Project 3 Unsupervised Learning

Name: Luping Yang

GATech ID: lyang38

***Abstract*— In this project, I explore the unsupervised learning algorithms: K-Means clustering and Expectation Maximization. Four dimension-reduction algorithms: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Random Projection Analysis (RPA) and Random Forest Classification (RFC) are applied to reduce the dimension of datasets. Then the reduced data is applied to Artificial Neural Network. The impact of dimension reduction on clustering and classification is examined. In this project, the exploration consists of five parts: 1) analysis of clustering algorithms; 2) analysis dimension reduction algorithms; 3) clustering using dim-reduced data; 4) ANN using dim-reduced data; 5) ANN using dim-reduced data appended with clustering labels.**

Keywords*— *K-Means clustering, Expectation Maximization, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Random Projection (RPA) and Random Forest Classification (RFC).

I. Introduction to Datasets

A. *Phishing Websits Dataset*

The Phishing Websites dataset is downloaded from OpenML PhishingWebsites. This dataset consists of 45 features and 11055 samples. Each sample is labeled 1 and -1, indicating if the website is a phishing website or not, with 1 being phishing website and -1 being not.

For the Phishing dataset, all the features and the label are categorical. Most of the features are binary values {-1, 1} and some features are {-1, 0, 1}. For the features with two values, one-hot encoding is used to convert these features to {0, 1}. For the features with three values, one-hot encoding is used to convert the features to {0, 1}. Thus, all features are {0, 1}.

The Phishing data was used in my first assignment. I will use it in clustering, dimension reduction, artificial neural network.

B. *MNIST Handwritten Dataset*

The Mnist dataset is one of the most famous dataset in machine learning field. The dataset is available at http://yann.lecun.com/exdb/mnist/. This dataset has a training set of 60,000 samples and a test set of 10,000 samples. Each sample is a hand-written digit. The samples have been size-normalized and centered in a fixed-size image.

The images were centered in a 28x28 image by computing the center of mass of the pixels and translating the image to position this point at the center of the 28x28 field. Then each sample is flattened to a vector with dimension of 784. With that being said, the mnist samples have 784 features, which is a lot. Thus mnist dataset serves as an interesting dataset for clustering algorithms and dimension reduction algorithms.

II. Part 1: Clustering Algorithm

In this section, I document how to determine the number of clustering for K-Means and EM algorithms.

K-means algorithm starts with random cluster centers. Based on the distance of samples to the cluster centers, each sample is assigned to a cluster. Then the cluster centers are recomputed and then the samples are reassigned until the algorithm converges.

Expectation Maximization (EM) iteratively run two steps. Expectation step computes the probability of each points belonging to a particular cluster. Maximization step calculates the cluster centers based on the probability of a sample belonging to a particular cluster. EM converges to local or global maximum, but it will not diverge.

A. *Phish Dataset*

Figure 1 presents how different metrics varies with the number of clusters. The metrics are Sum of Squared Error (SSE) for K-Means, Log-Likelihood for EM, Homogeneity Score, Adjusted Mutual Information and Accuracy Score.

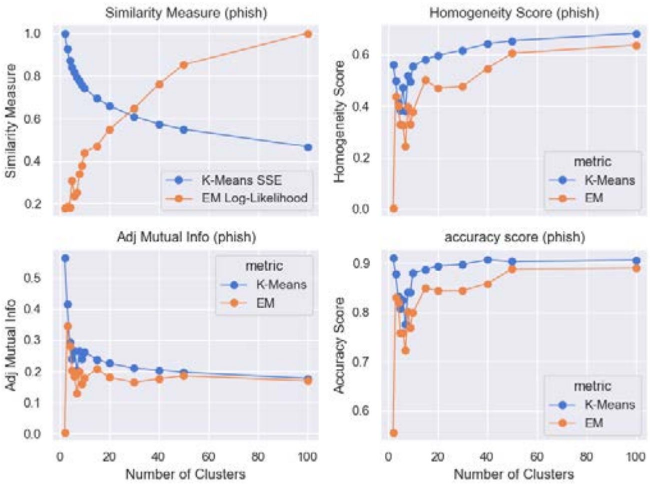

Fig. 1. Metrics as a function of number of clusters (Phish data).

How to choose the number of clusters, K? If we choose K=1, then the whole dataset is one cluster and it is against our goal to divide the dataset. If we choose K being number of samples, then we will achieve a perfect score but this is too granular and leads us to overfitting. One method to determine the number of clusters is Elbow Method: the number of clusters should be chosen so that adding one more cluster does not improve modeling of data significantly.

Since K-Means use distance to assign samples to clusters. The distance used in my K-Means algorithm is Euclidean distance. Thus, the objective function of my K-Means algorithm is Sum of Squared Error (SSE): the sum of squared distance of each sample to its corresponding cluster center. The magnitude of SSE is not important (it depends on how many samples in dataset and the scale of features), but the trend is important, thus SSE is normalized by largest SSE. From Figure 1, it is observed that the SSE decrease as the number of clusters increases for Phish data. This is reasonable, since the more clusters, the less distance will a point be away from a cluster, resulting to a smaller SSE. From Figure 1, the decrease of SSE becomes much slower after K=20. According to Elbow Method, K=20 is good for Phish.

Since the EM algorism is a probability model. The objective function used in my EM algorithm is log-likelihood function. The magnitude of Log-likelihood is not important (it depends on how many samples in dataset and the scale of features), but the trend is important, thus Log-Likelihood is normalized by largest Log-likelihood. The observation makes sense for the same reason as SSE: the more clusters, the samples make clusters better locally, leading to higher log-likelihood value. From Figure 1, the increase of Log-likelihood becomes much slower after K=40.

According to SK-Learn, a clustering result satisfies homogeneity if all clusters contain only data points which are members of a single class. For Figure 1, the increase of Homogeneity score becomes much slower after K=20. The Homogeneity score of K-Means is higher than EM, which can be better explained when checking the scatter plots.

According to SK-Learn, Adjusted Mutual Information (AMI) is an adjustment of the Mutual Information (MI) score to account for chance. It accounts for the fact that the MI is generally higher for two clusters with a larger number of clusters, regardless of whether there is actually more information shared. From Figure 1, it is observed the AMI does not change much after K=20. The AMI score of K-Means is higher than EM.

Ideally, assigned cluster label of each sample should agree will the true class label of the sample. Thus calculating accuracy score can tells us how the K-Means and EM algorithms performs. From Figure 1, the accuracy of K-Means and EM do not increase after K=20. It is interesting the K-Means has better accuracy than EM.

In Figure 2, since there are 45 features and 2 classes for Phish data, it is extremely hard to view the clusters in 45 dimensions. Thus, I use the TSNE algorithms to map the 45-dim samples to 2-dim space, so that we can plot the samples and color the samples either by their true class labels, cluster labels, and cluster predicted labels by majority vote. The left two plots are colored by cluster labels generated by K-means and EM. The cluster predicted labels are classification labels based on K-means and EM cluster labels using majority vote.

