Modified K-means Clustering Algorithms for Feature Selection

By

Ayeasha Akhter

A thesis is submitted to the Faculty of Graduate Studies of

The University of Manitoba

In the fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba

Abstract—

Computational effort is difficult when dealing with high dimensional data that has hundreds or thousands of features. Features that don't significantly influence class predictions throughout the classification process increase the computing load. By eliminating unnecessary, redundant, or noisy features from the original features, feature selection, as a dimensionality reduction strategy, tries to pick a small subset of the important features from the original features. Two new feature selection methods are described in this study in relation to the effectiveness of k-means-based clustering methods. This research project aims to reduce the number of different features by clustering the D features into k (k < D) clusters, determining the cluster center to represent its members by finding the closest feature to the cluster center or selecting the highest weighted features among the cluster members, and performing feature selection. After removing 41.4% of the features from the VIRUS-MNIST dataset, we are able to deliver accuracy equivalent to the original dataset using both of our suggested methods in a shorter amount of time. Our proposed methods outperform sparse k-means, PCA, LLE, and wk-means-based feature selection method for clustering by ANN following feature reduction in the Wine dataset. With fewer features than the modified k-means feature selection method, our second method performs more accurately on the CNAE dataset.

Acknowledgments

I would like to thank and praise Almighty ALLAH for my life, unconditional love and, granting me to do M.Sc. at the University of Manitoba.

I would like to express my gratitude to Prof. Dr. Ken Ferens for his supervision in my research and his guidance during my M.Sc. study at the University of Manitoba. I would like to thank him from the core of my heart for being a great supervisor and an awesome person to work with.

I would like to thank my parents M. Ashraf Ali and Nasima Ashraf for inspiring me to do M.Sc. abroad and teaching how to keep faith in me. Also, for their numerous support, I am finally at the end of the program.

Finally, I would love to thank from the bottom of my heart to my husband Mohaimen Rahman due to his tremendous support, inspiration and motivation for completing the research program.

Dedicated to

I dedicate this work to my parents and my husband for their love and moral support. I also dedicate this thesis to my supervisor Professor Dr. Ken Ferens for his innovative ideas and supervision during the entire period of the master's program.

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

Recent years have seen an incredible growth in the amount of data available for several machine learning applications, including text mining, computer vision, and biomedicine. Computational effort is difficult when dealing with high dimensional data that has hundreds or thousands of features. Studying how to use these big data sets is crucial and vital for knowledge growth. Our attention is primarily drawn to the great dimensionality of the data. Existing machine learning techniques have been substantially challenged by the huge amount of high dimensional data. In addition to making learning algorithms extremely slow and even degrading the performance of learning tasks, the addition of noisy, redundant, and irrelevant dimensions can also make models difficult to interpret.

A common method for minimizing computational work in the processing of high dimensional data is feature selection. An algorithm known as a feature selection algorithm looks for the most important features in highly dimensional data. The purpose of feature selection is to decrease the size of the data file, remove irrelevant features, and identify the useful data features for data analysis. Data analysis might be affected by irrelevant features. Those features are effective features which is able to maintain the structure of the data properly such as the existing data clusters. During feature selection, there must be no changes to the data structure.

The K-Means popularity is rather evident just over 4,460,000 results for "K-Means" were found in a search on scholar.google.com in June 2023. In addition to these remarkable figures, this approach is implemented in many popular data analysis software programs, including SPSS, MATLAB, R, and Python. Because of its simplicity and applicability to many variables, k-means is one of the most effective approaches for clustering [18].

Two new feature selection methods are provided in this research with reference to the effectiveness of k-means-based clustering algorithms [19, 20, 21]. This research work aims to reduce the number of different features. This idea clusters the D features into k ($k < D$) clusters, determines the cluster center to represent its members by finding the nearest feature to the cluster center or choosing the highest weighted features among the cluster members, and thus performs feature selection. But in sparse k-means, we cannot explicitly determine the number of selected features but it can be done in our proposed methods.

Experimental results on VIRUS-MNIST dataset [12] show that, after reducing 41.4% features by using both of our proposed methods, we are able to provide accuracy similar to the original dataset with less time. According to experimental findings on the WINE dataset [14], our suggested approaches are more successful at clustering by ANN after feature reduction than sparse k-means, PCA, LLE, and the wk-means-based feature selection method [9]. From the experimental result it is also found that, the accuracy of our second method, for the CNAE Dataset [11] is higher with less features than modified k-means feature selection method of paper [10].

1.1. Thesis Statement and Overview

In this research, two new feature selection techniques are presented in relation to the efficiency of k-means-based clustering algorithms. The goal of this research is to minimise the number of features. By grouping the D features into k ($k < D$) clusters, this concept executes feature selection by the following two methods

- selecting the feature that is closest to the cluster centre
- selecting the feature with the highest weight among the cluster members

In our first proposed algorithm we classify our features into k groups by using the standard k-means clustering technique [22]. The initial four steps of our algorithm are identical to those of the standard k-means clustering algorithm. However, the fifth step of our technique differs from the standard k-means clustering algorithm because for feature selection we select a feature for each cluster that is closest to the cluster centre and this selected feature represents its whole cluster.

In the second proposed method, the features are initially divided into k clusters in accordance with the traditional k-means clustering algorithm. The initial four steps of our algorithm are identical to those of the traditional k-means clustering algorithm. After that we have modified the traditional k-means algorithm by calculating the weight of the features of each cluster such a way that it indicates how much a feature is similar to its own cluster compared to other clusters. Then for each cluster we will choose the feature with highest weight because that feature is the best fit for that cluster.

1.2.The contribution of the Thesis

Features that don't substantially influence class predictions throughout the classification process increase the computing load. To reduce the quantity of the high dimensional data and therefore the computational load, we have employed two different types of modified k means clustering methods in this study. The two proposed methods' experimental findings on three real datasets indicated that:

- For VIRUS-MNIST dataset, after reducing 41.4% features by using both of our proposed methods, we are able to provide accuracy similar to the original dataset with less time.
- For WINE dataset, our suggested approaches have better performance at clustering by ANN after feature reduction than sparse k-means, PCA, LLE, and the wk-means-based feature selection method [9].
- For CNAE Dataset, the accuracy of our second method is higher with less features than modified k-means feature selection method of paper [10].

1.3. Outline of the thesis

The structure of this thesis paper is as follows:

The concept of employing modified k-means clustering methods to decrease the features from a high-dimensional dataset is introduced in Chapter 1. In Chapter 2, a few literature reviews of the methodologies currently in use for feature selection methods are provided. Chapter 3 explains the history of a number of machine learning methods, such as the k-Means Clustering Algorithm, Artificial Neural Network, Decision tree, and k-NN Classifier. Chapter 4 provides a detailed description of the whole methodology used for this study, including dataset collections, modified k-means clustering, application of classifiers to datasets, models, and performance indicators, and k-fold cross validation. Chapter 5 covers the experimental work, analysis of the findings, and performance evaluation with other feature selection techniques. The conclusion of the thesis and the efforts to be tried in the near future to enhance the feature selection techniques utilized in this thesis paper are described in Chapter 6.

Chapter 2

2. Background Research

The dimensionality of the data used for machine learning and data mining tasks has grown explosively over the past thirty years. The curse of dimensionality [17], or data with extremely high dimensionality, has created significant challenges to existing learning methods [1].

Among practitioners, feature selection is a common strategy for minimising complexity. In order to improve learning performance, lower computational costs, and better model interpretability, it tries to choose a small subset of the relevant features out of the total set in accordance with some relevance evaluation criteria. [2, 13].

Feature selection techniques can be categorised in various ways [2]. The classification into filters, wrappers, embedded, and hybrid techniques is the most popular. Without using a mining algorithm, the filter model evaluates and chooses feature subsets based on the general traits of the data. The performance of the required mining algorithm serves as the assessment criterion in the wrapper model. In an effort to increase mining performance, it looks for features that are better suited to the mining algorithm, but it is also typically more computationally costly than the filter model. By utilising the two models' various evaluation criteria at various phases of the search process, the hybrid model aims to combine the best features of the two models.

Each of the aforementioned feature decision categories has drawbacks [3][4]. The disadvantage of using filters to select features is that the objective function (classifier) is not used to decide

which features are most crucial. As a result, it's possible that the results of the objective function on the features that were chosen are different from the results of the objective function on the dataset's original features. The picking of wrapper features has the drawback of having a high computational complexity and a lengthy execution time. The disadvantage of embedded feature selection is that it cannot be generalised for various types of classification models because it relies so heavily on the type of classification model.

