Decision (December 2020) 47(4):431–445
https://doi.org/10.1007/s40622-020-00265-3

RESEARCH ARTICLE

An approach of improving decision tree classifier using condensed informative data

Archana R. Panhalkar · Dharmpal D. Doye

Accepted: 10 November 2020 / Published online: 28 January 2021

Abstract The advancement of new technologies in today's era produces a vast amount of data. To store, analyze and mine knowledge from huge data requires large space as well as better execution speed. To train classifiers using a large amount of data requires more time and space. To avoid wastage of time and space, there is a need to mine significant information from a huge collection of data. Decision tree is one of the promising classifiers which mine knowledge from huge data. This paper aims to reduce the data to construct efficient decision tree classifier. This paper presents a method which finds informative data to improve the performance of decision tree classifier. Two clustering-based methods are proposed for dimensionality reduction and utilizing knowledge from outliers. These condensed data are applied to the decision tree for high prediction accuracy. The uniqueness of the first method is that it finds the representative instances from clusters that utilize knowledge of its neighboring data. The second method uses supervised clustering which finds the number of cluster representatives for the reduction of data. With an increase in the prediction accuracy of a tree, these methods decrease the size, building time and space required for decision tree classifiers. These novel methods are united into a single supervised and unsupervised Decision Tree based on Cluster Analysis Pre-processing (DTCAP) which hunts the informative instances from a small, medium and large dataset. The experiments are conducted on a standard UCI dataset of different sizes. It illustrates that the method with its simplicity performs a reduction of data up to 50%. It produces a qualitative dataset which enhances the performance of the decision tree classifier.

Keywords Data mining · Decision tree classifier · K-means clustering · C4.5 · Instance reduction

Introduction

Data mining refers to drawing knowledge from a vast amount of data available from various sources. In today's computerized world, a large amount of data get produced in banking, businesses, hospitals, internet users and other government organizations. To mine fruitful information from this large data is one of the complex tasks. To process, analyze such big data is very difficult. Every data mining task requires a quality dataset to extract the knowledge from it. The huge dataset is having problems like it requires more storage and processing time. Large data increase processing complexity without increasing performance. This explosion of data wants innovative techniques to transmit this data into valuable

A. R. Panhalkar (✉) · D. D. Doye
Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded, Maharashtra, India
e-mail: archana10bhosale@rediffmail.com

knowledge. To train the classifiers for decisions from this vast amount of data is challenging. Hence, to mine useful knowledge for accurate decision making, informative data need to be extracted from this large dataset. If the representative samples are taken which will overcome all the above drawbacks of giant data set, then the knowledge extractions grow to be straightforward.

With the advancement in mining techniques, various data reduction methods are proposed in the literature and study (Randall and Martinez 2000). To mine the useful information, supervised classification plays an important role (Han and Kamber 2006). Decision tree learning is one of the simple tools for detecting patterns, associations and knowledge from data (Han and Kamber 2006; Quinlan 1993). Decision trees are simple, powerful and analytical in nature and thus are most appropriate for data mining, so this research has emphasized the use of decision trees as classifiers. To train a decision tree from large data requires large processing time and space. The proposed work is contributed to the reduction of data to expedient the performance of the decision tree (DT) classifier. When there is an issue of data compression, two broadly categorized techniques called sample reduction and feature reduction are used. Feature reduction is the task of omitting the unwanted features of the dataset to shrink the dataset (Yodjaiphet et al. 2015; Phinyomark et al. 2012; Chao and Chen 2005). Feature reduction leads to loss of important information from data which increases the misclassification rate. A large dataset may have duplicate and less informative instances that increase the size of data. Sample reduction or instance reduction is a technique of horizontal reduction of the dataset without losing features (Quinlan 1993). Reducing duplicate, similar or uninformative instances have less or no impact on decision making in data mining. This work proposes to inculcate the dataset scaling based on instance reduction. It aims to prepare the data for the decision tree classifier using sample selection methods. This novel method enhances the performance of the decision tree. A novel algorithm called decision tree based on cluster analysis pre-processing (DTCAP) with supervised and unsupervised clustering is proposed. It reduces the data using thicken border. By using experiments, we argue that only selecting the border instances increases the possibility of choosing outliers. The novelty of the selection of instances is that it selects the representative instances from clusters. The second method lays its uniqueness in creating several small clusters and choosing the best informative of that cluster. The proposed preprocessing technique uses modified K-means clustering along with a selection of representative instances. These selected instances best represent the whole data. These selective instances are used for the training of decision tree instead of whole data. The proposed methods not only improve the performance of the DT but also reduce the size of the tree. The performance of both methods is evaluated on standard UCI datasets of different sizes.