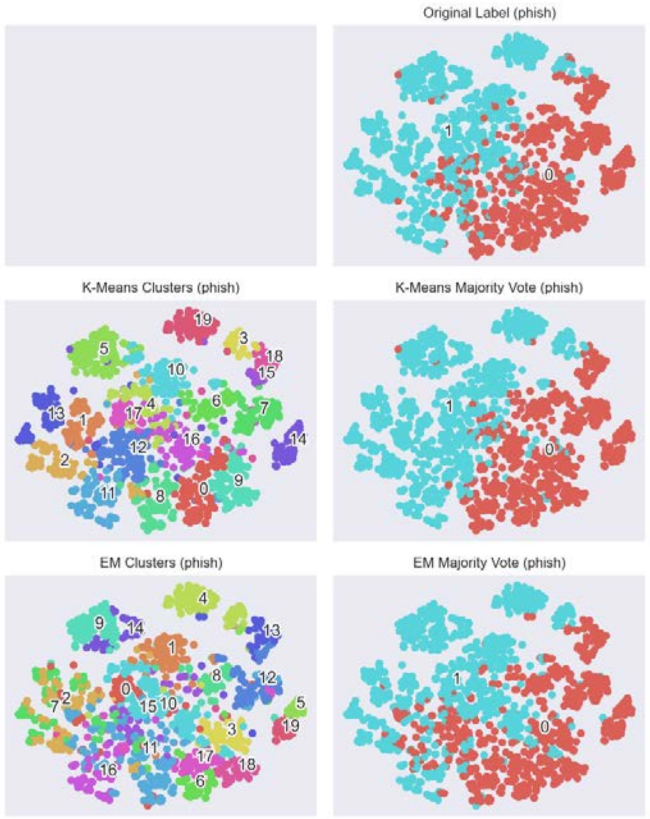

Fig. 2. TSNE dimension reduced clusters plot (Phish data).

Looking at the left two scatter plots, it is observed that the K-Means groups the samples more distinctly than EM. In another words, the cluster boundary of EM is blurrier than K-Means, since EM is soft boundary clustering method. For clusters in the outer region, both K-means and EM can separate them well, even though, differently. For the clusters in the center regions, it is hard to separate them for both K-means and EM.

Looking at the right three scatter plots, both K-Means and EM predicted the sample labels generally well, which implies that our clusters algorithms work well for Phish data. K-Means algorithm achieves a better match of true sample classification than EM, which agrees with that fact that K-Means has a higher accuracy score than EM.

B. *Mnist Dataset*

Figure 3 presents how different metrics varies with the number of clusters. These metrics help us to determine the number of clusters for Mnist dataset.

Since K-Means use distance to assign samples to clusters. The distance used in my K-Means algorithm is Euclidean distance. From Figure 3, it is observed that the SSE decrease as the number of clusters increases for Mnist data. From Figure 3, the decrease of SSE becomes much slower after K=30. According to Elbow Method, K=30 is good for Mnist.

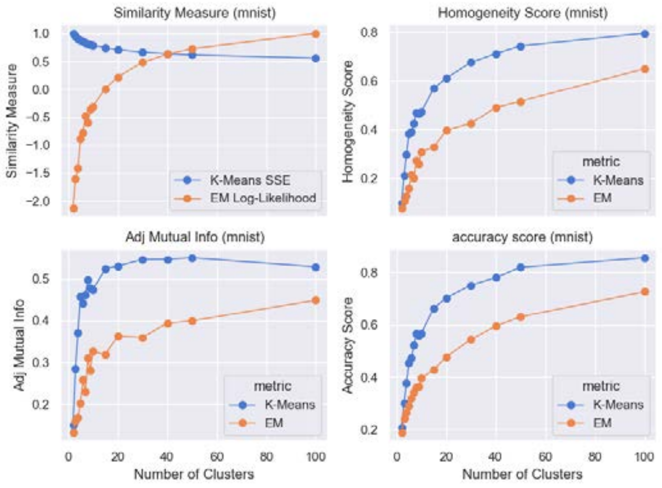

Fig. 3. Metrics as a function of number of clusters (Mnist data).

From Figure 3, the increase of Log-likelihood becomes much slower after K=30. Thus, the number of clusters are determined to be 30 for Mnist data.

For Figure 3, the increase of Homogeneity score becomes much slower after K=30. The Homogeneity score of K-Means is higher than EM, implying that K-means does a better job in grouping the samples than EM for Mnist data.

From Figure 3, it is observed the AMI does not change much after K=20 for Mnist data. The AMI score of K-Means is much higher than EM.

Accuracy score can tell us how the K-Means and EM algorithms performs in predicting the classification of samples. From Figure 3, the accuracy of K-Means and EM do not increase much after K=50. It is interesting that the K-Means has better accuracy than EM.

One notable observation in Figure 3 is that K-Means performs better than EM in terms of all performance measures in Figure 3. This is because K-means is more suitable for Minst data: the Mnist data are image data. Each sample has 784 features and most of the features are 0. For features that are not 0, most of their values are in the range of 150 to 260. The features are not continuous and there is big jump in their PDF. This makes EM less effective than K-means to model Mnist data, since I chose Gaussian Mixture model as EM. The Gaussian model is good at model data whose underline probability model is Gaussian. But for Mnist data, Gaussian PDF may not be a good PDF model for the Mnist features. If I can use a PDF that can better represent the distributions of Mnist features, I expect the performance of EM can improve.

In Figure 4, since there are 784 features and 10 classed for Mnist data, it is extreme to view the clusters in 784 dimensions. Thus I use the TSNE algorithms to map the 784-dim samples to 2-dim samples, so the we can plot the samples and color the samples either by their true class labels, cluster labels, and cluster predicted labels. The left two plots are colored by cluster labels generated by K-means and EM. The cluster predicted labels are classification labels based on K-means and EM cluster labels using majority vote method.

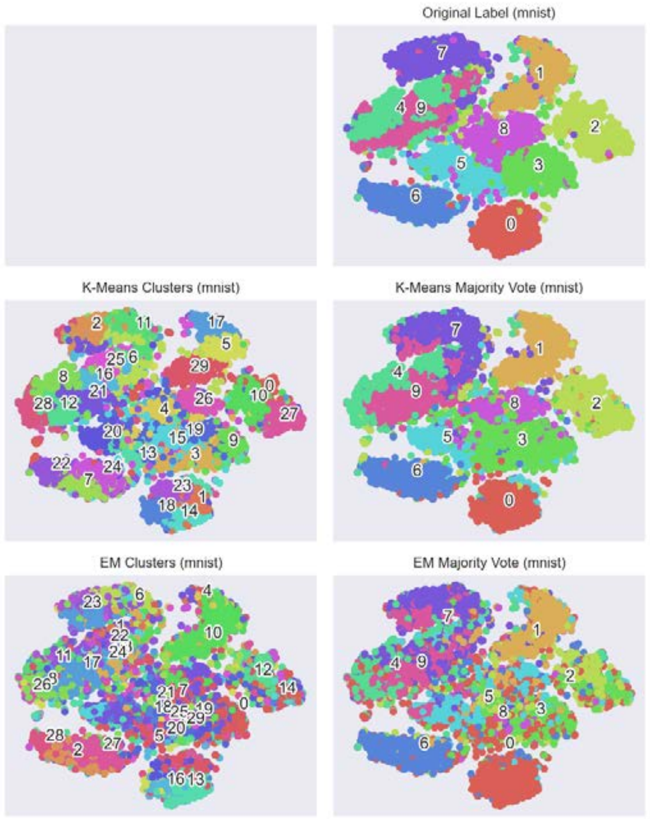

Fig. 4. TSNE dimension reduced clusters plot (Phish data).