Principle component analysis (PCA) is one technique for extracting features [5]. A feature with a variance close to zero will be viewed as a trivial feature or noise in this technique and a feature with the maximum variance will be the best extracted feature.

During feature reduction, there must be no changes to the data layout or data clusters. The features that reduce the data's intra-cluster distance when compared to the pair-wise distance between pairs of data are chosen in sparse k-means [6, 7] in order to preserve data clusters as much as feasible. Unfortunately, this method does not allow the user to specify how many features they want to pick.

Locally linear embedding (LLE) [8], a different feature extraction technique, has two phases: The coefficients of each data set are stored after each data set is written as a linear combination of other data in the first step. The following step is to construct low-dimensional data so that each low-dimensional data may be represented as a linear combination of the other low-dimensional data using the same coefficients that were recorded in the prior stage.

In article [9], a weighted k-means (wk-means) feature reduction technique is presented. In this method, the weights of the data features are first calculated using the wk-means method. The remaining features are then chosen after a set of weighted features with the least impact on data clusters is eliminated using a novel mathematical model. The accuracy of the clusters produced by the proposed method's wk-means after feature reduction is superior to that of sparse k-means, PCA, and LLE, but the speed of the proposed method is slower than that of PCA and sparse k-means, according to experimental findings on four real datasets.

In paper [10], it was discovered that K-means clustering can be used to reduce the number of redundant features because it tries to collect related characteristics into a single cluster. One representative feature could symbolise a group of similar features. Furthermore, because the centroid is calculated from the mean of all the features in the cluster, the value of those features that are part of the same cluster can be expressed by the centroid's value. It's interesting that high class prediction accuracy is obtained when using cluster centroids as representatives of the entire feature collection.

Chapter 3

3. Background of Machine Learning Algorithm

The history of computational intelligence methods including the K-means algorithm, artificial neural networks, decision trees, and k-nearest neighbor classifier is presented in this chapter.

3.1. K-Means Algorithm

In order for members of the same cluster to be relatively similar to one another and for members of other clusters to be substantially distinct from one another, objects are grouped into meaningful subclasses using the clustering method. The partitioning algorithm, hierarchical algorithm, density-based algorithm, and grid-based algorithm are the four major categories into which clustering algorithms can currently be divided. A database of N objects is divided into a set of K clusters by partitioning techniques [33,34]. Most frequently, they begin with an initial partition and then optimise an objective function using an iterative control technique.

A. The idea of algorithm

Given the D-dimension data set

$X = \{ x_i \mid x_i \, \varepsilon \, R^D , \, i = 1,2,3....N \}$. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters $W_1, W_2.....W_k$, the main idea is to define k centroids: $C = c_1 , c_2 .... c_k$, one for each cluster,

$$c_i = \frac{1}{n_i} \sum_{x \in w_i} x \qquad (1)$$

where n_i is the number of datasets in the cluster [35]. Next, each point from a particular data collection is taken and connected to the closest centroid. The first step is finished and an early groupage is finished when there are no points still open. At this stage, the centroids of the clusters produced by the preceding phase need to be recalculated as their barycenters. The identical data set points must now be bound to the closest new centroid once we obtain these new centroids. There has been created a loop. This loop causes the centroids to gradually shift their positions until no further modifications are made, as we could see. Therefore, centroids are no longer in motion. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d_{ij} \, (x_j, c_i) \qquad (2)$$

Where $d_{ij}(x_j, c_i)$ is a distance measure (Euclidean distance) between a data point X_j and the cluster center C_i, is an indicator of the distance of the data points from their respective cluster centers.

B. The steps of algorithm

Step 1 (Initialization): Randomly chose k instances c_1 , c_2 …. c_k from the dataset X and make them initial cluster centers of the clustering space;

Step 2 (Assignment): Assign each instances to its closest center: if $d_{ij}(x_j, c_i) < d_{im}(x_i, c_m)$, where *m = 1,2,… …,k; j≠ m; i-1, ……, n*, and then assign x_i to cluster c_i;

Step 3 (Updating): Recalculate the centroids of the clusters: $c_1^*, c_2^*, \ldots\ldots\ldots, c_k^*$;

Step 4 (Iteration): If i ∈ $\{1,2,\ldots\ldots,k\}, \in c_i^* = c_i$, then end the algorithm, and the current $c_1^*, c_2^*, \ldots\ldots\ldots, c_k^*$ represents the final cluster, otherwise assign $c_i^* = c_i$, and repeat steps 2 and 3 until there is no more updating.

3.2.Artificial Neural Network (ANN)

The ANN is a forward structure black-box model that is trained through back propagation, a supervised training method. The ANN functions similarly to the human nervous system or brain and has many connections between its nerve cells and other axons [24]. Even with faulty, complicated, or insufficient input data, it is still possible to draw conclusions that are relevant by using examples. It can thus fully duplicate the human nervous system. Only one input layer, one output layer, and at least one hidden layer is present in the ANN, though. Neurons that resemble brain nerves form each layer (Fig. 1). These neurons are composed of nonlinear processing units. Networks are created by connecting the neurons in each layer to the neurons in the layers above and below them as well as to one another. The connections between neurons in following levels are also given weight. Information transfer from one neuron to another or from one layer to another is referred to as forward connection. This automatic learning is accomplished by a dynamic alteration of the network connections linked to each neuron [23].

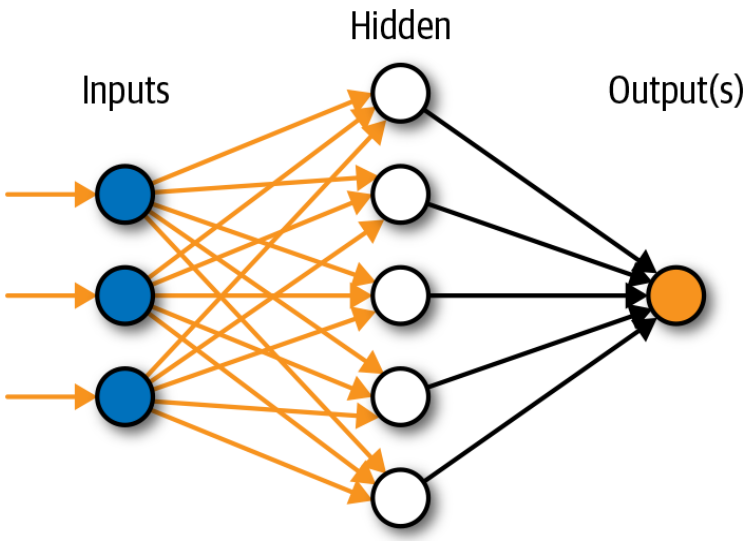

Fig. 3.1: Artificial Neural Network

One of the most important techniques that ANN commonly employs is the back propagation algorithm, a gradient-decent approach. Its main goal is to decrease the error between the actual network outputs and the outputs of the training input/output pairs [25]. On a regular basis, the network receives a large number of input/output pairs, and the output layer eventually transmits the error back to the input layer. The learning rate and update rule are used to update the weights of the backward paths [26,27]. Additionally, the default processing unit, training rate, or learning rate cannot be used to uniquely specify the ANN. Therefore, modifying model parameters by trial and error is the only effective way to achieve better results.

Mathematical justification for the MLP architecture is possible. In MLP architecture, the input layer comprises the n_0 neurons, which collect a normalized set of input variables of x_i (i = 1, 2 . . . . . n_0). The second layer is also known as the hidden layer that contains the n_1 neurons and receives a set of variables of y_j (j = 1, 2 . . . .n_1), which are the output of the first layer. In each of the neurons that correct their outputs, each layer receives a bias value of 1. The third or output layer consists of the n_2 neurons with number equal to output variables of z_k (K = 1, 2 . . . ., n_2). A

continuous non-linear mapping is performed in the n_0 neurons of x_i variables in the output layer to the y_i variables in the hidden layer after summing them up using an activation function. The weights of the neurons in each hidden layer for each outcome of the neurons in the input layer are another definition of the parameter for this function [32]. The back-propagation algorithm, which is defined by minimising the cost function as shown in Equation (3), is one of the most used techniques for ANN training.

$$m = \frac{1}{2} \sum_{i=1}^{n} (a_i - b_i)^2 \qquad (3)$$

where n represents the number of classes, a_i denotes the expected output, and b_i is the response of designed ANN from the i neuron of the total n neurons in the output layer.