Related work

To build accurate and powerful classifiers that have less misclassification rate requires training with significant data. Finding significant instances from a huge dataset to train classifiers is one of the challenging approaches. Supervised classification like decision tree best performs for the representative instances because they are the best classifier of the models. Condensing of dataset differ in the manner in which the search for best representative instances is done. An outstanding work of condensing dataset is based on the nearest neighbor approach, graph-based approaches and opponent-based approaches which are proposed in the literature (Wilson 1972; Hart 1968; Gates 1972; Angiulli 2005; Chou et al. 2006; Toussaint and Foulsen 1979; Marchiori 2008). One of the simplest approaches of using the K-nearest neighbor (K-NN) was Edited Nearest Neighbor (Wilson 1972), which filters noisy instances by using K-NN to increase classification accuracy. Edited Nearest Neighbor method considers the most powerful method which uses nearest neighbor rules to reduce the dataset. It filters instances until the correctly classified instances are part of the reduced dataset. Condensed nearest neighbor by Hart (1968) presented the classical work based on the nearest neighbor. It constitutes the reduced set by inserting instances that are not classified by the training dataset. Many variants and extensions of condensed nearest neighbor were proposed like Reduced Nearest Neighbor Rule (Gates 1972), the fast Condensed Nearest Neighbor (Angiulli 2005), and Generalized Condensed Nearest Neighbor (Chou et al. 2006) which performs the discarding superfluous instances and scaled dataset to small size.

All these methods use a similar feature of neighboring instances to condense the dataset.

Very few graph-based approaches are presented in the literature. The first approach is proposed by Toussaint and Foulsen (1979) which is based on Voronoi diagrams. It partitions the planes into disjoint regions. It compares the point (instance) from the current polygon with neighboring polygons and if these are of the same class then the point from the dataset is reduced. This method does not compute or compare distances among all data instances to shrink the dataset which was a major drawback of all nearest neighbor methods used for data reduction (Wilson 1972; Hart 1968; Gates 1972; Angiulli 2005; Chou et al. 2006; Toussaint and Foulsen 1979; Marchiori 2008). The second graph-based approach proposed by Marchiori (2008) uses the Hit Miss Network graph for the scaling of the dataset. It showed how the home structural property of nodes in the graph provides information about the similarity of the corresponding points to the decision edge of the 1-Nearest Neighbor rule. Without affecting the accuracy of classifiers, it reduces data efficiently.

Border selection approaches (Olvera-López et al. 2010; Nikolaidis et al. 2011; Hernandez-Lea et al. 2013; Cavalcanti et al. 2013; Alvar and Abadeh 2016) are found to be best for selecting representative instances from large datasets. Border selection approaches use clustering to reduce dataset. Researchers proved that the instances which lie on the border of clusters are the best representatives of data. Most of the algorithms in the literature use border instances because they best categorize among the classes and give efficient results for a classifier. Classifiers give the best performance for such representative datasets. Olvera-López et al. (2010) proved that all instances will not provide qualitative information. Prototype Selection Clustering (PSC) proposed in Olvera-López et al. (2010) uses homogeneous and heterogeneous clusters to select the best prototypes(instances). For homogeneous clusters, the instances near to the mean are selected and for heterogeneous clsters, the instances near to the other clusters are selected. In this method, cluster selection is based on trial and not much reduction in a dataset. Nikolaidis et al. (2011) used a multistage method or pruning the dataset which is also a boundary preserving method. It smoothes boundaries and selects adjacent enemies. These enemy instances whose line segment at angles greater than user-defined thresholds are selected. It prunes the border instances and clusters the non-border points for prototype selection. The results are compared using different datasets. Hernandez-Lea et al. (2013) proposed the IRB instance selection algorithm which first filters the noisy instances and smoothes the boundaries to avoid overlap of classes. It uses the ranking of instances. The best-ranked instances are those which are near to the border. IRB selects best ranked, instances on the border and some medium and low-rank non-border instances to improve the classification accuracy. Cavalcanti et al. (2013) proposed adaptive threshold-based instance selection algorithms [ATISA1-ATISA2] which uses various thresholds as a distance of every instance with its nearest enemy instance for selecting representatives. It has given a more reduced set with increased accuracy on various datasets. Alvar and Abadeh (2016) used fuzzy frequent patterns for the reduction of datasets. This method has significance in preserving appropriate border points. The method evaluates the best performance on K nearest neighbor classifier.

