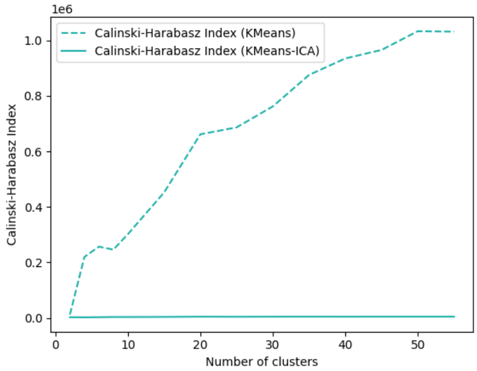
Appendix:

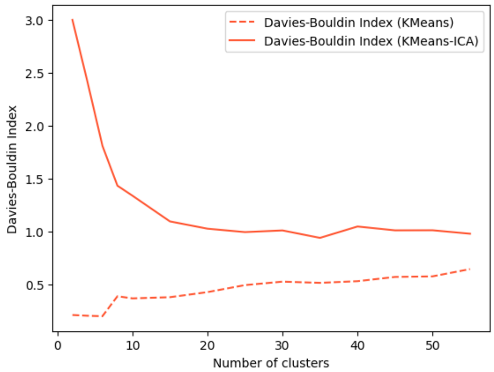
Malware ICA-kmeans



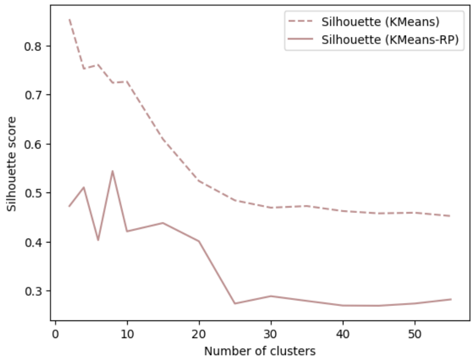


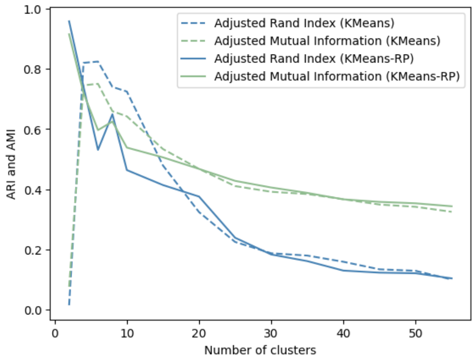


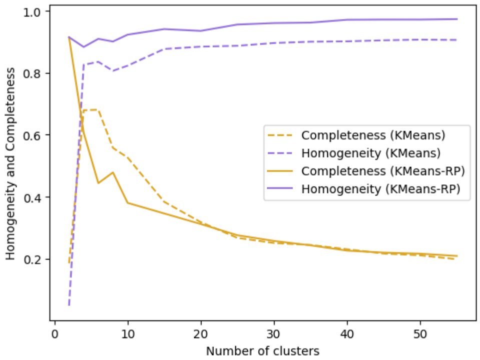


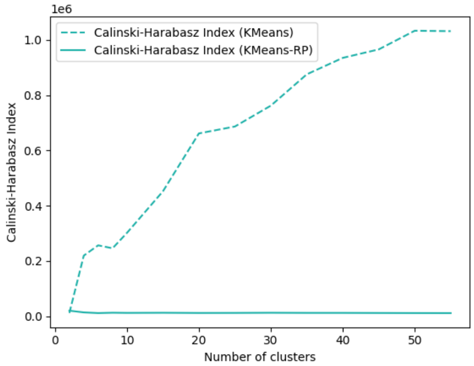


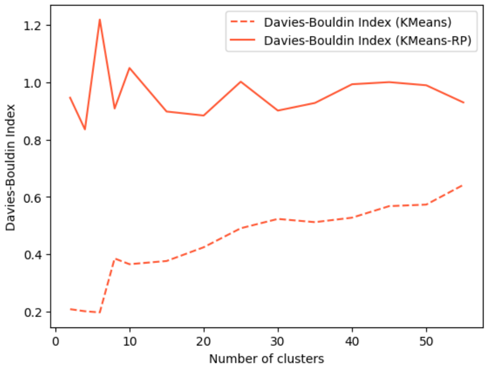
Malware RP-Kmeans



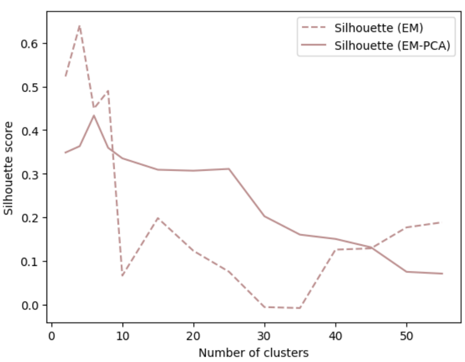