3.3.Decision Tree

One of the most well-known machine learning algorithms is Quinlan's decision tree classifier [28]. Decision Tree algorithms were applied in a variety of industries and applications. These algorithms can be used to extract text, find data in sectors that require medical certification, substitute statistical techniques, and search engines. Various decision tree algorithms, including ID3, C4.5, CART, and others, have been developed based on their accuracy and cost-effectiveness.

Decision nodes and leaf nodes are the components of a decision tree. individually decision node has a number of branches that individually deal with the test X findings. Each decision node is a test X over a single attribute of the input data. Each leaf node is a class that represents the conclusion of a case's decision. A divide and conquer strategy is primarily used while creating a decision tree [28]. Fig. 2 shows a decision tree's construction.

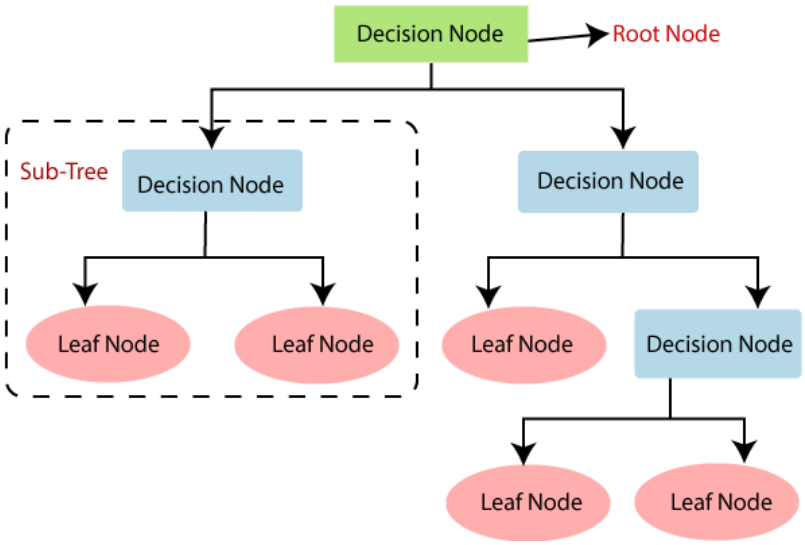

Fig. 3.2: Decision Tree

A set T of training data consists of k classes (C_1, C_2, ... C_k). If T only consists of cases of one single class, *T* will be a leaf. If *T* contains no case, *T* is a leaf and the associated class with this leaf will be assigned with the major class of its parent node (this is the choice of C4.5). If *T* contains cases of mixed classes (i.e., more than one class), a test based on some attribute a_i of the training data will be carried and *T* will be split into *n* subsets (T_1, T_2, ..., T_n), where *n* is the number of outcomes of the test over attribute a_i. The same process of constructing decision tree is recursively performed over each T_j, where $1 \leq j \leq n$, until every subset belongs to a single class.

How to pick the ideal attribute for each decision node when building the decision tree is the issue at hand. Gain Ratio Criterion is the criterion C4.5 selects. The basic concept of this criterion is to select a characteristic at each splitting stage that yields the most information gain while decreasing the bias in favour of tests with numerous results through normalization.

A decision tree can be used to categorise testing data that shares the same attributes as the training data after it has been constructed. The test is run on the same property of the testing case that the root node represents, starting at the decision tree's root node. The branch whose condition is satisfied by the value of the tested attribute is chosen by the decision-making process. This

branch guides the choice to a root node's child. Up until a leaf node is reached, the same procedure is repeated. The test case's provided class is connected to the leaf node.

3.4. K-Nearest Neighbour (KNN) Classifier

One of the most popular classification methods is the k-nearest neighbour (k-NN) technique since it is straightforward and simple to apply. A common non-parametric classifier that has been applied as the default classifier in many pattern classification applications is the k-nearest neighbour (k-NN) classifier [29]. To determine the final classification result, the distances between the test data and each of the training data are measured.

This kind of lazy learning algorithm does not require off-line training. The k-NN algorithm directly searches through all of the training examples for a given testing example during the classification stage by calculating the distances between the testing example and all of the training data in order to identify its closest neighbours and generate the classification output [30]. The k-NN classifier computes the distances between the point and points in the training data set to categorise an unknown instance represented by some feature vectors as a point in the feature space. The distance metric most frequently employed is the Euclidean distance. The point is then distributed among the class's k closest neighbours (where k is an integer). Fig. 3 illustrates this concept where * represents the point. If k = 1, the point belongs to the dark square class; if k = 5, the small circle class which are the majority class of the five nearest points. As k-NN does not require the off-line training stage, its main computation is the online ‘searching’ for the k nearest neighbours of a given testing example. Although using different k values are likely to produce

different classification results, 1-NN is usually used as a benchmark for the other classifiers since it can provide reasonable classification performances in many pattern classification problems [31].

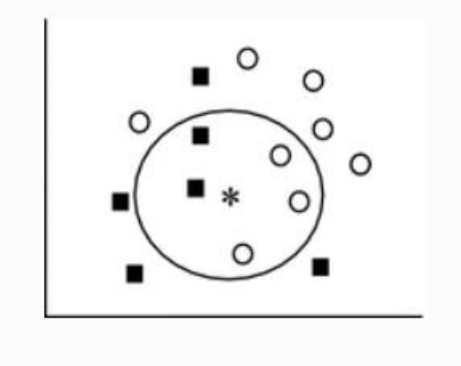

Fig. 3.3: k-Nearest Neighbor Classifier

To measure the distance between points *A* and *B* in a feature space, various distance functions have been used in the literature, in which the Euclidean distance function is the most widely used one. Let *A* and *B* are represented by feature vectors $A = (x_1, x_2, \ldots, x_m)$ and $B = (y_1, y_2, \ldots, y_m)$, where *m* is the dimensionality of the feature space. To calculate the distance between *A* and *B*, the normalized Euclidean metric is generally used by Equation (4).

$$dist\,(A, B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \qquad (4)$$

Chapter 4

4. Methodology

The methodology has been divided into the following three steps:

1. Dataset Collection
2. Performing Proposed Algorithms
3. Applying Classifier on Datasets

4.1.Dataset Collection

In this research we have collected the dataset from the paper called “VIRUS-MNIST: A BENCHMARK MALWARE DATASET” [12] which has 48,422 data points containing 1024 features. This dataset provides class label information of each data point. This dataset consists of 10 different classes. Among them 9 are malicious and one is benign.

Table 1: Class distribution and example types for Malware and Beneware data.

Class	Count	Group	Type
0	2516	Beneware	Good
1	7684	Malware	Adware
2	3037	Malware	Trojan
3	2404	Malware	Trojan
4	796	Malware	Installer
5	6662	Malware	Backdoor
6	15377	Malware	Crypto
7	7494	Malware	Backdoor
8	2571	Malware	Downloader
9	3339	Malware	Heuristic

4.2. Performing Proposed Modified K-means Clustering Algorithms

In order to minimize the quantity of high dimensional data, two new distinct types of modified k-means clustering algorithms are suggested in this study. These approaches seek to provide two reduced datasets that only include necessary characteristics. Modified k-means clustering methods are used in this stage to partition the original dataset's characteristics into a predetermined number of groups. The clustering algorithms accept as input parameters the original dataset and the target number of clusters. In this study, k-means clustering techniques are used to create clusters of characteristics with comparable values across data points. Assume that the dataset comprises N features and D features in total. When this dataset is subjected to k-means clustering, the goal is to create a feature set of size k, where k D, that is representative of the entire feature set.

The division is carried out horizontally since the clustering method creates clusters from the characteristics of the original data. The original dataset is collected, and after being converted into its transpose matrix, it is fed into the modified K-means clustering algorithms.

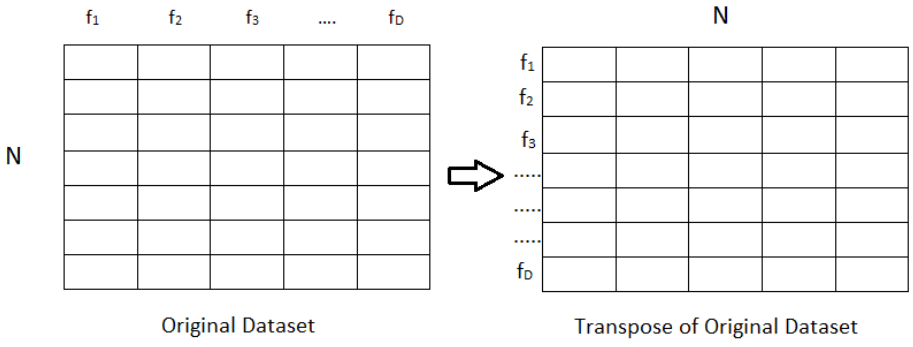

Fig 4.1: Transform original data into its transpose matrix.