Dimensionality reduction can also be achieved using clustering-based approaches (Sanguinetti 2008; Chen and Cheng 2008; Czarnowski 2012; Ougiaroglou and Evangelidis 2012; Pechenizkiy et al. 2006). Sanguinetti (2008) presented the latent variable model to reduce clustered dataset. It selects optimal linear projections using unsupervised Linear Discriminate Analysis of a large dimensional dataset. Chen and Cheng (2008) proposed cluster support vector machines for selecting representative instances from huge data. It first forms the clusters and selects the samples nearest to the hyper-plane and the centers of homogeneous clusters. It deletes the surplus data and increases the speed of classification. Czarnowski (2012) combined clustering with agent-based population method for the reduction of the dimension of data sets. Four variants of the instance selection approach that is similarity coefficients, stratification strategy, modified stratification strategy and K-means clustering methods are used in Czarnowski (2012). Ougiaroglou and Evangelidis (2012) proposed a fast, nonparametric reduction based on clustering. It selects the centroids of homogeneous clusters, if the cluster is not homogeneous, then applied the k clustering till it will not become homogeneous. But this method does not increase classification accuracy. Pechenizkiy et al. (2006) used fuzzy C-means clustering to cluster the

input prototypes and select the representative instances. CLU (Lumini and Nanni 2006) is one of the best methods based on clustering which is applied to the biometric signatures. Principle component analysis-based mini-batch clustering (Peng et al. 2018) method is used for reducing data, which best performs in intrusion detection. This method compares performance with K-means clustering. This method is applicable and verified or intrusion detection dataset. Multi Kernel SVM is applied to a reduced dataset in Tang et al. (2019). The reduced data are obtained by K-means clustering followed by removing outliers. This method fails to utilize knowledge from outliers.

From the literature, it is observed that numerous varied methods have been studied which selects representatives from a huge dataset. This reduced dataset gives significant performance for various classification strategies. All these methods using the K-means clustering approach select representatives. These methods only select center instances or border instances. Selecting border instances may lead to the selection of noisy instances. When cluster along with centers of clusters is formed, there are many representative instances which lies near to centers. Most discriminate samples found at the borders of clusters also utilizes more knowledge about datasets. Hence, in this work, the emphasis is given to find representative instances from all over clusters which improve the performance of the decision tree classifier.

DTCAP method proposed in this paper put down its simplicity in the use of modified K-means clustering with a novel selection approach of instances for preparing data for outperforming the performance and reduction of the size of the decision tree classifier. It not only increases the accuracy of decision tree but also shows significant improvement in various aspects of the decision tree. The proposed technique reduces the size of the tree and decreases the number of leaves and increases the accuracy of the decision tree. The organization of sections in this paper is as follows. Literature review is described in Sect. 2. Some basic terminologies related to our DTCAP approach are explained in Sect. 3. Section 4 presents our novel methods along with the entire DTCAP algorithm. Section 5 discusses the results of experiments carried out on various datasets.

Decision tree classifier and modified K-means clustering

A decision tree is a self-illustrative, accurate supervised classifier used in data mining. Decision tree learning offers flowchart like tools for finding patterns by creating classifier from data. Decision trees are straightforward, influential and analytical in nature and thus are most suitable for various data mining tasks. In this research, the emphasis is given to improve the decision tree classifier. Aim of the research is to prepare a significant dataset for decision tree construction which increases the performance of the decision tree. Various algorithms like CHAID (Kass 1980), ID3 (Quinlan 1986), C4.5 (Quinlan 1993) and CART (Breiman et al. 1984) are implemented and tested in the literature which are found to be promising classifiers for various size and type of dataset. As compared to all decision tree learners in terms of prediction accuracy, size of a tree, construction time of tree, various size and type of dataset, C4.5 is an effective decision tree classifier (Sathyadevan and Nair 2015). Due to various advantages and properties of the C4.5 decision tree, the performance of condensed data produced by the proposed approach is applied to C4.5. This efficient C4.5 decision tree construction (Quinlan 1993) takes place by selecting the best attribute, which best partitions the instances and construct trees by recursively partitioning a training set. The decision tree is a tree like structure which consists of edges and nodes. Nodes are divided into internal nodes and leaf nodes. Internal nodes are called testing attributes. Leaf nodes depict classes, which get recognized after testing attributes from root to leaf node.

Construction of decision tree

The decision tree classifier (Quinlan 1986) is created using the divide and conquer approach. The dataset is divided into two parts, training and testing dataset. It evolves a decision tree for a given T training set comprising a set of data instances. Let the classes be denoted by $\{C1, C2, C3, \ldots Cn\}$. The steps are as follows.

1. Initially, the class occurrence is computed for all data instances in training set T.

2. If all instances belong to an identical class, node K is created with that class. This K node becomes leaf node.
3. If set T include instances belong to more than one class, then select the best attribute satisfying splitting norm and choose for the test.
4. The training set T is partitioned on the basis of this test into K exclusive subsets $\{T1, T2, T3, \ldots, Tn\}$.
5. Go to step 2 for every remaining non-empty partition.
6. Stop

An increase in the dataset unexpectedly increases the building time of decision tree because more instances increases the scanning time of the dataset. A large number of records increase the size of the decision tree without enhancing accuracy. This proposed work aims to preprocess the data and scale the dataset such that in all aspects the decision tree becomes efficient. The proposed approach focus on one of the simplest method K-means clustering to shrink the dataset.