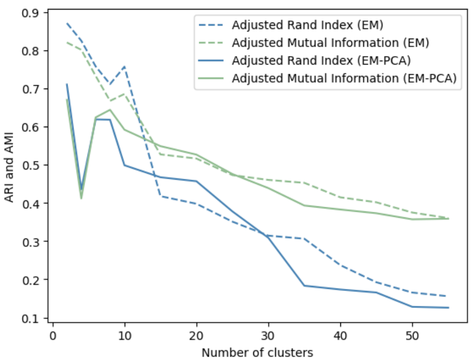


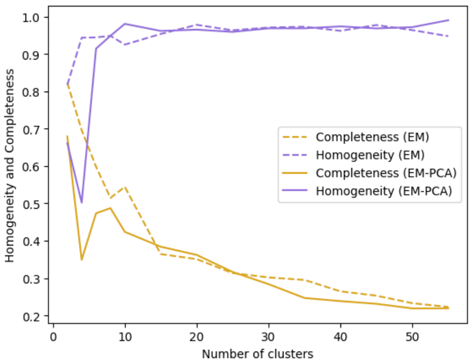


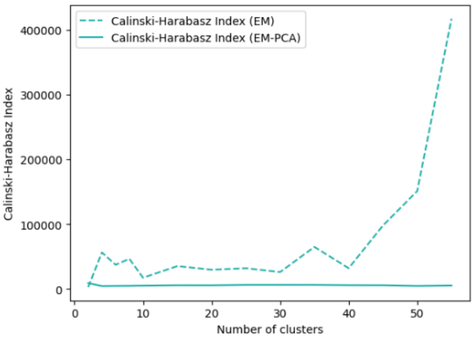


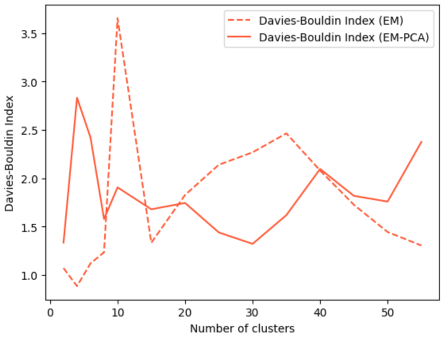
Malware-PCA-EM:



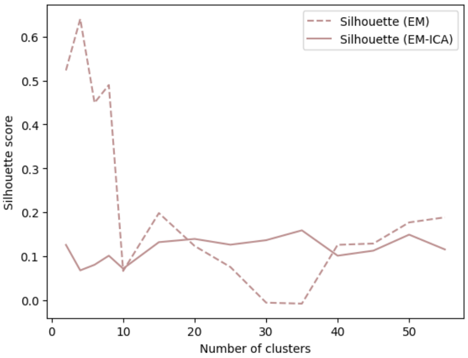


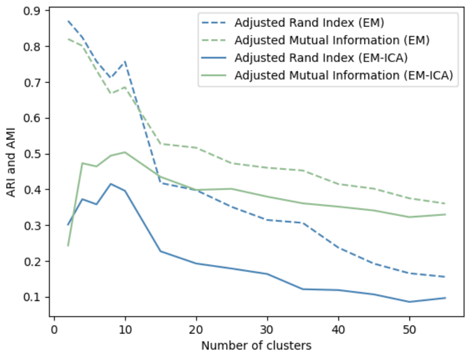


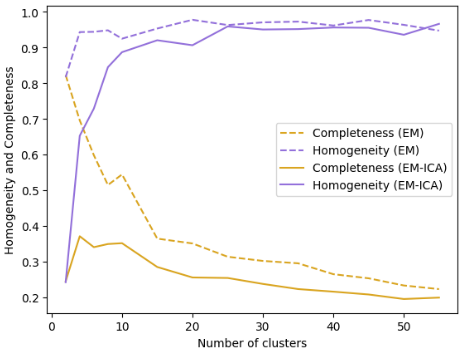


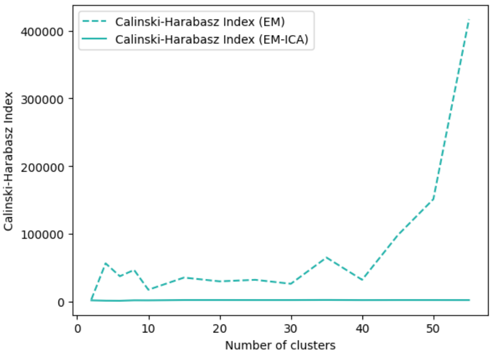


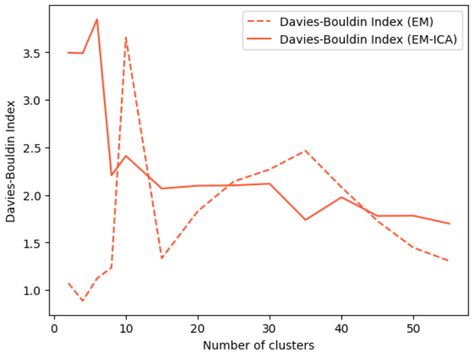
Malware-ICA-EM:



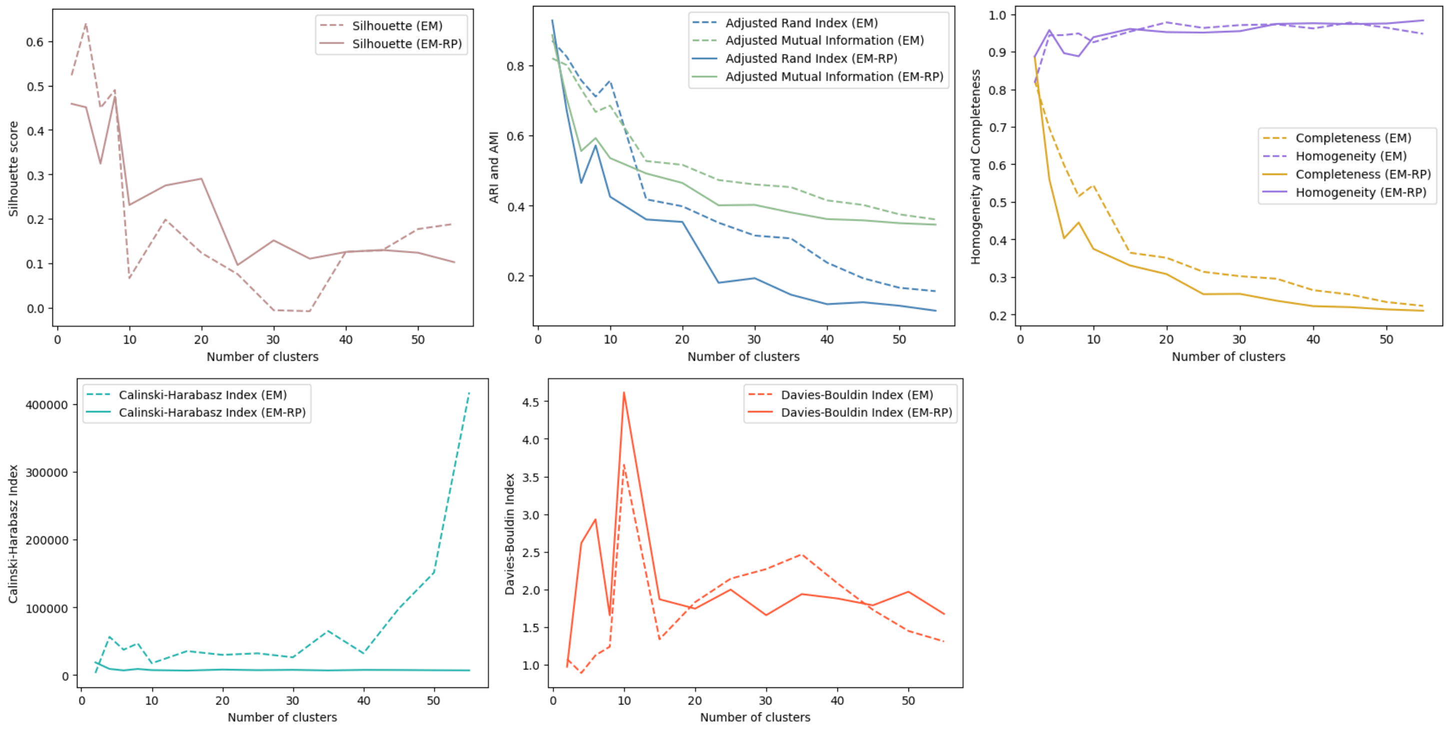




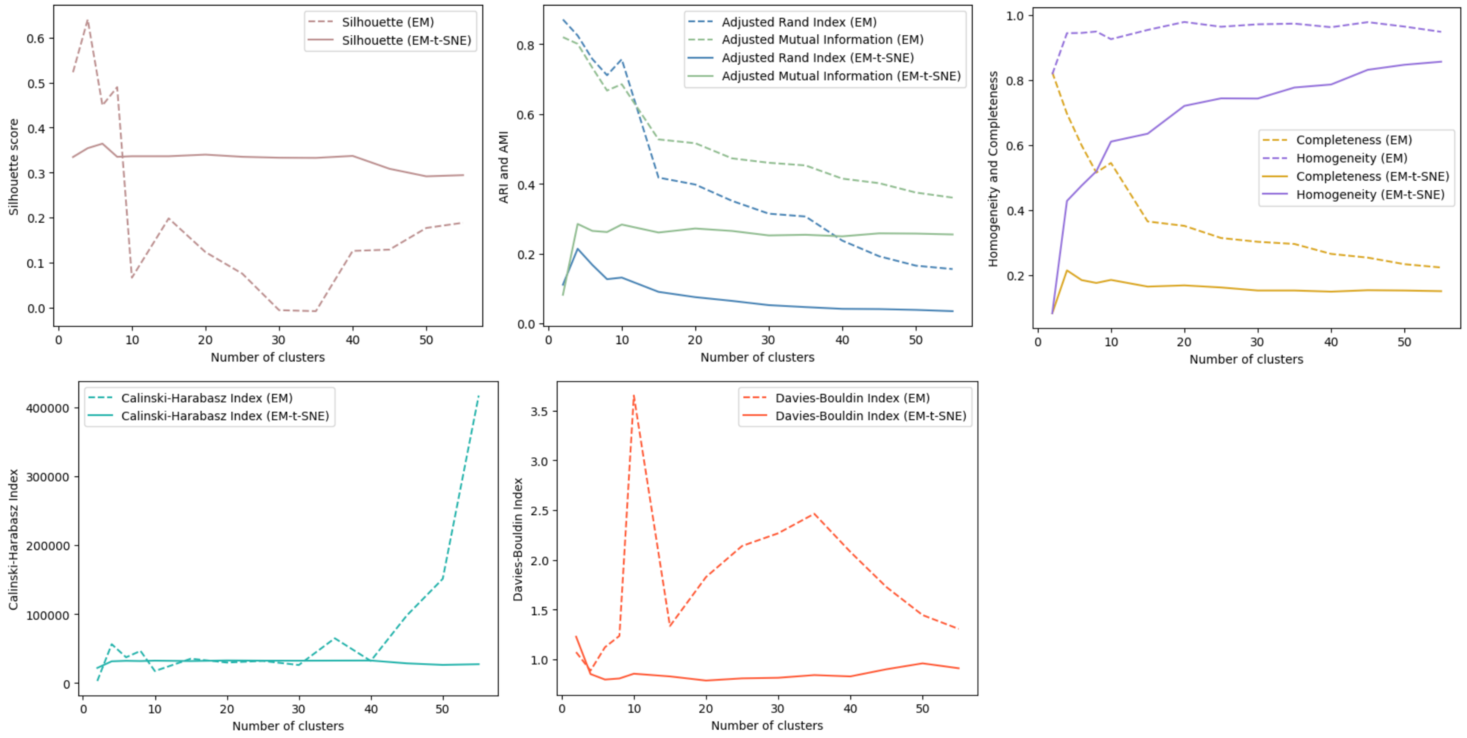




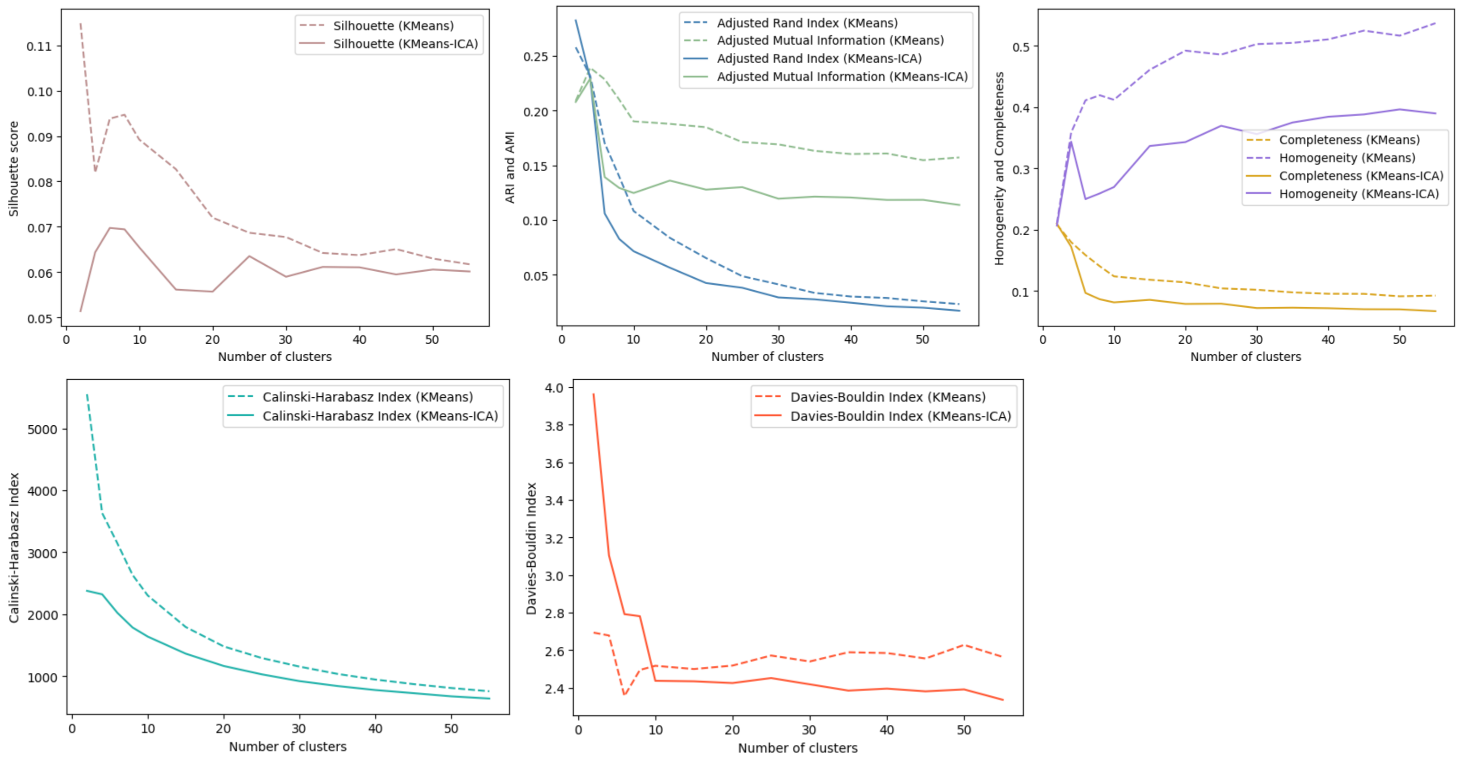
Malware-RP-EM



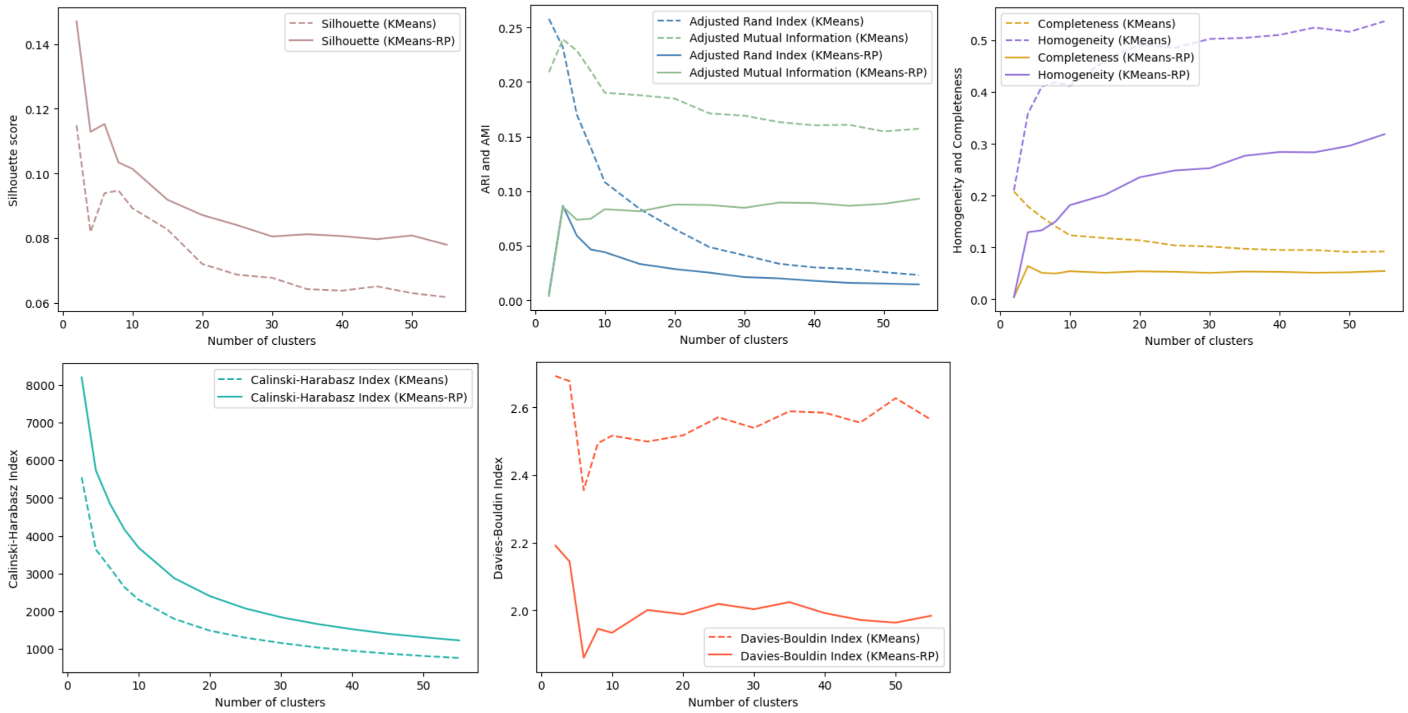
Malware-tSNE-EM



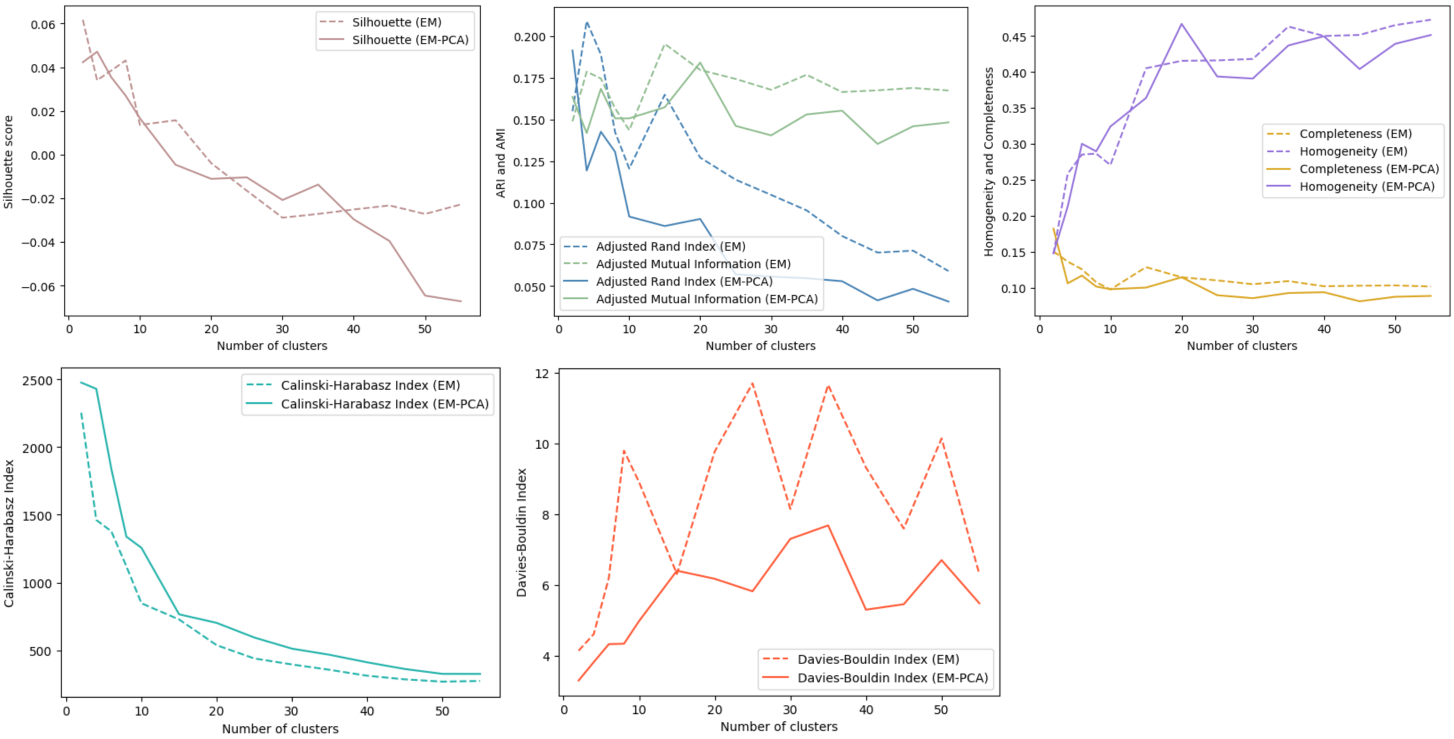