In this research, we have proposed two modified K-means clustering algorithm for feature reduction:

1. Modified K-means clustering Algorithm (Based on Distance)
2. Modified K-means clustering Algorithm (Based on Weight)

4.2.1. Modified K-means clustering Algorithm (Based on Distance)

We partition our features into k groups in our first proposed algorithm by using the traditional k-means clustering technique [22]. The initial four steps of our algorithm are identical to those of the traditional k-means clustering algorithm. However, the fifth step of our technique differs from the traditional k-means clustering algorithm because for feature selection we select a feature for each cluster that is closest to the cluster centre and this selected feature represents its whole cluster.

Algorithm Steps:

Step 1: Initialize k centroids to random values.

Step 2: Each training example is assigned to one of the k centroids (clusters) by computing its Euclidean distance from each of the k-centroids and choosing the nearest centroid.

Step 3: Update the centroid to the average of its members.

Step 4: Repeat step-2 and step-3 or stop at convergence.

Step 5: At convergence, for each centroid (cluster), calculate the distance between the centroid and its each member to find the closet feature to this centroid. Do the same thing for all the centroids and thus k features will be found.

4.2.2. Modified K-means clustering Algorithm (Based on Weight)

It is believed that feature weighing is a generalization of feature selection [15]. However, feature weighting gives every feature a value, typically in the range [0,1] or [-1,1]. The feature will be more apparent the higher this number is.

In the second proposed method, the features are initially divided into k clusters in accordance with the standard k-means clustering algorithm [22]. The initial four steps of our algorithm are identical to those of the standard k-means clustering algorithm. After that we have modified the standard k-means algorithm by calculating the weight of the features of each cluster using equation (5) such a way that it indicates how much a feature is similar to its own cluster compared to other clusters. The weight of the feature ranges between [-1,1]. Then for each cluster we will choose the feature with highest weight because that feature is the best fit for that cluster and this selected feature represents its whole cluster.

Algorithm Steps:

Step 1: Initialize k centroids to random values.

Step 2: Each training example is assigned to one of the k centroids (clusters) by computing its distance from each of the k-centroids and choosing the nearest centroid.

Step 3: Updated the centroid to the average of its members.

Step 4: Repeat step-3 and step-4 or stop at convergence.

Step 5: At convergence, we will compute the weights of all the features of each cluster by using the equation given below:

$$w(i) = \frac{b(i) - a(i)}{\max\left(b(i), a(i)\right)} \qquad (5)$$

Referring to (5):

a(i) is the average distance of that feature with all other features in the same clusters.

b(i) is the average distance of that feature with all the features in the closest cluster to its cluster.

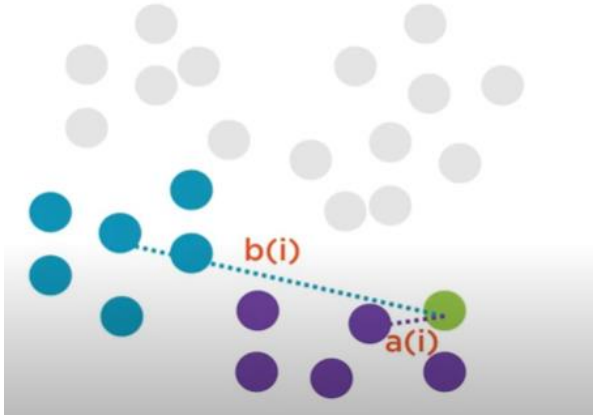

Fig 4.2: Diagrammatic representation of a(i) and b(i) from the above-mentioned formula to compute the weights w(i)

Step 6: For each centroid (cluster) we will choose a feature with the highest weight. Do the same thing for all centroids and thus k features will be found.

The objective of this research work is to reduce the features through modified K-means Clustering algorithm and maximize the classification accuracy and minimize the execution time. To choose the right value of K and to get better classification accuracy both the simulation processes have been done several times with 7 different values of K. After performing those two algorithms, we have found 7 different reduced datasets for both the algorithms.

Table 2: Seven different values of k.

Number of Reduced Features (k)
50
100
200
400
500
550
600

4.3. Applying Classifier on Datasets

Now, we must perform a machine learning classifier for both the D features dataset (actual dataset) and the reduced k features datasets (found from the proposed algorithms). For each training example, the classifier will classify them as normal or malware program for all those three data sets. After that, we will have three sets of results. Now we will compare the performance of D-features and k-features data sets. Performance will be measured by the accuracy, f1 score and the execution time needed for classification.

4.4. Models and Performance

Accuracy, precision, recall, and F1-score of the two models' performance were examined. One of the most used performance measures is accuracy. It is the proportion of observations that were successfully predicted to all of the observations.

$$\textit{Accuracy} = \frac{\textit{True Positive} + \text{True Negative}}{\textit{True Positive} + \textit{False Positive} + \textit{True Negative} + \textit{False Negative}} \quad (6)$$

The percentage of correctly identified positive instances among all projected positive cases is the implicit definition of precision. It deals with how well your model can forecast the real positives.

$$\textbf{\textit{Precision}} = \frac{\textbf{\textit{True Positive}}}{\textbf{\textit{True Positive + False Positive}}} \qquad (7)$$

The percentage of correctly identified positive cases among all positive instances is known as recall.

$$\textbf{\textit{Recall}} = \frac{\textbf{\textit{True Positive}}}{\textbf{\textit{True Positive + False Negative}}} \qquad (8)$$

The harmonic mean of the model's accuracy and recall is known as the F1-score, which is a method of combining the model's precision and recall.

$$\textbf{\textit{F1 Score}} = 2 * \frac{\textbf{\textit{Precision}} * \textbf{\textit{Recall}}}{\textbf{\textit{Precision}} + \textbf{\textit{Recall}}} \qquad (7)$$

4.4. K-fold Cross Validation

Last but not least, we used K-fold cross validation on the dataset to test the effectiveness of our findings and assure the model's generalizability. The K-fold cross validation method is one way to evaluate how well the model works with fresh data [16]. At some stage in the K-Fold cross validation process, a given data set is separated into K portions, or folds, and each fold is utilised as a testing set. Consider the 5-Fold Cross Validation (K=5) situation. The data collection is divided into 5 folds in this case. The first fold is used in the first iteration to evaluate the model, and the remaining folds are used to train the model. The second version uses the second fold as

the testing set and the remaining data as the training set. Up until all five of the folds have been used as the testing set, this procedure is repeated.

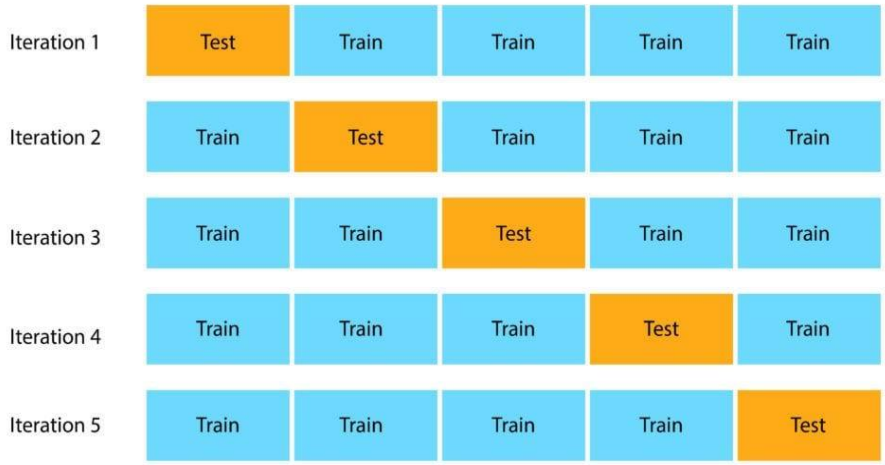

Fig 4.3: K-fold simulation process

Chapter 5

5. Experiments and results

This section consists of four different types of experiments and results. In the first experiment, modified k-means clustering algorithms were performed on the VIRUS-MNIST dataset [12], ANN classifier was applied to compare performance and finally, K-fold cross validation was applied to check the generalizability of both algorithms. In the second experiment, the performance of these two proposed methods have been compared with the four other feature

selection methods [5,7,8,9] through WINE dataset [14]. In the third experiment, the proposed modified K-means algorithms were compared to another modified k-means algorithm [10] through CNAE dataset [11]. Finally in the fourth experiment, the intersection of feature selection of two proposed methods for those three datasets is showed.