Modified K-means clustering

Clustering is one of the unsupervised learners in data mining which performs natural grouping of similar instances. It is a preprocessing step for outlier detection and compressing the data. Though K-means clustering was first proposed 50 years ago, it is one of the most commonly used algorithms for clustering. The straightforwardness of implementation, efficiency and practical success are the key reasons for its popularity (Jain 2010). K-means is one of the fast, simplest and robust unsupervised clustering approaches (Han and Kamber 2006). There are several methods studied in the literature (Sanguinetti 2008; Chen and Cheng 2008; Czarnowski 2012; Ougiaroglou and Evangelidis 2012; Pechenizkiy et al. 2006) show that clustering can perform the best reduction of data because by using similarity-based clusters, it is easy to delete redundant information and a healthy way to find out most ambassador instances. Proposed research emphasizes the use of simple K-means clustering with modification. This step is used as a preprocessing step for a shrinking dataset which gives considerable performance for decision tree. Other clustering algorithms with better features tend to be more expensive. In this case, K-means becomes a great solution for pre-clustering, reducing the space into disjoint smaller sub-spaces where other clustering algorithms can be applied.

In this research work, the variation of K-means clustering is used. To find a new centroid, instead of choosing mean, the nearest data point is selected to the mean as shown in Eq. 3. Because of this mapping of data points is done with real data points instead of mean in forming clusters. In this work, an innovative approach is used to create many small clusters on supervised data. Modified traditional unsupervised clustering is also used which is found to be promising on a numerical and categorical dataset. Modified K-means algorithm is as follows.

Algorithm 1 Modified K-Means Clustering Algorithm

Input: $D = \{d_1, d_2, d_3, \ldots .d_n\}$ be n data points.
Output: K number of Clusters

K-Means(D, K)

1: *Randomly select K cluster centroids* $\{C_1, C_2, C_3 \ldots .C_k\}$.
2: *Calculate the Euclidean distance between each data point* d_p *and cluster centers* C_i.
3: *Assign the data point to the cluster center* $C_i^{(t)}$ *whose distance from the cluster center is minimum of all the cluster centers.* m_i *is a set of data points assigned to cluster* $C_i^{(t)}$ *and calculated using Eq.(1).*

$$m_i = \{d_p : ||d_p - C_i^{(t)}||^2 \leq ||d_p - C_j^{(t)}||^2 \in \forall j \qquad 1 \leq j \leq k\} \tag{1}$$

4: *Recalculate the new cluster center using Eq.(2).*

$$C_i^{(t+1)} = \frac{1}{|m_i|} \sum_{d_i \in m_i} d_i \tag{2}$$

5: *Recalculate the distance between each data point and new obtained cluster centers.*
6: *If no data point was reassigned then stop, otherwise repeat from step 3.*

Decision Tree classifier based on Cluster Analysis Pre-processing algorithm (DTCAP)

A novel approach of improving decision tree classifier based on cluster analysis pre-processing (DTCAP) is proposed in this paper. Clustering techniques are used to aggregate the objects into groups according to similarity measures. Whether the number of groups is pre-defined (supervised clustering) or not (unsupervised clustering), clustering techniques do not provide decision rules or a decision tree for the associations that are implemented (Han and Kamber 2006). The current study proposes and evaluates a new technique to define a decision tree based on cluster analysis. The DTCAP approach is divided into three parts:

1 Supervised and Unsupervised Clustering of data.
2 Opting representatives from clusters.
3 Decision tree construction with Scaled data.

DTCAP consists of three steps. In the first step, supervised and unsupervised clustering is done by making modifications in the K-means algorithm with supervised and unsupervised data. The second step consists of a novel approach of selecting the best representative instances from clusters, which uses two different approaches. These informative instances selected by the proposed approach constitute quality scaled data. These scaled data are used to create an efficient and optimized decision tree.

Supervised and unsupervised clustering of data

As studied in the literature, K-means is one of the superior approaches to group similar elements together. Similar instances represent identical information so if they are terminated from data will not affect classifier performance. So if instead of selecting the number of similar prototypes, few representatives are selected for building classifiers. It will not only save space but also building performance without compromising the performance of decision tree. In this proposed method, the varied K-means algorithm is used.

In K-means, the centroid is the average of all clustered instances, so it is not a real instance. Instead of taking the mean of the data point as a new centroid, the proposed algorithm seizes the real instance nearest to the mean centroid as a new centroid. Two different approaches Clustering Approach-1 and Clustering Approach-2 are used for clustering. First Clustering Approach-1 is forming clusters for unsupervised data (without class) and second Clustering Approach-2 is to form the clusters on distributed data according to class. In Clustering Approach-2, distributed means the data instances are divided according to classes and then it will form many smaller clusters. The required parameters like the number of clusters (K) are entered depending on the size and nature of the dataset. In clustering, an important parameter is the number of clusters (K). To optimize the time required for clustering and finding significant data, experiments are performed on all K values. Experiments are carried out to find the impact of the reduction of data using a different number of clusters on the efficiency of the decision tree. It is observed that for most of the dataset, minimum 3 clusters and maximum 10 clusters can be formed. In the second approach, 50% of clusters are formed on the size of the data. Both approaches are used for scaling data which will increases decision tree performance. Both approaches are summarized as follows.