Looking at the left two scatter plots, it is observed the K-Means groups the samples more distinctly than EM. Looking at the clusters of K-means and EM, it is observed that the clusters of K-means are more homogeneous than EM, whereas the clusters of EM is still blended.

Looking at the right three scatter plots, both K-Means and EM predicted the sample labels generally well. Comparing the predicted class labels of K-means and EM to true labels, it is astonishing how well the labels match true label: not only the label numbers match well, but also the label locations, especially K-means. This observation implies that our clusters algorithm works well for Mnist data. K-Means algorithm achieves a better match of true sample classification than EM, which agrees with that fact that K-Means has a higher accuracy score and homogeneity score than EM.

III. Part 2: Dimension Reduction Algorithm

In this section, I document four dim-reduction algorithms: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Random Projection (RPA) and Random Forest Classification (RFC).

A. *PCA*

According to Wikipedia, Principal Component Analysis (PCA) is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

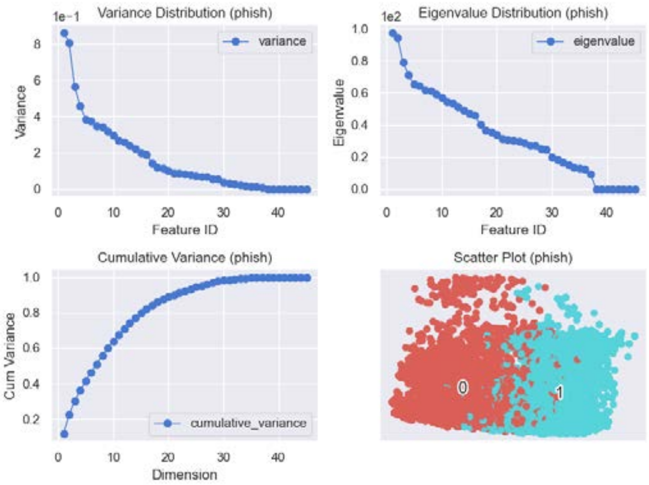

Fig. 5. PCA analysis for Phish data.

Figure 5 presents the PCA analysis for Phish data. Looking at the variance and eigenvalue distribution, it is observed that the variance and eigenvalue decrease exponentially with the component ID, which is well expected for PCA. The cumulative variance plot shows that the first 20 components explain roughly 90% of variation of Phish data. For PCA, dimension is determined so that 95% of cumulative variance is explained by top components.

The scatter plot of Figure 5 shows the scatter using the first two components. It is observed that the first two components make an excellent prediction of the classification of the samples, indicating that PCA works well for Phish data and the first two components capture the majority of the predictability of Phish features.

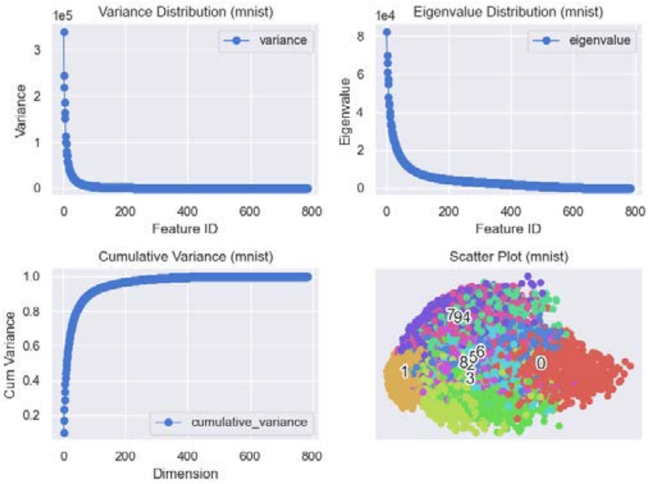

Fig. 6. PCA analysis for Mnist data.

Figure 6 presents the PCA analysis for Mnist data. Looking at the variance and eigenvalue distribution, it is observed that the variance and eigenvalue decrease exponentially with the component ID. After 30 components, there is not much variance and eigenvalue left for Mnist data. The cumulative variance plot shows that the first 30 components explain roughly 90% of variation of Phish data. The scatter plot shows the scatter of the first two components. It is observed that the first two components can distinguish samples with label 0 and 1, which is good enough, since Mnist has 784 features. If the first two components out of 784 can distinguish label 0 and 1, then using 30 principal components are expected to model the Mnist data much better than only use 2 components.

B. *ICA*

According to Wikipedia, Independent Component Analysis (ICA) is a computational method for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are non-Gaussian signals and that they are statistically independent from each other. Thus, kurtosis is important to determine the none-Gaussian of data and to evaluate ICA.

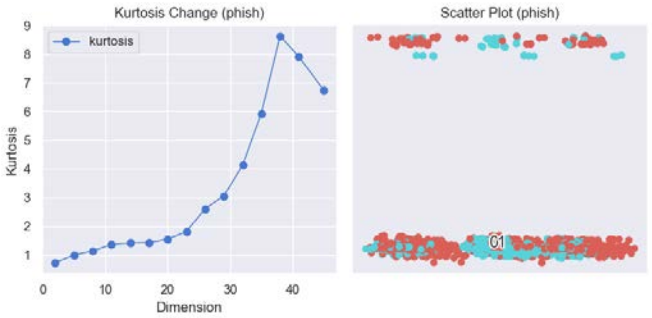

Fig. 7. ICA analysis for Phish data.

Figure 7 presents the ICA analysis for Phish data. The kurtosis reaches its highest value when dimension is 38, indicating that several features are not adding additional information from independent source, therefore they are not important. The scatter plot shows the scatter of the first two independent components. The scatter plot shows that the samples with different labels are all mixed up, indicating that using the first two components are not enough to group the Phish data into clusters.

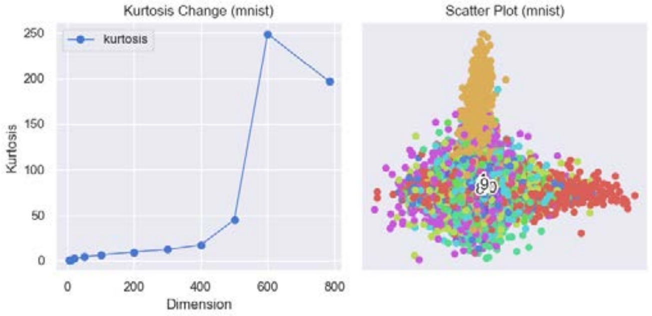

Fig. 8. ICA analysis for Mnist data.