5.1. Experiments & Results

This experiment was done through the following steps:

1. Choosing classifier
2. Applying ANN Classifier on actual dataset and new datasets
3. Comparing performance between actual dataset and new datasets
4. Applying K-fold cross validation on two new datasets

5.1.1. Choosing Classifier

To verify how valid our two proposed algorithms are we must perform a machine learning classifier. To choose the best suited classifier for this dataset, three different machine learning classifiers have been applied to the original dataset which includes K-nearest neighbor, Decision tree and Artificial Neural Network. From Table 3, we can see that, among the three classifiers the Artificial neural network showed better f1 scores than others.

Table 3: F1 scores of ANN, decision tree and KNN by using original VIRUS-MNIST dataset.

Class	Group	F1 Score For ANN	F1 Score For Decision Tree	F1 Score For KNN
0	Beneware	**21.0%**	19.1%	11.0%
1	Malware	**99.7%**	98.8%	99.4%
2	Malware	**86.2%**	62.5%	90.6%
3	Malware	**93.0%**	90.2%	94.8%
4	Malware	**100%**	99.8%	100%
5	Malware	**87.1%**	74.2%	58.6%
6	Malware	**94.5%**	83.1%	81.5%
7	Malware	**85.7%**	67.0%	71.0%
8	Malware	**85.2%**	63.1%	78.2%
9	Malware	**90.4%**	75.7%	84.4%

In Fig 6 and Fig 7, the average accuracy and execution time of the 3 different classifiers are shown. If we look at the overall accuracy and execution time of these 3 machine learning classifiers, ANN shows better accuracy and less execution time than others. That's why Artificial neural network was chosen for the classification of both the original and reduced datasets to compare the results.

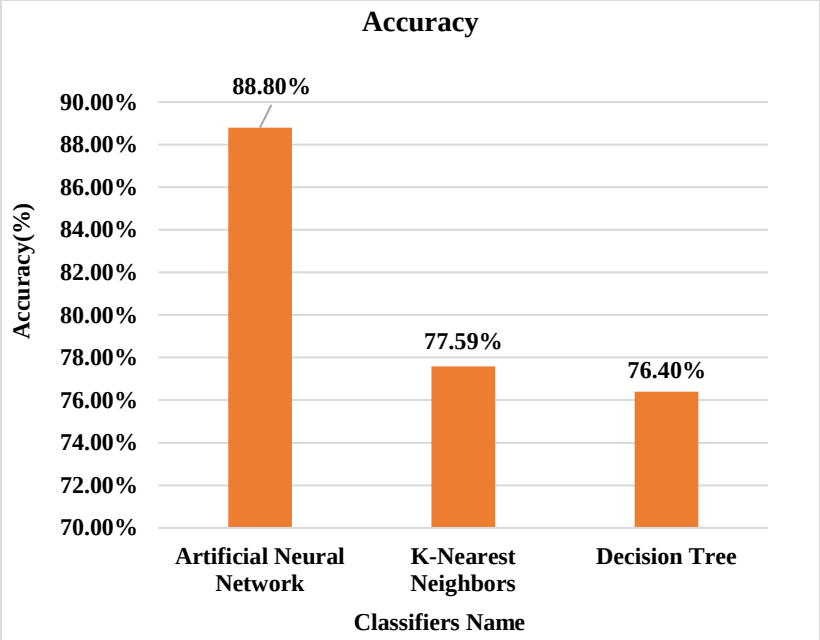

Fig. 5.1: Accuracy of ANN, Decision Tree and KNN by using original VIRUS-MNIST dataset.

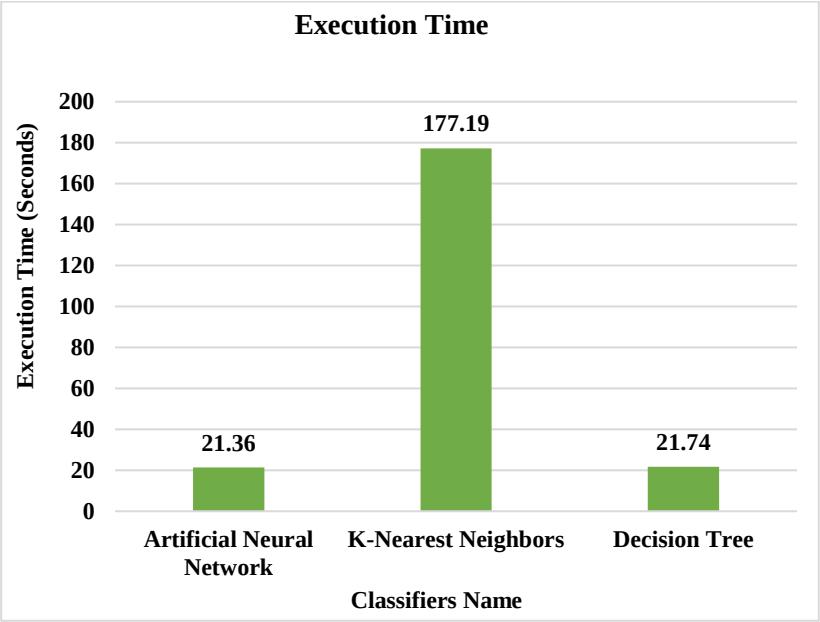

Fig. 5.2: Execution time of ANN, Decision Tree and KNN by using original VIRUS-MNIST dataset.

5.1.2. Applying ANN Classifier on Actual Dataset & New Datasets

At first ANN Classifier has been applied on the original dataset without reducing any feature. Then this same ANN classifier has been applied on the reduced features datasets which are generated from the modified k-means clustering algorithms. For original dataset and for each reduced datasets ANN have been applied 10 times to validate the performance and then the average values have been taken. When the ANN was performed on the original dataset with 1024 features the accuracy was 88.8% the execution time was 21.36 seconds.

5.1.2.1. Modified k-means clustering Algorithm (Based on Distance)

For our first algorithm, we can see from Table 4 that, as the number of features (K) increases the accuracy and the execution time increases. In such case, we can see with 600 features the accuracy is 88.7% which is almost same as the accuracy of the original dataset (1024 features) and the execution time is also less than the original dataset.

Table 4: Performance of ANN classifier by using different reduced VIRUS-MNIST datasets (from algorithm no.1) and original VIRUS-MNIST dataset.

Number of Features (K)	**Accuracy (%)**	**Execution Time (Seconds)**
50	80.9	7.89
100	83.8	8.83
200	86.6	10.32
400	87.5	11.60
500	87.9	12.42

550	88.1	14.96
600	**88.7**	**15.04**
1024	**88.8**	**21.36**

5.1.2.2. Modified k-means clustering Algorithm (Based on Weight)

For the second algorithm, we can see from Table V that we have again reduced the features to 600 and got the accuracy of 88.6% which is almost equivalent with the results found with 1024 features. That's why we have taken the 600 features into account.

Table 5: Performance of ANN classifier by using different reduced VIRUS-MNIST datasets (from algorithm no.2) and original VIRUS-MNIST dataset.

Number of Features (K)	**Accuracy (%)**	**Execution Time (Seconds)**
50	80.2	8.53
100	83.9	7.34
200	86.0	9.57
400	87.5	11.60
500	87.9	11.83
550	88.1	13.51
600	**88.6**	**13.43**
1024	**88.8**	**21.36**

5.1.3. Comparing Performance Between Actual Dataset & New Datasets

5.1.3.1. Comparing Accuracy & Execution Time

5.1.3.1.1. Modified k-means clustering Algorithm (Based on Distance)

We have reduced approximately 41.4% features and reduced the execution time 29.6% which means with around 58.6% features we are able to provide accuracy like the original dataset with less time.