Algorithm 2 Clustering Approach-1

Input: $D = \{d_1, d_2, d_3, \ldots .d_n\}$ be n data points.
Output: K number of Clusters

Unsup K-Means(D, K)
1: *Calculate the mean of a cluster using Eq.(2).*
2: *Select new centroid* $C_i^{(t+1)}$ *to mean Eq.(3).*

$$C_i^{(t+1)} = \{d_j : ||d_j - C_i^{(t)}||^2 \leq ||d_p - C_i^{(t)}||^2 \in \forall d_j \in m_i\} \tag{3}$$

3: *Recalculate the distance between each data point and new obtained cluster centers* $C_i^{(t+1)}$.
4: *If no data point was reassigned then stop, otherwise repeat from step 3.*

In the second innovative Clustering Approach-2, preprocessing is performed to distribute data according to class. We are forming a number of clusters for each class instance and then applying opting Representative Approach-2 to the clusters. The selection of K is 50% the size of the input dataset D_j which provides the best shrinking increased performance of DT.

Algorithm 3 Clustering Approach-2

Preprocessing: Distribute the instances according to classes. Suppose j number of classes is present in D then Divide dataset in j parts. Apply the following algorithm for each dataset D_j .

Input: $D_j = \{d_1, d_2, d_3, \ldots . d_n\}$ be n data points.
Output: K number of Clusters.

Sup K-Means(D_j, K) .
1: *Perform first three steps of* ***K-Means(D, K)***.
2: *Calculate the mean of a cluster using Eq.(2).*
3: *Select new centroid* $C_i^{(t+1)}$ *to mean Eq. (3).*
4: *Recalculate the distance between each data point and new obtained cluster centers* $C_i^{(t+1)}$.
5: *If no data point is reassigned then stop, otherwise repeat from step 3.*

Opting representatives from clusters

This is the second step of the DTCAP method which selects representative instances from clusters formed in the previous step. For selecting the representative instances from clusters that are formed in the previous section, two different methods depending on the two clustering approach are used.

Opting representatives Approach-1

In clusters, the data points are either border or non-border points. Border points are situated near to the cluster boundaries. From the literature (Olvera-López et al. 2010; Nikolaidis et al. 2011; Hernandez-Lea et al. 2013; Cavalcanti et al. 2013; Alvar and Abadeh 2016), the majority of instance selection (IS) algorithms are emphasized on selecting border points because it contributes more than the non-border points. From experiments, it is argued that only selecting enemies that lie on the border may select outliers. The border points as well as points near to border which utilizes maximum information of clusters are selected. This novel approach of a selection of border and instances close to the border is illustrated in Fig. 1. Consider that there are two clusters formed on the dataset. In Fig. 1a, the boundary between two different clusters of a dataset is shown with a solid line. Figure 1b shows that instances which lies on thicken border are selected as representative instances in the reduced dataset. The width of the border is selected using experiments. The main reason for choosing the instances from a specific width border is to choose more informative instances which lie on the border and close to the border. Algorithms are also choosing instances at the center of the cluster. For that purpose, all instances lie inside the circle of radius T and circle center as cluster centers are selected in a reduced dataset. Figure 1c shows how we have selected the data points using a radius of the circle (T). Radius (T) is flexible depending upon the size of data and the size of scaled data. The same approach is summarized using Algorithm 4.

Algorithm 4 Opting Representatives Approach-1

Input: K number of Clusters obtained from ***Unsup K-Means(D, K)***.
Output: Scaled Dataset S.

OptScaleb(K)
for $i \leftarrow 1, K$ **do**
  Consider C_i as a centre and draw the circle with threshold (T)distance.
  Select the data points d_i which not lies in radius(T).
  if $Dist(d_i) \geq T$ **then**
    $S = S \cup \{d_i\}$
  end if
end for
for $i \leftarrow 1, K$ **do**
  Add the centroid instance of cluster i in reduced set.
  $S = S \cup \{C_i\}$
end for

Opting representatives Approach-2

As the centroids are the main contributors of the clusters (Han and Kamber 2006), concentration is done on centers of clusters to minimize the dataset. In this algorithm, a large number of clusters formed in *sup K-means* algorithm is used for selecting representative instances. Then, centroids which best represents the clusters are chosen. Here, the number of clusters is more and selects only centroids of clusters as a representative. Instead of forming large clusters, many small clusters are formed. This simple but effective approach is summarized in Algorithm 5.

Algorithm 5 Opting Representatives Approach-2

Input: K number of Clusters obtained from ***Sup K-Means(D, K)***.
Output: Scaled Dataset S.