Figure 8 presents the ICA analysis for Mnist data. The kurtosis reaches its highest value when dimension is 600, indicating that roughly 25% of features are not from independent source, therefore not important. The scatter plot shows the scatter of the first two independent components. The scatter plot shows that the samples with different labels are all mixed up, indicating that using the first two components are not enough to group the Phish data into clusters.

C. *RPA*

According to Wikipedia, random projection (RPA) is a technique used to reduce the dimensionality of a set of points which lie in Euclidean space. Random projection methods are known for their power, simplicity, and low error rates when compared to other methods. According to experimental results, random projection preserves distances well, but empirical results are sparse.

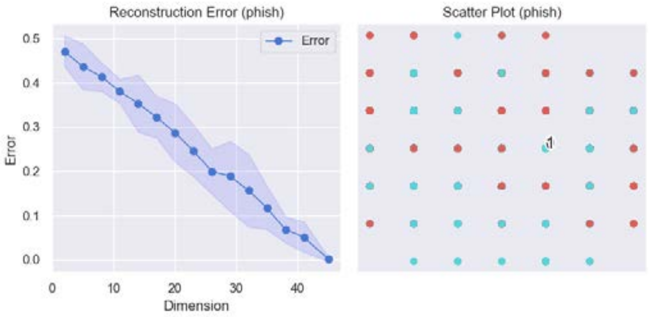

Fig. 9. RPA analysis for Phish data.

Figure 9 presents the RPA analysis for Phish data. Since RPA is random in nature, each dimension value in figure will run 10 times to calculate mean and standard derivation of reproduction error. Figure 9 shows that the more dimensions used by RPA, the smaller the reconstruction errors, indicating that all features of Phish data are important in the perspective of RPA. One notable observation is that the variation of reconstruction error is largest when dimension is around 30. As dimension further increase, the variation in error decreases.

In Figure 9, the scatter plot shows the scatter of the first two components of RPA. The scatter plot consists of distinct points, because we used only two RPA-reduced features and the features of Phish data is 0 or 1. The scatter plot shows that the samples with different labels are all mixed up, indicating that using the first two components are not enough to group the Phish data into clusters.

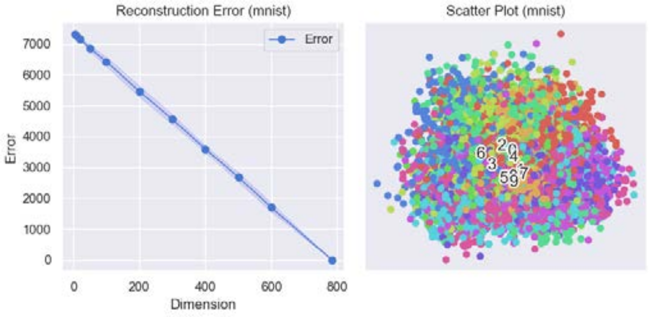

Fig. 10. RPA analysis for Mnist data.

Figure 10 presents the RPA analysis for Mnist data. Figure 10 shows that the more dimensions used by RPA, the smaller the reconstruction errors, indicating that all features of Mnist data are important in the perspective of RPA. One notable observation is that the variation of reconstruction error is largest when dimension is in the range of 200 to 600. As dimension further increase, the variation in error decreases.

In Figure 10, the scatter plot shows the scatter of the first two components of RPA. The scatter plot shows that the samples with different labels are all mixed up, indicating that using the first two components are not enough to group the Mnist data into clusters.

D. *RFC*

Random Forest classifier (RFC) is also one choice for dimension reduction, since this model uses decision trees to select features that are important to classify samples.

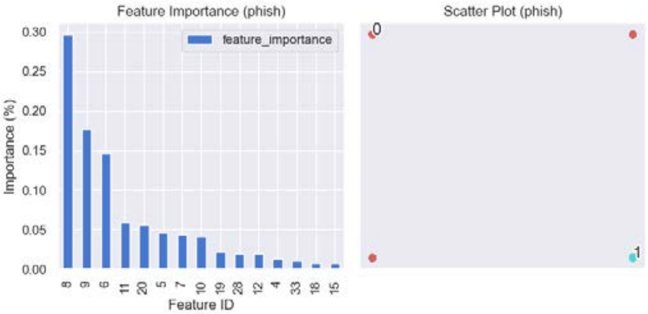

Fig. 11. RFC analysis for Phish data.

Figure 11 presents the RFC analysis for Phish data. In the feature importance distribution plot (the features are sorted in descending order according to their importance), it is observed that the importance of feature decrease exponentially based on RFC's evaluation. The features 8, 9, 6 are important features for Phish data, which is consistent with PCA, where the first two principal components are a good representation of Phish data.

The scatter plot of Figure 11 shows the scatter of the top two important features. Since the values of feature 8 and 9 are 0 and 1, the scatter plots only have four dots. According to this scatter plot, only the point of (1,0) of feature 8 and 9 belongs to class 1. The top features contain significant amount of information to distinguish or classify the samples of Phish data.

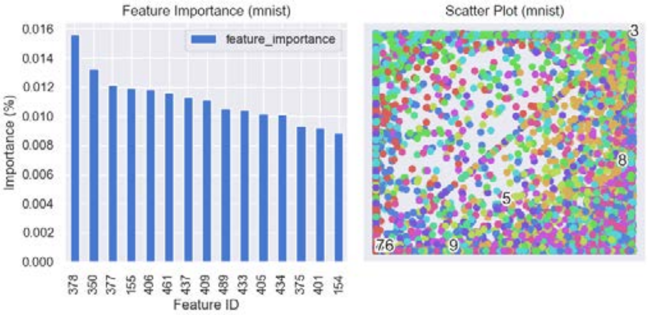

Fig. 12. RFC analysis for Mnist data.

Figure 12 presents the RFC analysis for Mnist data. In the feature importance distribution plot, it is observed that the importance of feature decreases linearly based on RFC's evaluation. This is different from Phish data, since Minst data has 784 features and no features significantly dominate other features.

The scatter plot of Figure 12 shows the scatter of the top two important features. Since no features dominate other features, only the top two features are not enough to distinguish or classify the samples. Thus, we observe that there is no pattern of clustering in the scatter plot and the labels are not quite meaningful for Mnist data.

IV. Part 3: Clustering on Dim-Reduced Data

In this section, I document apply K-means and EM algorithm to dimension reduced Phish and Mnist datasets. The dimension of Phish and Mnist data are reduced using the four methods of Part 2.

A. *Clustering of PCA-Recuded Data*

This section explores the effect of PCA dim-reduction on clustering algorithm: K-means and EM.