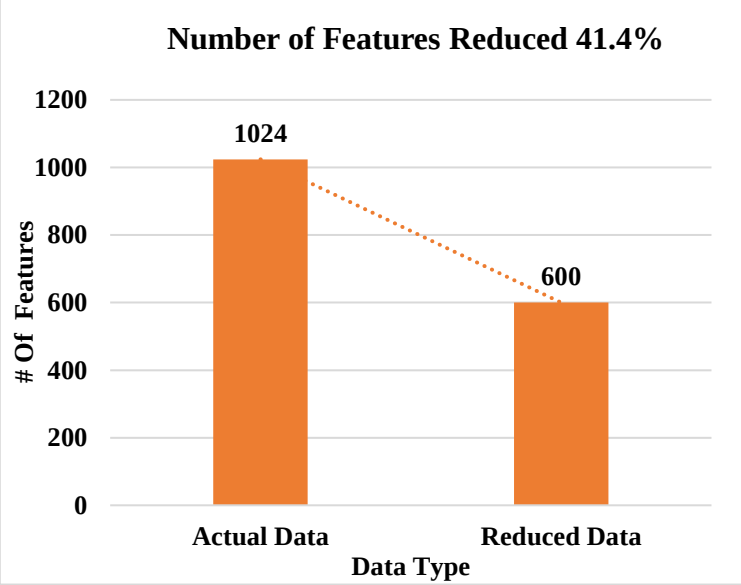

Fig. 5.3: Reducing features by using Modified k-means clustering Algorithm (Based on Distance).

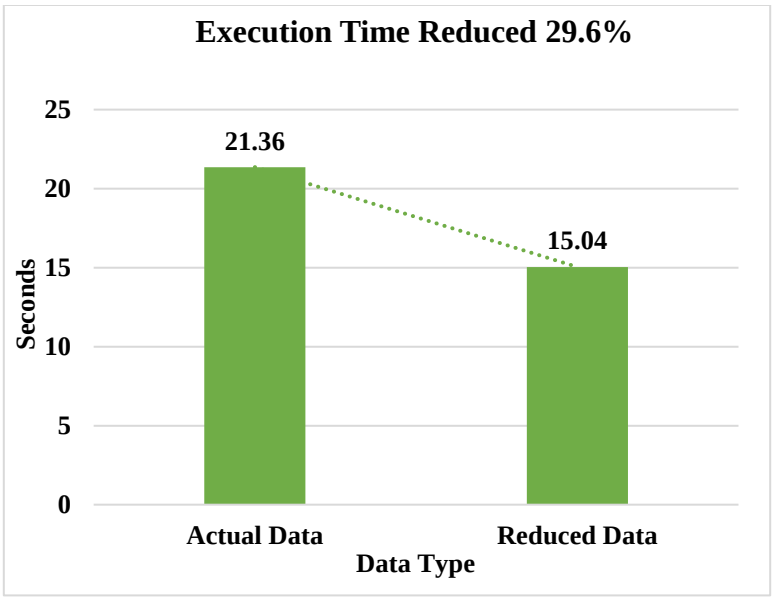

Fig. 5.4: Reducing execution time by using Modified k-means clustering Algorithm (Based on Distance).

5.1.3.1.2. Modified k-means clustering Algorithm (Based on Weight)

For the second algorithm, we have reduced 41.4% features as previous experiment and also reduced the execution time 37.1% which means with around 58.6% features, we are able to provide accuracy similar to the original dataset with less time.

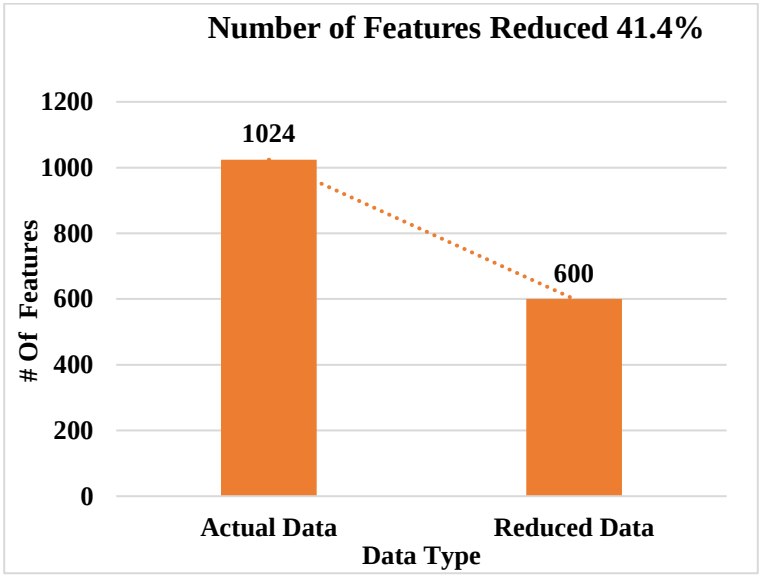

Fig. 5.5: Reducing features by using Modified k-means clustering Algorithm (Based on Weight).

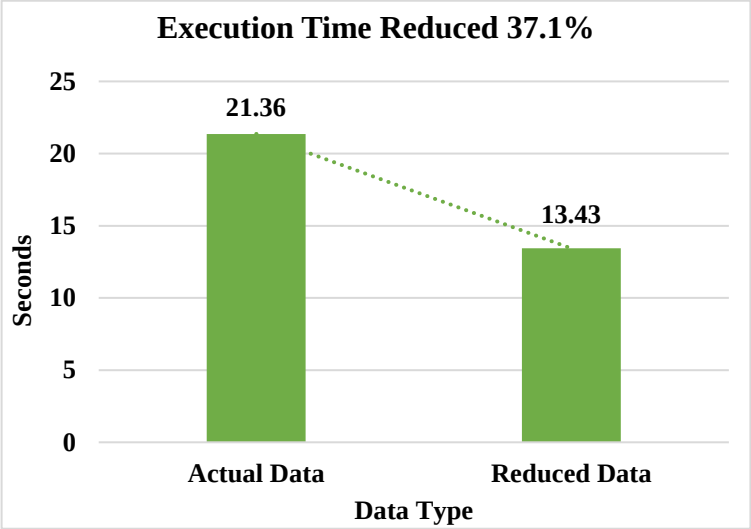

Fig. 5.6: Reducing execution time by using Modified k-means clustering Algorithm (Based on Weight).

5.1.3.2. Comparing Confusion Matrix

5.1.3.2.1. Modified K-Means Clustering Algorithm (Based on Distance)

From Figure 9 we can see that, In X-axis these are the labels of the actual class and in the Y-axis, these are the labels of the estimated/predicted class. Here the diagonal green blocks show the number and percent of the samples which are correctly classified. Red blocks show misclassifications. White blocks show total classification percentages for each row and column. And this gray block shows average classification rate.

We can see for the original dataset in Fig. 12 we have got an accuracy of 88.8% and error of 11.2% and for the reduced dataset in Fig. 13 the accuracy is 88.7% and error is 11.3%.