Airline-ICA-Kmeans



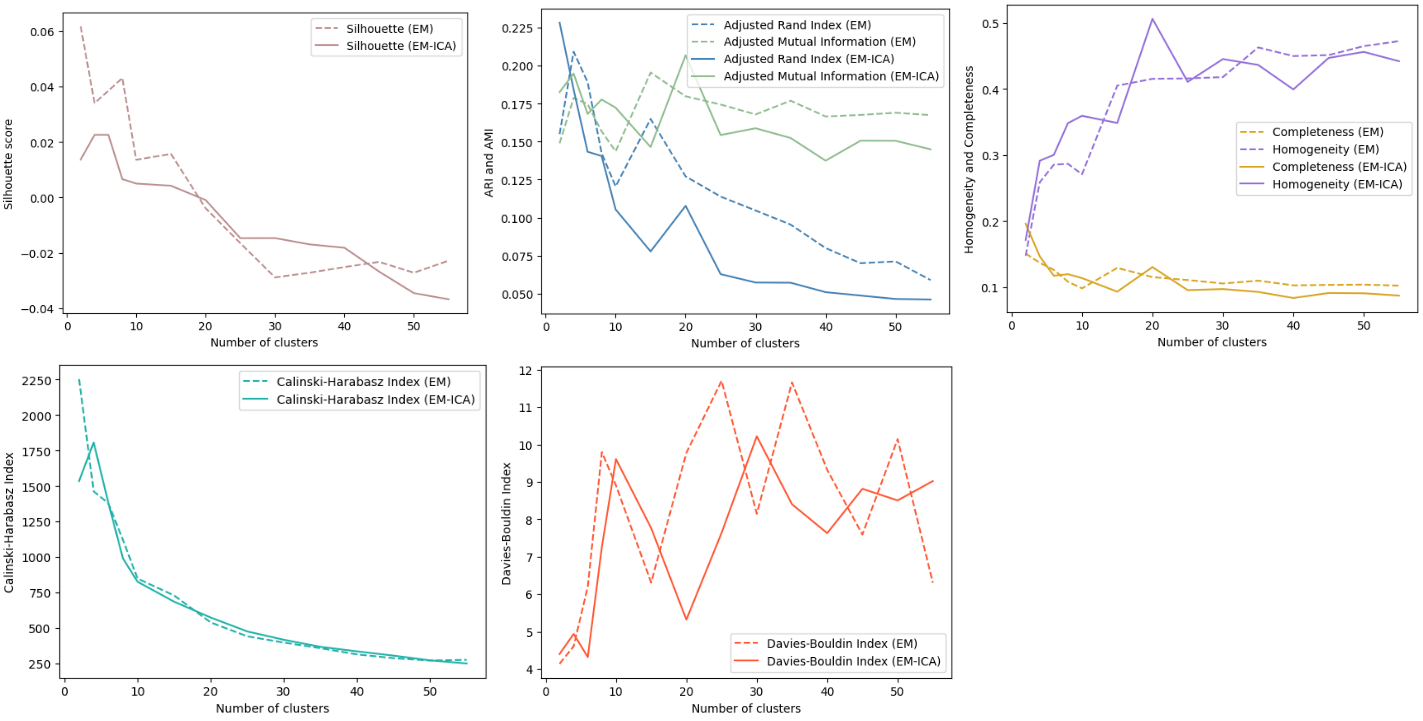
Airline-RP-Kmeans



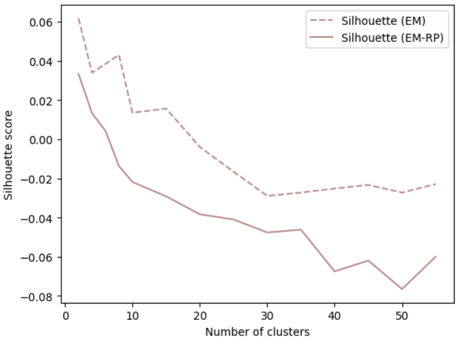
Airline-PCA-EM

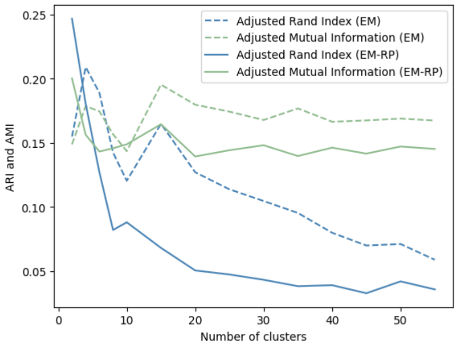


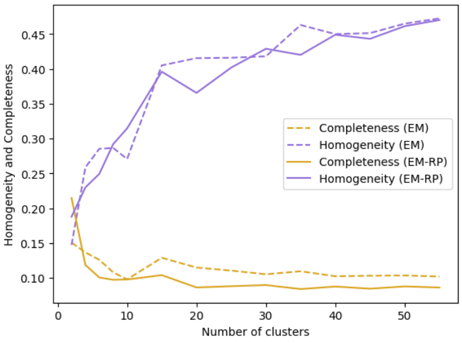