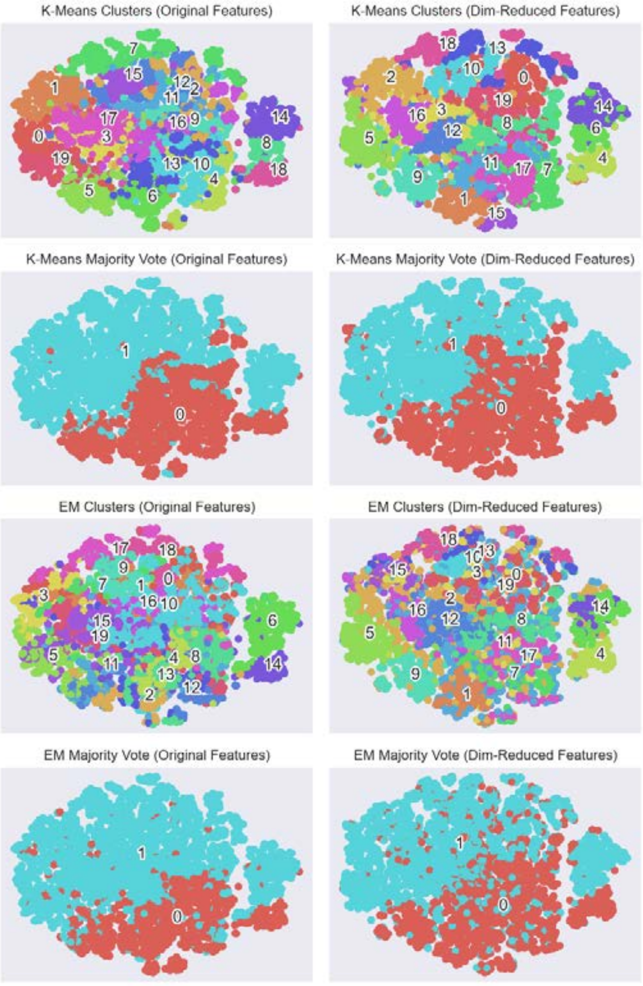

Fig. 13. Impact of PCA dim-reduction on clustering algorithm (Phish Data).

The number of features used in PCA is 26 for Phish data. The left column of plots in Figure 13 shows the results using original features. The right column of plots shows the results using PCA dim-reduced features.

In Figure 13, for K-means, comparing the first row, the clusters based on original data largely match the clusters from dim-reduced features. Looking at the second row, the predicted classification based original data matches the predicted classification based on PCA dim-reduced features, indicating that PCA does well for extract the principal components for Phish data.

In Figure 13, for EM, comparing the third row, the clusters based on original data largely match the clusters from dim-reduced features in the outer region, but the center region is mixed, which is due to EM, since EM does not do so well in group the samples in the center region. Looking at the Forth row, the EM predicted classification based original data matches the EM predicted classification based on PCA dim-reduced features.

Figure 13 shows that the PCA results are very close to original results. This is because PCA is orthogonal projection.

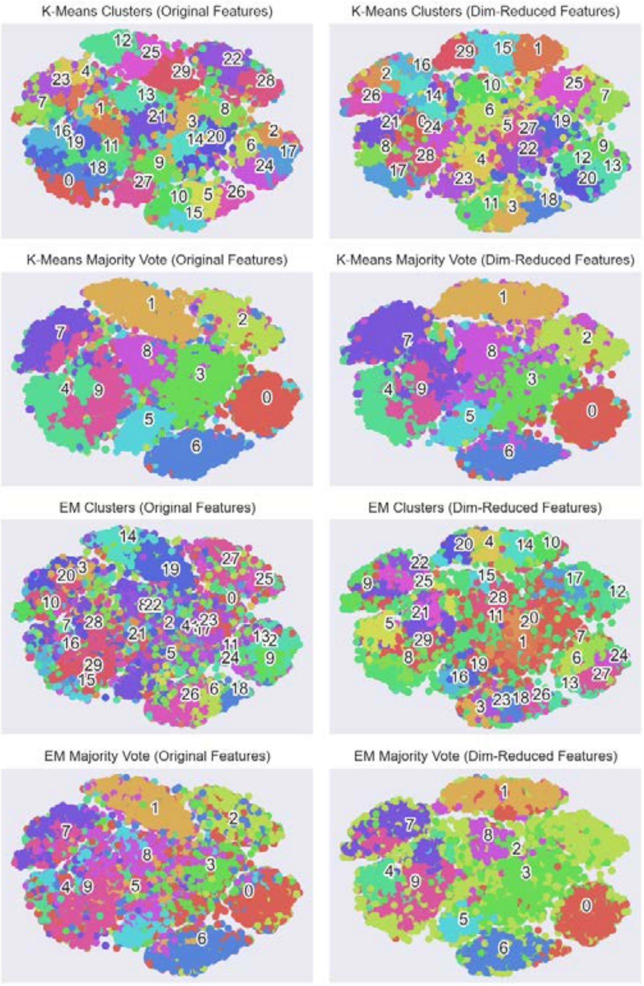

Fig. 14. Impact of PCA dim-reduction on clustering algorithm (Mnist Data).

The number of features used in PCA is 152 for Mnist data. In Figure 14, for K-means, comparing the first row, the clusters based on original data largely match the clusters from dim-reduced features. Looking at the second row, the predicted classification based original data matches the predicted classification based on PCA dim-reduced features, indicating that PCA does well for extract the principal components for Mnist data.

In Figure 14, for EM, comparing the third row, the clusters based on original data largely match the clusters from dim-reduced features. Looking at the fourth row, the EM predicted classification based original data matches the EM predicted classification based on PCA dim-reduced features.

B. *Clustering of ICA-Recuded Data*

This section explores the effect of ICA dim-reduction on clustering algorithm: K-means and EM.

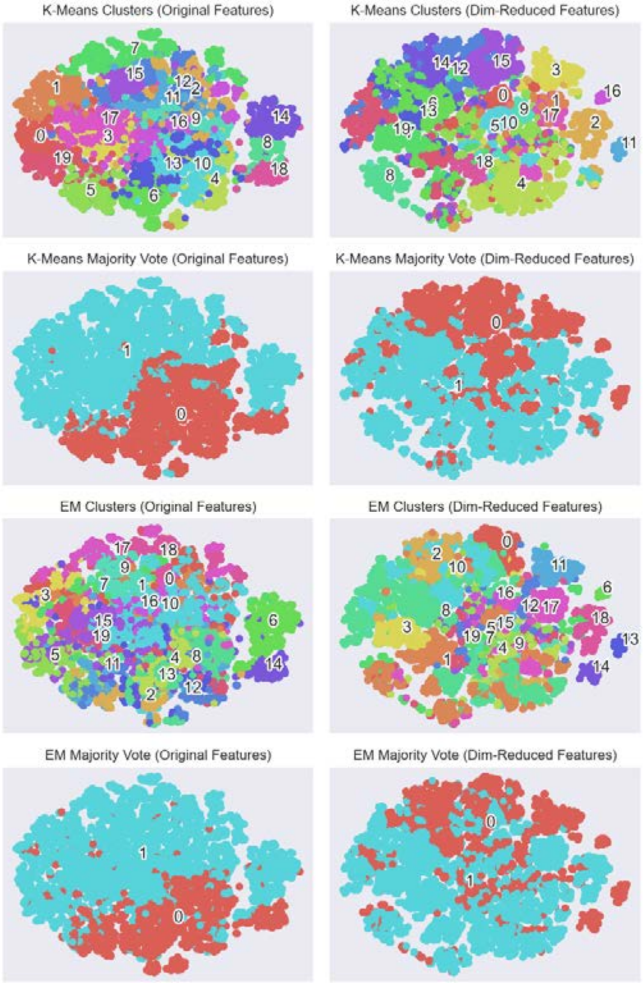

Fig. 15. Impact of PCA dim-reduction on clustering algorithm (Phish Data).