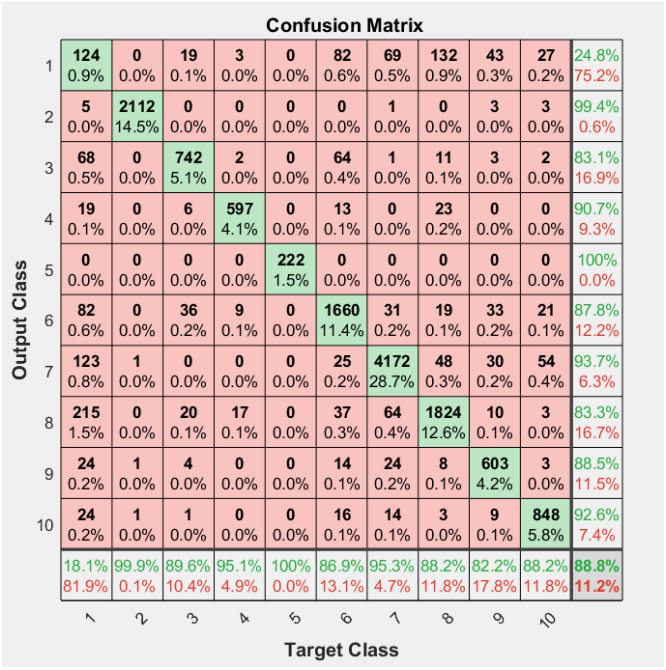

Fig. 5.7: Confusion matrix of original VIRUS-MNIST dataset

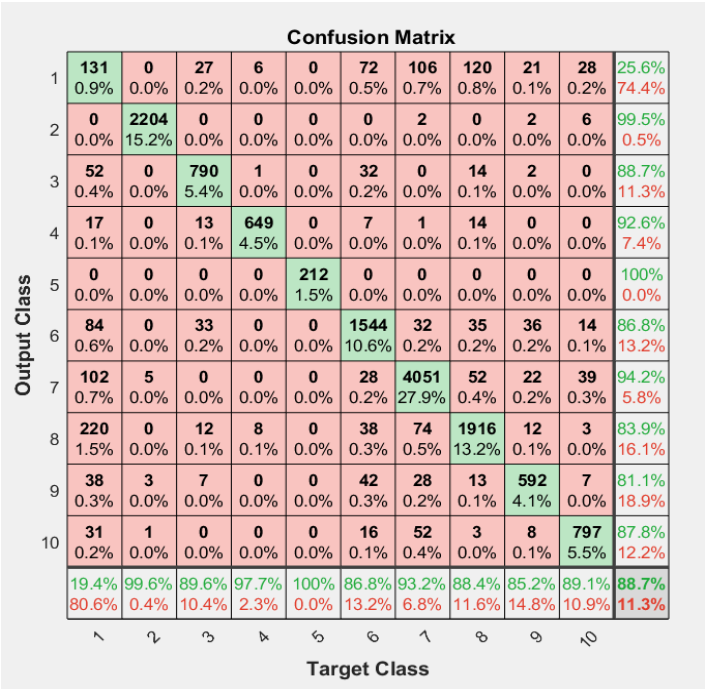

Fig. 5.8: Confusion matrix of reduced VIRUS-MNIST dataset by using Modified k-means clustering Algorithm (Based on distance)

5.1.3.2.1. Modified K-Means Clustering Algorithm (Based on Weight)

For the second algorithm, as per the confusion matrix we can see from Fig. 14, the accuracy for the reduced dataset with 600 feature is 88.6% and error is 11.4%.

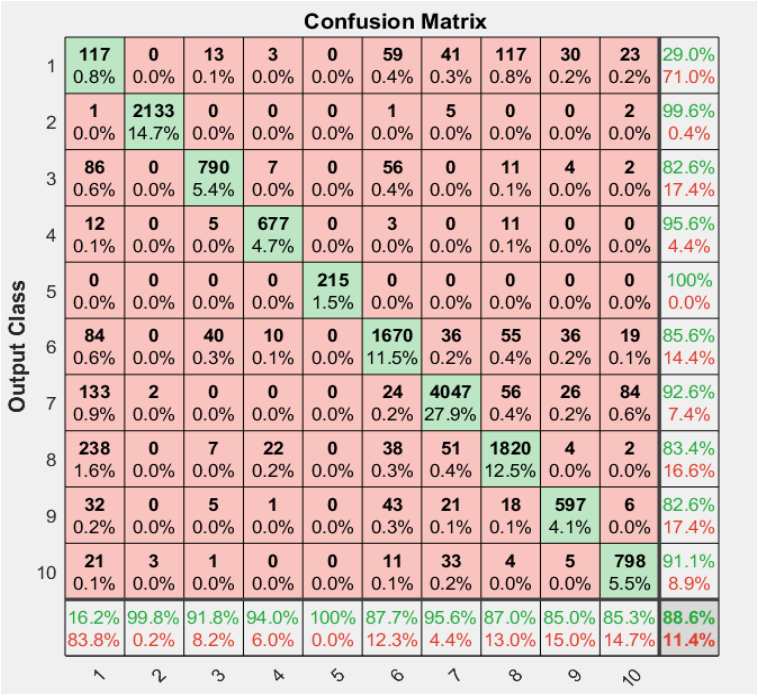

Fig. 5.9: Confusion matrix of reduced VIRUS-MNIST dataset by using Modified k-means clustering Algorithm (Based on weight)

5.1.3.3. Comparing F1 Score

In Table 6, F1 Scores for both original dataset and reduced datasets have been described. As per the following table we have an average f1 score of 84.39% for the original dataset and for the reduced datasets we have got the average f1 score of 84.28% for the first algorithm which is

modified k-means clustering algorithm ((Based on weight). I also found, average F1 score for the modified k-means clustering algorithm (Based on distance) is 84.01%.

Table 6: Comparing f1 scores of original VIRUS-MNIST dataset and reduced VIRUS-MNIST datasets.

Class	**F1 Scores of Original Dataset (1024 Features)**	**F1 Scores of Reduced Dataset from first algorithm (600 Features)**	**F1 Scores of Reduced Dataset from second algorithm (600 Features)**
0	21.0%	22.1%	20.8%
1	99.7%	99.5%	99.7%
2	86.2%	89.1%	87.0%
3	93.0%	95.1%	94.8%
4	100%	100%	100%
5	87.1%	86.8%	86.6%
6	94.5%	93.7%	94.1%
7	85.7%	86.1%	85.2%
8	85.2%	83.1%	83.8%
9	90.4%	88.4%	88.1%

Still both of our average f1 scores are higher than the average f1 score of the refence paper [12] from which the dataset was taken where the average f1 score was 80% with CNN classification model.

5.1.4. Applying K-fold Cross Validation

For k-fold cross validation, we have considered the value of K=5. We have applied this k-fold cross validation on both datasets we found from 2 different algorithms and from Table 7 and Table 8 we can see that, accuracy we found was almost same for both.

Table 7: Results of k-fold cross validation of reduced VIRUS-MNIST dataset (by using modified k-means clustering algorithm based on distance).

Iterations	**Accuracy (%)**
Iteration 1	88.1%
Iteration 2	88.1%
Iteration 3	88.7%
Iteration 4	88.5%
Iteration 5	88.0%

Table 8: Results of k-fold cross validation of reduced VIRUS-MNIST dataset (by using modified k-means clustering algorithm based on weight).

Iterations	**Accuracy (%)**
Iteration 1	88.0%
Iteration 2	88.1%
Iteration 3	88.6%
Iteration 4	88.1%
Iteration 5	88.2%

5.2. Experiments & Results

The following real dataset from the UCI repository was used in paper [9] to compare their proposed wk-means-based modified k-means feature reduction technique with sparse k-means, LLE, and PCA:

- Wine: contains 3 clusters, 178 data and 13 features

In each experiment, a real dataset was subjected to one of the aforementioned feature reduction methods. The resulting dataset was then clustered using the wk-means approach, and its accuracy and execution time were then reported.

Now to compare the performance of our two proposed methods with the above-mentioned feature selection methods, we have also applied our proposed two algorithms on the same wine dataset then the obtained reduced datasets are classified using ANN classifier and then the accuracy and execution time of the proposed feature reduction methods are reported.

The outcomes of clustering and clustering-based feature reduction heavily rely on the initial cluster centres that are chosen at random. As a result, each experiment was carried out ten times, and Table 9 presents the mean accuracy.

Table 9: Accuracy and execution time of clustering after feature selection by using our proposed methods, LLE, PCA, sparse k-means and modified wk-means.

# Of features after feature reduction	**Feature Reduction Algorithm**	**Accuracy (%)**	**Execution Time (seconds)**
K=2	Modified wk-means [9]	65.1	0.058
	Sparse K-means [7]	56.2	0.025

	LLE [8]	54.4	0.08
	PCA	64.9	**0.004**
	Our proposed Algorithm #1	**71.7**	**0.005**
	Our proposed Algorithm #2	**73.6**	**0.005**
K=4	Modified wk-means [9]	69.2	0.061
	Sparse K-means [7]	67.4	0.025
	LLE [8]	68.2	0.075
	PCA	67.9	**0.004**
	Our proposed Algorithm #1	**84.9**	**0.006**
	Our proposed Algorithm #2	**83.1**	**0.005**
K=6	Modified wk-means [9]	66.8	0.064
	Sparse K-means [7]	65.2	0.027
	LLE [8]	49.2	0.073
	PCA	54.3	**0.006**
	Our proposed Algorithm #1	**92.5**	**0.008**
	Our proposed Algorithm #2	**94.3**	**0.007**

According to Table 9, the accuracy of our two proposed methods is better than Modified wk-means [9], PCA, LLE [8] and sparse k-means [7] for all cases. Therefore, the proposed methods are more accurate than the four other methods. According to Table 9, it can also be stated that the speed of our proposed methods is better than other methods except PCA.