Airline-ICA-EM

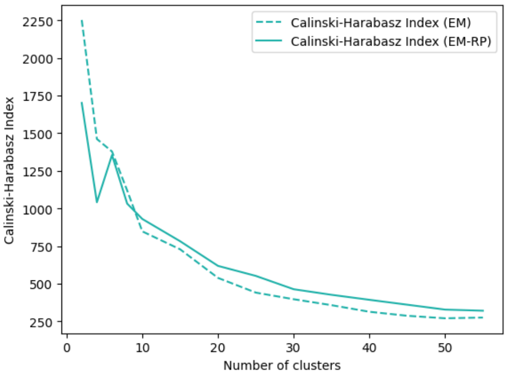


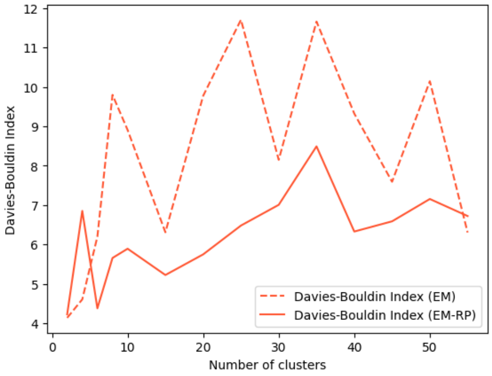
Airline-RP-EM



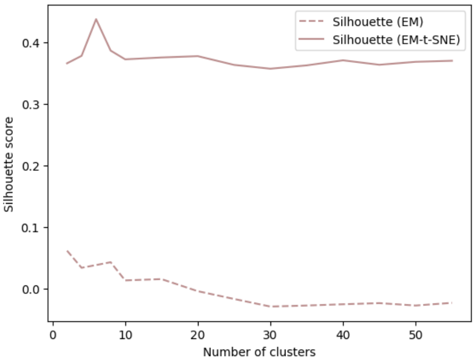


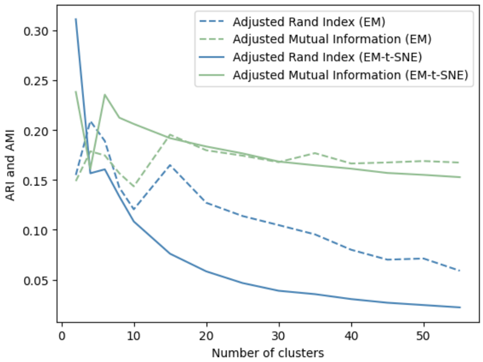