The number of features used in ICA is 38 for Phish data. In Figure 15, for K-means, comparing the first row, it is observed that the clusters in the center region of original features are more refined than those of ICA dim-reduced features, indicating that the ICA dim-reduced features ignore minor structure in the original features. Looking the second row, it is observed that the decision boundary changed significantly, which may be due to the fact that ICA projects the original axis to new axis.

In Figure 15, for EM, comparing the third row, it is observed that the clusters of original features are more refined than those of ICA dim-reduced features. Looking the fourth row, it is observed that the decision boundary changed significantly.

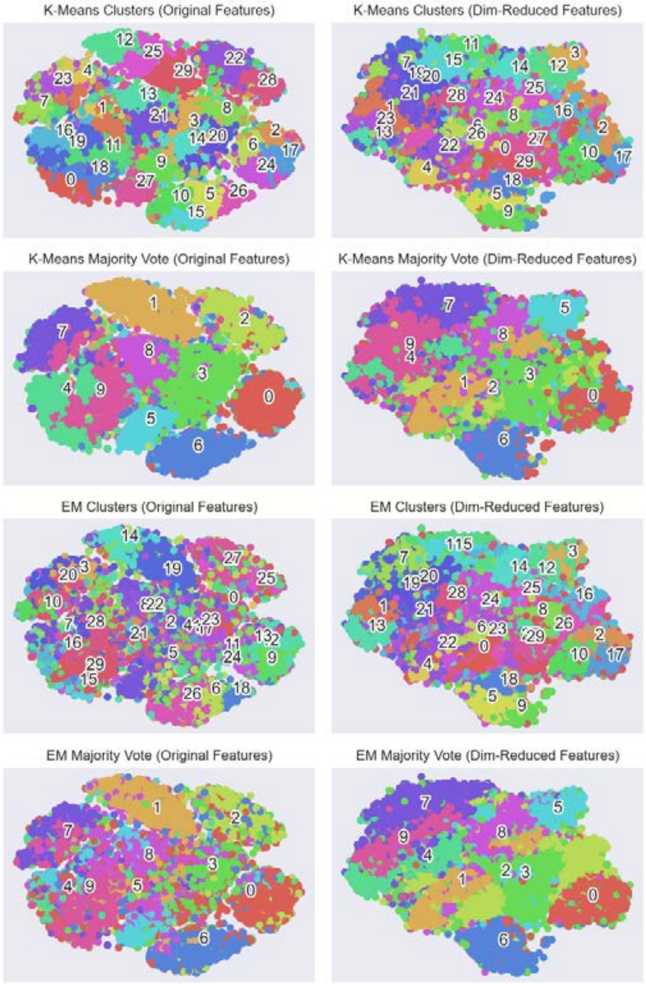

Fig. 16. Impact of PCA dim-reduction on clustering algorithm (Mnist Data).

The number of features used in ICA is 100 for Mnist data. In Figure 16, for K-means, comparing the first row, the shape of the clusters change from original to ICA dim-reduced features, but the pattern of clusters generally match. Looking at the second row, the predicted classification based original data matches the predicted classification based on ICA dim-reduced features.

In Figure 16, for EM, comparing the third row, the clusters of original features are more refined than those of ICA dim-reduced features. Looking at the fourth row, the shape of decision boundary is different, the classification pattern generally match, which is due to ICA projects features to new axis.

C. *Clustering of RPA-Recuded Data*

This section explores the effect of RPA dim-reduction on clustering algorithm: K-means and EM.

The number of features used in RPA is 18 for Phish data. In Figure 17, for K-means, comparing the first row, it is observed that the clusters in the center region of original features are more refined than those of RPA dim-reduced features, indicating that the RPA dim-reduced features ignore minor structure in the original features. Looking the second row, it is observed that the shape and direction of decision boundary changed significantly, which is because the RPA randomly projects the features to new axis.

In Figure 17, for EM, comparing the third row, it is observed that the clusters of original features are more refined than those of RPA dim-reduced features. Reducing the number of dimensions allows the cluster algorithms focus more on the structure in features. Looking the fourth row, it is observed that the decision boundary changed significantly.

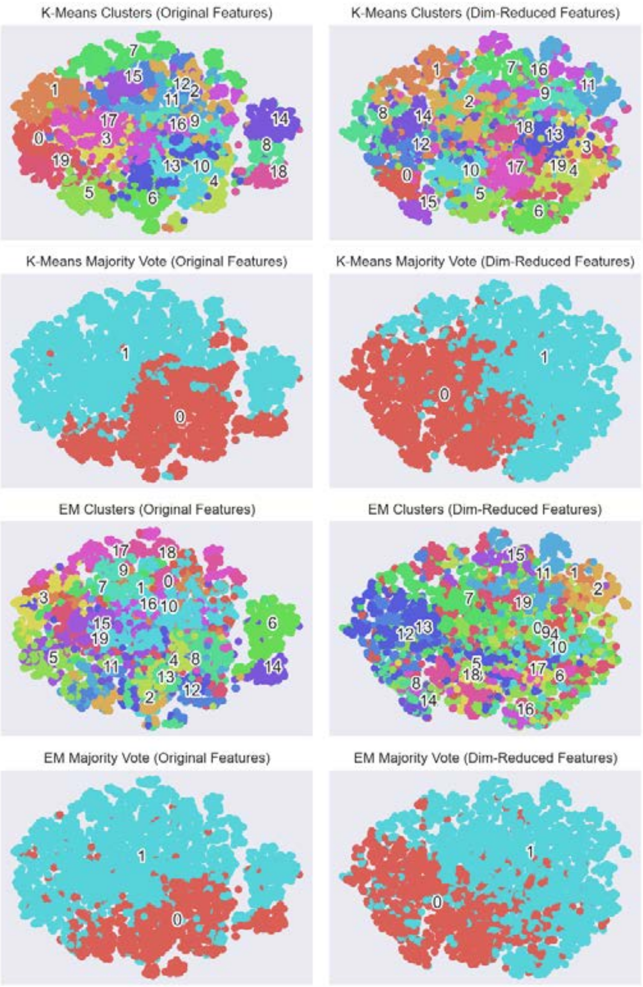

Fig. 17. Impact of RPA dim-reduction on clustering algorithm (Phish Data).

The number of features used in RPA is 78 for Mnist data. In Figure 18, for K-means, comparing the first row, the clusters based on original data largely match the clusters from RPA dim-reduced features. Looking at the second row, the predicted classification based original data matches the predicted classification based on RPA dim-reduced features, indicating that RPA does well for extract the main information in features for Mnist data.

In Figure 18, for EM, comparing the third row, the clusters of original features are more refined than those of RPA dim-reduced features. Looking at the fourth row, the EM predicted classification based original data matches the EM predicted classification based on RPA dim-reduced features, but the outlines of the classification boundaries are different, which is due to RPA projects features to new axis.