5.3. Experiments & Results

In this experiment, we have applied both the algorithms on CNAE dataset [11] to compare the performance with another paper [10] where the authors gained 88.3% accuracy with 60 features through their algorithm. According to Table 10, with our first algorithm which is modified k-means

clustering algorithm (based on distance), we got 88.4% accuracy with 70 features. And our second algorithm, which is k-means clustering algorithm (based on weight), we got 88.9% accuracy with 55 features which is better than their accuracy with less features.

Table 10: Accuracy and execution time of clustering after feature selection by using our proposed methods and modified k-means.

Feature Reduction Algorithms	**# Of Features**	**Highest Classification Accuracy (%)**
Modified K-means Algorithm [10]	60	88.3
Our proposed Algorithm #1	70	88.4
Our proposed Algorithm #2	55	88.9

5.4. Experiments & Results

In this experiment, the intersection of feature selection of our two algorithms is measured which means we have calculated those features only which are chosen by both of our algorithms. For VIRUS-MNIST Dataset the intersection rate is 68%, for Wine Dataset it is 83% and for CNAE Dataset it is 62%. So, we can say the average intersection of feature selection of our two algorithms is 71%.

Chapter 6

6. Conclusion and Future Work

The conclusion and next effort for the advancement of this research are discussed in this chapter.

6.1. Conclusion

In this research, two new feature reduction techniques based on the k-means clustering algorithm are given in order to evaluate the effectiveness of k-means-based clustering methods. K-means clustering can be used to reduce the number of redundant features because it tries to collect related features into a single cluster, as was discovered in our study. One representative feature could represent a group of similar features. The two proposed methods' experimental findings on three real datasets indicated that:

- For VIRUS-MNIST Dataset, using our first algorithm, we were able to reduce the number of features by 41.4%, while also cutting the execution time by 29.6%. Using our second algorithm, we were able to reduce the number of features in the dataset by 41.4%, while also cutting the execution time by 37.1%. This means that with only 58.6% of the features, we can deliver accuracy that is comparable to that of the original dataset in less time.
- For Wine Dataset, the accuracy of our two proposed methods is better than Modified wk-means [9], PCA, LLE [8] and sparse k-means [7] and the speed of our proposed methods is better than other methods except PCA.

- For CNAE Dataset, with fewer features the accuracy is better than our second method which is k-means clustering algorithm (based on weight) than proposed method of article [10].
- The intersection of feature selection of our two algorithms is approximately 71%.

6.2. Future Work

In this study, we suggest two new algorithms that perform better than PCA but require more computation time. Both of these algorithms' upcoming work may involve some changes to reduce the execution time.

In Future, we can combine the features which we found from our two algorithms to get better accuracy.

Instead of a fuzzy clustering model, our suggested models are based on k-means clustering. Each data point can be a member of many clusters when using fuzzy clustering, also known as soft clustering or soft k-means. Future research will focus on our proposed models' fuzzy iteration.

With the intention of understanding the sensitivity and nature of the data analysed, this research work can potentially be expanded to various datasets. Feature selection should be prioritised more, however, in light of the expanding data universe and the demand for online classifiers. In every aspect of life, one of the most crucial tools will be the analysis of trends and patterns discovered in big data.

References

[1] H. Liu and H. Motoda, “Computational Methods of Feature Selection”, Chapman and Hall/CRC Press, 2007.

[2] Jiliang Tang, Salem Alelyani and Huan Liu, “Feature Selection for Classification: A Review”, Data Classification: Algorithms and Applications, CRC Press pp. 37, 2014.

[3] Saeys, Y., I. Inza, dan P. Larranaga, “A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics,” Vol. 23, No. 19, pp 2507-2517, 2007.

[4] Liu, H. dan L. Yu, “Toward Integrating Feature Selection Algorithms for Classification and Clustering”, IEEE Transaction on Knowledge and Data Engineering. Vol. 17, No. 4, 2005

[5] I. Jolliffe, “Principal Component Analysis”, 2nd ed., NY: Springer, 2002

[6] X. Qiu, Y. Qiu, G. Feng and P. Li, "A sparse fuzzy c-means algorithm based on sparse clustering frame work," Neurocomputing, vol. 157, pp. 290-295, 2015.

[7] R. Tibshirani and D. M. Witten, "A framework for feature selection in clustering," J Am Stat Assoc., vol. 105, no. 490, p. 713–726, 2010

[8] S. T. Rovis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Sience, vol. 290, pp. 2323-2326, 2000.

[9] Mahboubeh Haghayeghipour, Yahya Forghani, “A Novel K-means-based Feature Reduction”, International Journal of Integrated Engineering, Vol. 11, No. 1, 2019

[10] Dewi Pramudi Ismia, Shireen Panchoob, Murintoc, “K-means clustering based filter feature selection on high dimensional data”, International Journal of Advances in Intelligent Informatics ISSN: 2442-6571 Vol 2, No 1, March 2016.

[11] CNAE-9 Dataset. https://archive.ics.uci.edu/ml/datasets/CNAE-9

[12] Noever, D. and Noever, Sam, “Virus-MNIST: Portable Executable Files as Images for Malware Detection”, https://www.kaggle.com/datamunge/virusmnist.

[13] R. Tibshirani and D. M. Witten, "A framework for feature selection in clustering," J Am Stat Assoc., vol. 105, no. 490, p. 713–726, 2010.

[14] Wine Dataset. https://archive.ics.uci.edu/ml/datasets/wine

[15] D. Wettschereck, D. Aha, and T. Mohri, “A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms”, Artificial Intelligence Review, 11:273–314, 1997.

[16] Geisser, S. (1975), “The predictive sample reuse method with applications”, Journal of the American Statistical Association 70(350): 320–328.

[17] T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning”, Springer, 2001.

[18] Dhanachandra N., Manglem K. and Chanu Y., “Image Segmentation using K-Means clustering algorithm and subtractive clustering algorithm”, Eleventh International Multi-Conference on Information Processing 2015, Procedia Computer Science 54, 2015, pp 764-771.

[19] X. Huang, Y. Ye and H. Zhang, "Extensions of K-means-Type Algorithms: A New Clustering Framework by Integrating Intra-cluster Compactness and Inter-cluster Separation," IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, vol. 25, no. 8, pp. 1433 - 1446, 2014.

[20] K. C. F. C. a. S. W. Z. Deng, "Enhanced soft subspace clustering integrating within-cluster and between-cluster information," Pattern Recognition, vol. 43, p. 767–781, 2010.

[21] A. S. S. A. D. Gujar, "A new framework of K-means algorithm by combining the dispersions of clusters," International Journal of Advance Research in Computer Science and Management Studies, vol. 4, pp. 73-78, 2016.

[22] Meng Jianliang, Shang Haikun, Bian Ling, “The Application on Intrusion Detection Based on K-means Cluster Algorithm”, International Forum on Information Technology and Application, DOI 10.1109, 2009.

[23] Schuman, C.D.; Birdwell, J.D. Dynamic artificial neural networks with affective systems. PLoS ONE 2013, 8, e80455.

[24] Dixon, B.; Candade, N. Multispectral landuse classification using neural networks and support vector machines: One or the other, or both? Int. J. Remote Sens. 2008, 29, 1185–1206.

[25] Yilmaz, I.; Kaynar, O. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. Expert Syst. Appl. 2011, 38, 5958–5966.

[26] Moody, J.; Darken, C.J. Fast learning in networks of locally-tuned processing units. Neural Comput. 1989, 1, 281–294.

[27] Shafizadeh-Moghadam, H.; Hagenauer, J.; Farajzadeh, M.; Helbich, M. Performance analysis of radial basis function networks and multi-layer perceptron networks in modelling urban change: A case study. Int. J. Geogr. Inf. Sci. 2015, 29, 606–623.

[28] Quinlan, J. R. (1993). C4.5, Programs for Machine Learning. Morgan Kaufmann San Mateo Ca, 1993.

[29] Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27

[30] Mitchell T (1997) Machine learning. McGraw Hill, New York

[31] Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

[32] Ghassemieh, M.; Nasseri, M. Evaluation of stiffened end-plate moment connection through optimized artificial neural network. J. Softw. Eng. Appl. 2012, 5, 156–167.

[33] R.Agrawal,J.Gehrke,D.Gunopulos, and P. Raghavan.Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proc.ACM SIGMOD,June 1998: 94-105

[34] G. Milligan. AValidation Study of a Variable Weighting Algorithm for Cluster Analysis.J.Classification 1989: 53-71

[35] Bradley,Fayyad.Refining Initial Point for K-means Clustering.Proceedings of the Fifteenth International Conference on Machine Learning,1998