OptScalec(K)
for $centroid(i) \leftarrow 1 to K$ **do**
  Add the centroid instance of cluster i in reduced set.
  $S = S \cup \{C_i\}$
end for

Decision tree construction with scaled data

This is the last step in the DTCAP algorithm. In the second step, significant instances from a large dataset are obtained using Opting Representatives Approach-1 and Opting Representatives Approach-2. Condensed data obtained from the previous step is small in size as well as informative. Previous steps provide preprocessed data which is highly suitable for the supervised decision tree classifier. It constructs C4.5 (Quinlan 1993) decision tree using scaled Data S. The C4.5 algorithm is used because it is one of the best known and most widely used classification algorithms whose fragrance always lies in its accuracy and efficient performance. C4.5 deals with both numerical and categorical data. But DTCAP algorithm is only implemented for numerical attributes. The C4.5 algorithm is already depicted in Sect. 3.1.

The entire DTCAP algorithm depicts the various combinations of clustering and selection of instances from the border. DTCAP algorithm is divided into two types, first is *Unsupervised DTCAP* and *supervised DTCAP*. In the *Unsupervised DTCAP(D, K)* algorithm, the first *UnSup K-Means* algorithm is applied to data without labels or classes to obtain clusters of the entire dataset. Then *OptScaleb* algorithm to select informative instances is applied. In the *Supervised DTCAP(D,K)* algorithm, the first *Sup K-Means* algorithm is applied to the data. This dataset is preprocessed by dividing the dataset according to classes followed by *Sup K-Means* to find many small clusters. Then, the *OptScalec* algorithm is applied to select centers of these small clusters as representatives of a large dataset. The entire DTCAP algorithm is summarized in Algorithm 6.

Algorithm 6 Entire **DTCAP** Algorithm

Input: $D = \{d_1, d_2, d_3, \ldots .d_n\}$ be n data points
$K \leftarrow Numberofclusters$
$S \leftarrow Empty$
Output: Decision Tree DT based on Scaled Dataset S

Unsupervised DTCAP(D, K) .
1: *Input Data D as a unsupervised Data without classes.*
2: *Apply K-means*
K= ***Unsup K-Means(D, K)***
3: *Apply opting Representative approach-1*
S= ***OptScaleb(K)***
4: *Construct C4.5 DT*
DT= ***Construct_C4.5***(S)
5: *Evaluate Performance of DT*

Supervised DTCAP(D, K) .
1: *Input Data D as a supervised Data without classes.*
2: *Distribute D into j Classes*
D_j= j_{th} class Instances(D)
3: *Apply K-means*
K= ***Sup K-Means(D$_j$, K)***
4: *Apply opting Representative approach-2*
S= ***OptScalec(K)***
5: *Construct C4.5 DT*
DT= ***Construct_C4.5***(S)
6: *Evaluate the Performance of DT*

Results and discussion

In this work, the experiments are performed by the proposed approach on 9 numerical datasets from the UCI machine learning repository (Dua and Graff 2019) and best results are stressed through bold-face. The summary of the data is presented in Table 1. These datasets are applied to supervised and unsupervised clustering approaches to reduce dataset such that it will give better performance for decision tree. Various experiments are performed based on two proposed DTCAP method. Dataset is divided into 10 parts. Tenfold cross-validation is used where 9 parts are used for training and one part is used for testing. The C4.5 decision tree is constructed from scaled data for checking the performance of scaled data. The results are compared using three metrics prediction accuracy of the classifier, size of a tree, and the number of leaves. Accuracy is estimated by using the number of unseen instances classified on the trained tree classifier. Numeric dataset of different size is used which consist of a different number of features and classes.

First, the results of *Unsupervised DTCAP* on the above dataset are obtained. Table 2 shows a reduction of the number of instances when applying the *UnSup K-Means* algorithm followed by the *OptScaleb* algorithm on a different dataset. For large datasets like Segmentation, Waveform and Page Blocks, data is reduced to half. One of the important parameters in

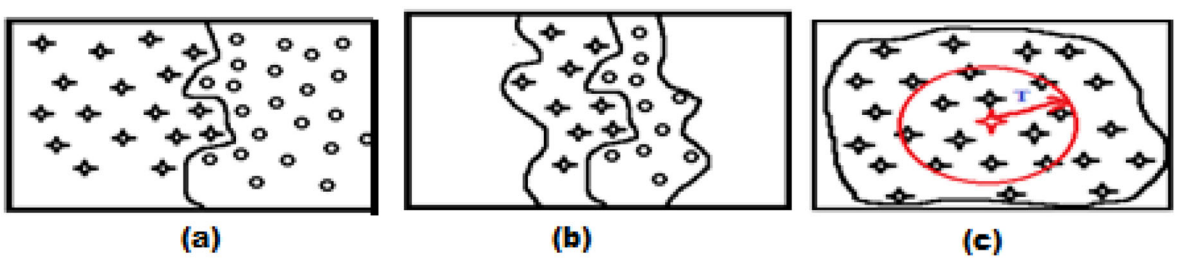

Fig. 1 **a** Two clusters with border. **b** Border and near to border representatives (Thicken Border). **c** How the selection of representatives is done