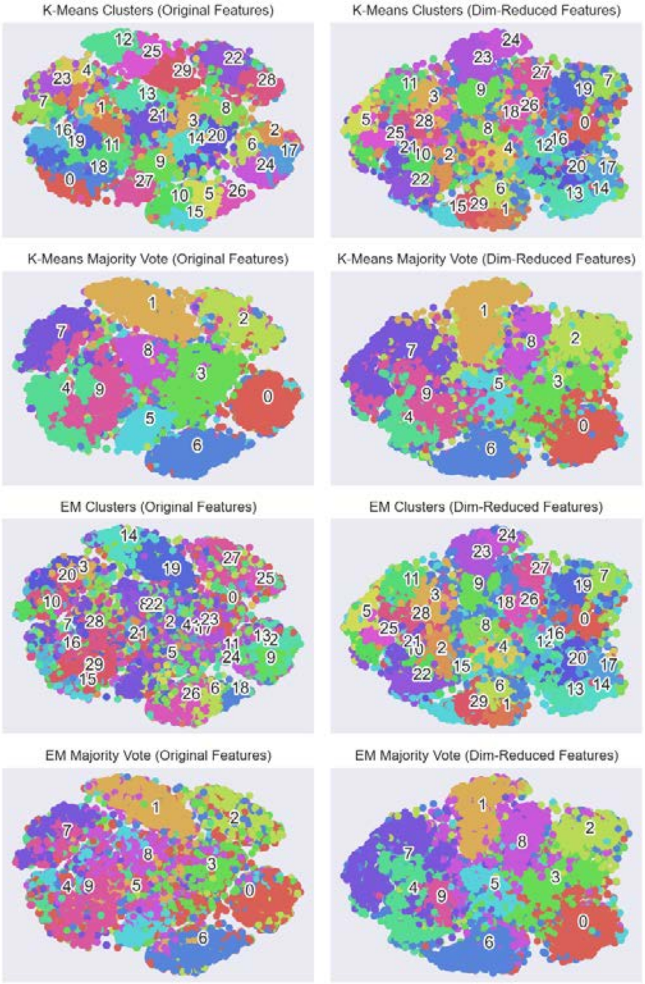

Fig. 18. Impact of RPA dim-reduction on clustering algorithm (Mnist Data).

D. *Clustering of RFC-Recuded Data*

This section explores the effect of RFC dim-reduction on clustering algorithm: K-means and EM.

The number of features used in RPA is 14 for Phish data. In Figure 19, for K-means, comparing the first row, the clusters based on original data are significantly different from the clusters from dim-reduced features. The RFC dim-reduced clusters are reduced to dots. This is because the selected Phish features by RFC are with values of 0 and 1. With only 14 features selected, the clusters are displayed as dots. Looking at the second row, the classification boundary are significantly different.

In Figure 19, for EM, similar observation is made as K-means clustering.

The number of features used in RPA is 243 for Mnist data. In Figure 20, for K-means, comparing the first row, the clusters based on original data largely match the clusters from RFC dim-reduced features. Looking at the second row, the predicted classification based original data matches the predicted classification based on RFC dim-reduced features, even though the locations of the classes are different, this is because with only 243 out of 784 features used, it is expected to see changes of location.

In Figure 20, for EM, comparing the third row, the clusters of original features are more refined than those of RFC dim-reduced features. Looking at the fourth row, the EM predicted classification based original data matches the EM predicted classification based on RFC dim-reduced features, but the outline of the classification boundaries are different.

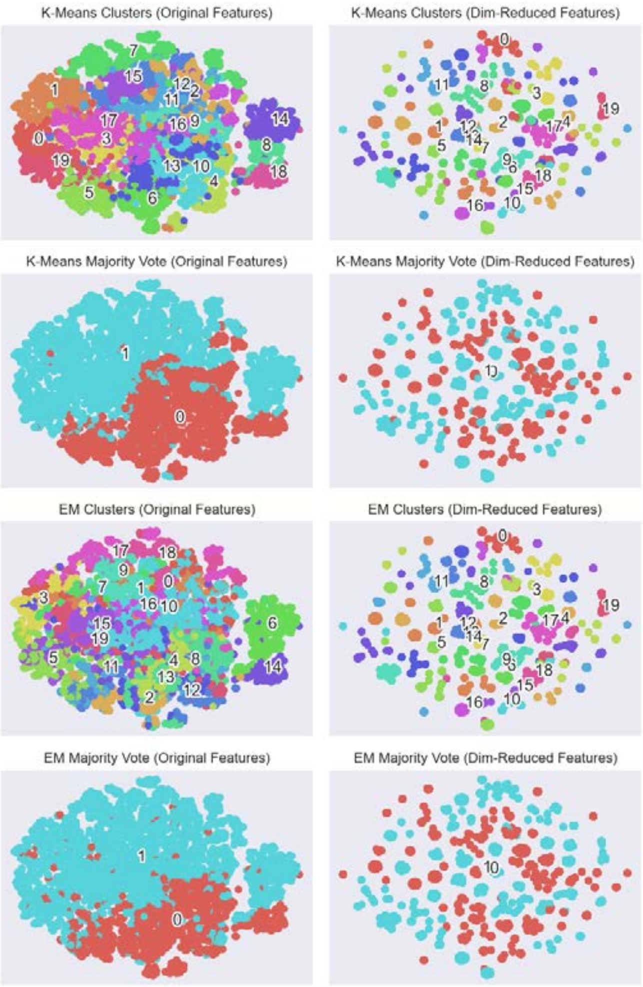

Fig. 19. Impact of RFC dim-reduction on clustering algorithm (Phish Data).

V. Part 4: ANN on Dim-Reducted Data

In this section, Artificial Neural Network is applied to dim-reduced data. For this section, I used Phish data from Assignment 1. In Assignment 1, it was determined that the ANN model with activation='logistic', hidden_layer_sizes=5, learning_rate_init=0.05 worked well for Phish data. The same hyper-parameters are used in this section.