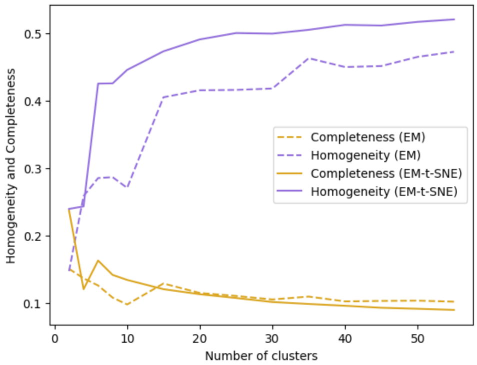


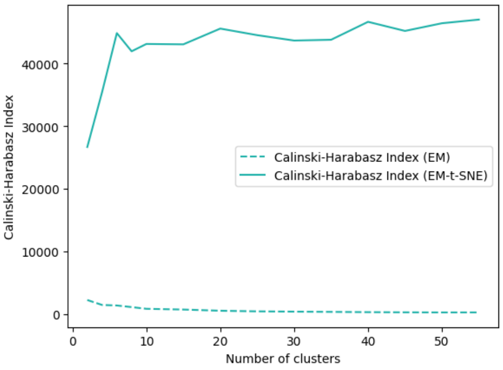


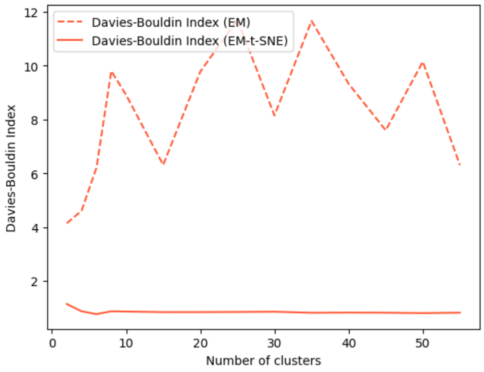
Airline-tSNE-EM


Silhouette (EM)
Silhouette (EM-t-SNE)
Silhouette score
Number of clusters


Adjusted Rand Index (EM)
Adjusted Mutual Information (EM)
Adjusted Rand Index (EM-t-SNE)
Adjusted Mutual Information (EM-t-SNE)
ARI and AMI
Number of clusters


Completeness (EM)
Homogeneity (EM)
Completeness (EM-t-SNE)
Homogeneity (EM-t-SNE)
Homogeneity and Completeness
Number of clusters


Calinski-Harabasz Index (EM)
Calinski-Harabasz Index (EM-t-SNE)
Calinski-Harabasz Index
Number of clusters


Davies-Bouldin Index (EM)
Davies-Bouldin Index (EM-t-SNE)
Davies-Bouldin Index
Number of clusters