clustering is the number of clusters (K). To determine the number of clusters in the dataset, the elbow method (Thorndike 1953; Bailey 1994) found to be promising. The selection of the number of clusters is determined using the elbow method. The experiments are performed with a different number of clusters for each dataset. From experiments, it is observed that $K = 4$ gives significantly reduced data for Segment, Waveform, Sonar, Liver and Diabetes dataset. For Page Block dataset, K value is 8 while for Vehicle, Wine and Liver dataset 5 clusters are created in *UnSup K-Means*. These different numbers of clusters are finalized by the quality dataset produced which improves the decision tree. This major improvement decreases the training time of the decision tree without compromising the performance of the classifier. This reduced dataset is applied to the C4.5 decision tree for performance evaluation.

Table 1 Summary of data used in DTCAP

Datasets	Instances	Attributes	Classes
Wine	178	13	3
Sonar	208	60	2
Glass	214	9	6
Liver	345	6	2
Diabetes	768	8	2
Vehicle	846	18	4
Segmentation	2100	19	7
Waveform	5000	40	3
Page Blocks	5473	10	5

Table 2 Reduction of data in *Unsupervised DTCAP*

Datasets	Original number of instances	Reduced instances
Wine	178	100
Sonar	208	142
Glass	214	110
Liver	345	240
Diabetes	768	500
Vehicle	846	458
Segmentation	2100	1090
Waveform	5000	3290
Page Blocks	5473	2293

The accuracy of *Unsupervised DTCAP* is compared with the latest data reduction algorithms C4.5 (Quinlan 1993), IRB (Hernandez-Lea et al. 2013), DROP3 (Randall and Martinez 2000), CLU (Lumini and Nanni 2006), PSC (Olvera-López et al. 2010). From Table 3, it is observed that the proposed *unsupervised DTCAP* gives better accuracy on unseen instances as compared to other reduction techniques. Table 3 shows the results of the first proposed method *Unsupervised DTCAP* with non-reduced data constructed C4.5 (Quinlan 1993). Figure 2 shows a Graphical comparison of results from Table 3. Figure 2 shows the major improvement in the accuracy of the decision tree compared to other reduction methods. It shows that the performance of CLU is very low as compared to *Unsupervised DTCAP* and other methods.

The size of the classifier is one of the important aspects to choose a classifier for knowledge discovery (Quinlan 1993). If the size of the tree is large, then it requires more time for decision making or predicting class for unseen instances. The size of a tree is calculated by using the number of nodes in the tree. Both methods not only decrease the size of a tree but also improve the performance of it for unseen data. *Unsupervised DTCAP* algorithm tested for various measures like accuracy, number of leaves and size of tree based on *Unsup K-Means(D,K)* along with *OptScaleb(D,K)*. Table 4 shows the results of 9

Table 3 Comparison of accuracy of *Unsupervised DTCAP* with other methods

Dataset	Original (C4.5)	IRB	DROP3	CLU	PSC	Proposed
Liver	63.67	64.64	59.48	54.19	63.67	**70.28**
Vehicle	73.80	66.00	57.40	58.80	**74.00**	73.86
Sonar	72.57	75.32	73.45	56.73	77.45	**77.93**
Segment	96.02	90.90	83.57	87.37	89. 10	**97.16**
Wine	94.44	91.50	84.43	75.55	90.77	**94.94**
Glass	67.29	59.74	60. 19	55.58	60.58	**68.69**

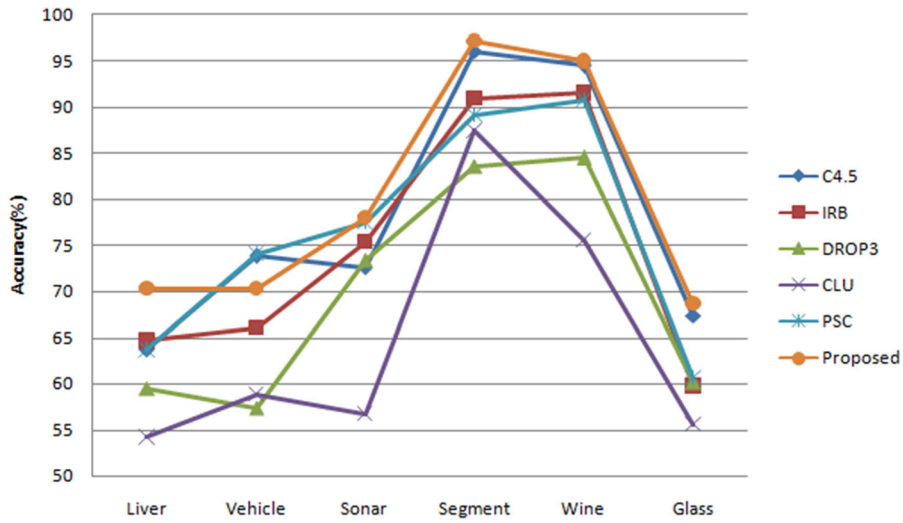

Fig. 2 Comparison of accuracy of *unsupervised DTCAP* with other methods

Table 4 Accuracy, Size and leaves comparison of C4.5 (without reduction) with the proposed *Unsupervised DTCAP* method