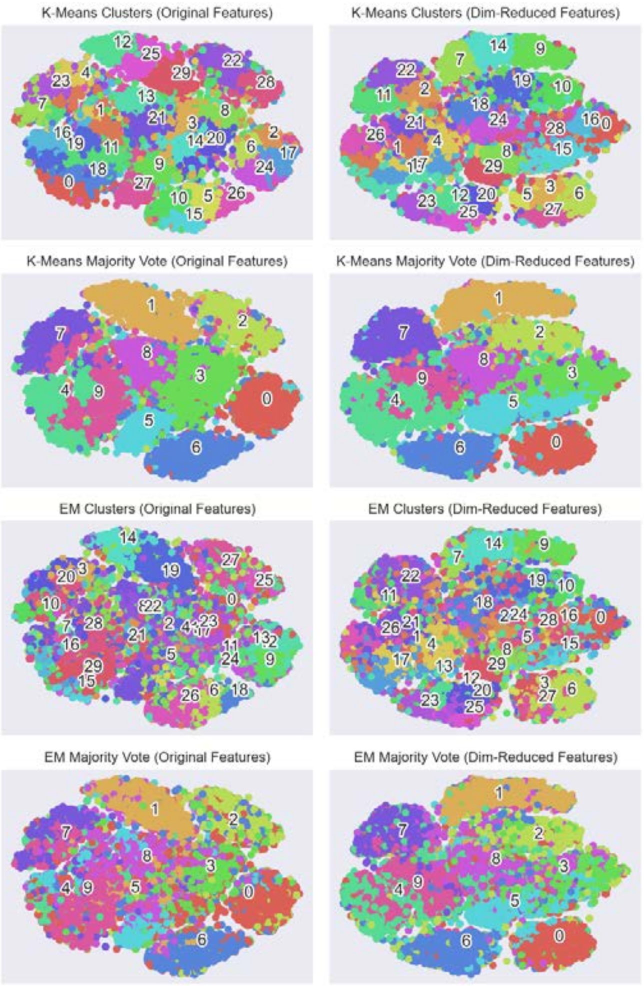

Fig. 20. Impact of RFC dim-reduction on clustering algorithm (Mnist Data).

A. *Performance comparison*

To compare the accuracy of results, I divided Phish data into 80% training data and 20% testing data. Then the f1_score, accuracy scores are calculated for training and testing data.

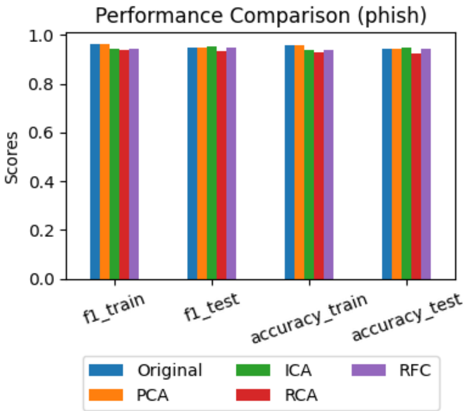

Fig. 21. Performance comparision for Phish Data.

Figure 21 compare the accuracy of five sets of data: original, PCA-, ICA-, RPA-, RFC-dim-reduced features. It is observed that both the training and testing scores are very close for the five sets of data. But ICA and RFC achieved slightly better f1 and accuracy test score. This might be because ICA and RFC extracted important features and reduced noise.

B. *Efficiency comparison*

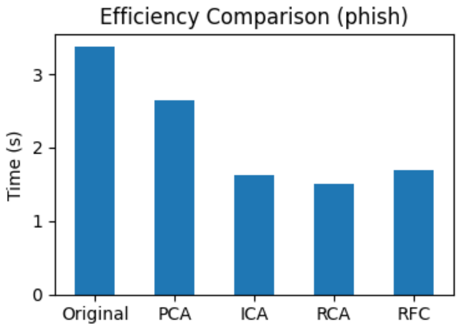

Fig. 22. Efficiency comparision for Phish Data.

Figure 22 compares the efficiency of ANN using five sets of data. It is observed that PCA reduced ANN training time by roughly 20%, whereas ICA, RPA and RFC reduced ANN training time by roughly 50%, which is a huge gain, considering the significant amount time need for training ANN.

VI. Part 5: ANN on Added Cluster Labels

In this section, Artificial Neural Network is applied to dim-reduced data plus the cluster labels generated by K-means and EM.

A. *Performance comparison*

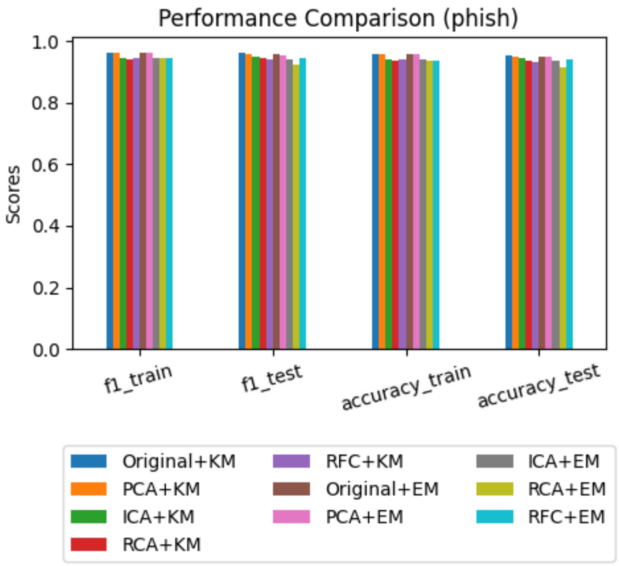

Fig. 23. Performance comparision for Phish Data.

Figure 23 compare the accuracy of ten sets data: (original, PCA-, ICA-, RPA-, RFC-dim-reduced features) x (K-means and EM generated labels). It is observed that both the training and testing scores are very close for the ten sets of data. Comparing Figure 23 to 21, it is observed that adding K-means and EM generated label features does not improve accuracy, which is well expected for two reasons: 1) the labels generated by K-means and EM are information extracted from features, but this extracted information already exists in original features; 2) ANN is capable of extracting all structural information if given enough samples, thus feature engineering or adding redundant information does not improve ANN performance.

B. *Efficiency comparison*

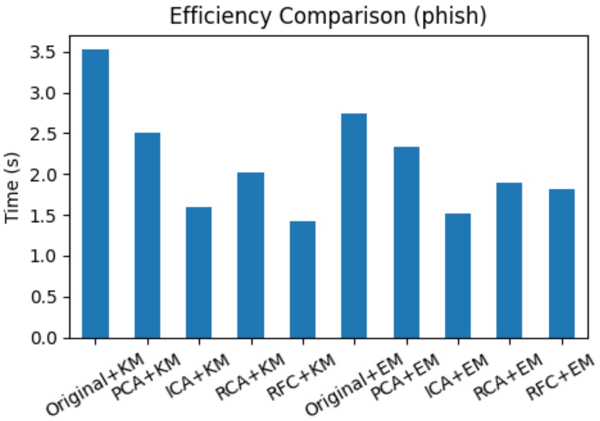

Fig. 24. Efficiency comparision for Phish Data.

Figure 24 compares the efficiency of ANN using ten different sets of data. Comparing Figure 24 to 22, it is observed that adding feature generally increase training time.

Conclusion and Future WOrk

In project, I conducted a thorough exploration of clustering and dim-reduction. It is found that the performance of K-means and EM really depends on the dataset. It is also important for EM to have a good PDF model to represent the sample features.

The performance of dimension-reduction algorithms depends on the underlying dataset. We should select the algorithm that really fits our problem. Dimension reduction can help us extract important features from original features, thus improve ANN efficiency, without hurting ANN accuracy.

Adding K-means and EM generated labels to features does not improve ANN accuracy but increases ANN training time. This is because ANN does not rely on feature engineering to perform well.

References

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_mutual_info_score.html

[2] https://en.wikipedia.org/wiki/Principal_component_analysis#:~:text=Principal%20component%20analysis%20(PCA)%20is,components%20and%20ignoring%20the%20rest.

[3] https://en.wikipedia.org/wiki/Random_projection