Dataset	Method	No. of instances	Accuracy	No. of leaves	Size of tree
Diabetes	C4.5	768	73.83	20	39
	Proposed	**615**	**77.11**	**16**	**31**
Liver	C4.5	345	63.67	26	51
	Proposed	**286**	**70.28**	**17**	**33**
Vehicle	C4.5	846	73.80	98	195
	Proposed	**765**	**73.86**	**60**	**119**
Sonar	C4.5	208	72.57	18	35
	Proposed	**156**	**77.93**	**12**	**23**
Segment	C4.5	1500	95.73	34	67
	Proposed	**950**	**97.16**	**16**	**31**
Waveform	C4.5	5000	75.08	330	659
	Proposed	**3980**	**76.25**	**276**	**551**
Page Blocks	C4.5	5473	96.88	44	87
	Proposed	**3036**	**97.47**	**27**	**53**
Wine	C4.5	178	93.28	5	9
	Proposed	**64**	**94.94**	5	9
Glass	C4.5	214	66.82	30	59
	Proposed	**108**	**68.69**	**16**	**31**

datasets for these different measures. The results of *Unsupervised DTCAP* are compared with the C4.5 decision tree without the reduction of datasets. From Table 4, it is observed that the size of a tree for every dataset is less than C4.5. The observed results from Fig. 3 show that as compared to the C4.5 with the whole dataset, the proposed method gives comparable accuracy. As shown in Fig. 3, for the diabetes dataset, the prediction accuracy is more using *Unsupervised DTCAP* as compared to C4.5 with original data. The reduced Liver dataset also improves the performance of the C4.5 decision tree. It decreases the size and number of leaves of the tree half than C4.5. The number of leaves is one of the important parameters in

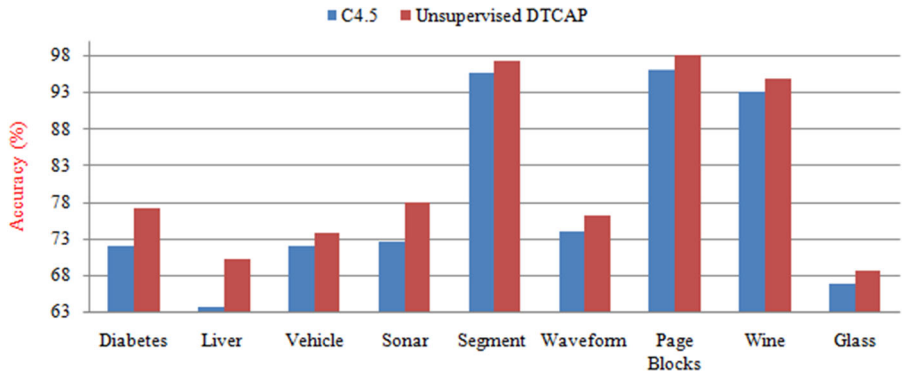

Fig. 3 Comparison of accuracy of C4.5 (without reduction) with the proposed *Unsupervised DTCAP* method

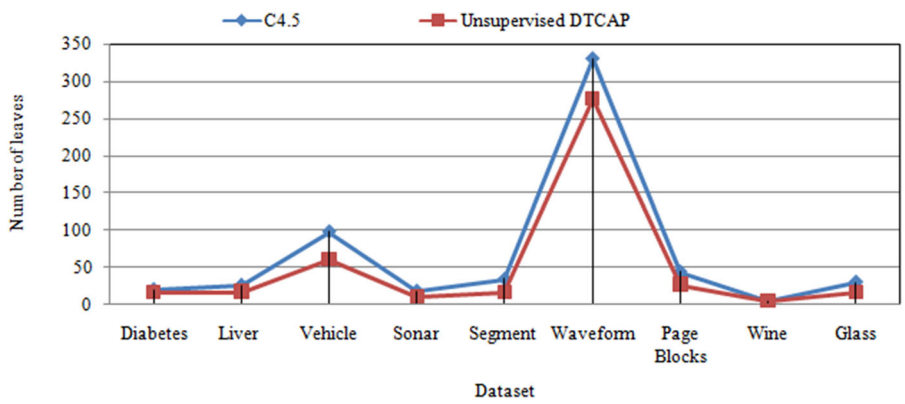

Fig. 4 Comparison of number of leaves in C4.5 (without reduction) with the proposed *Unsupervised DTCAP* method

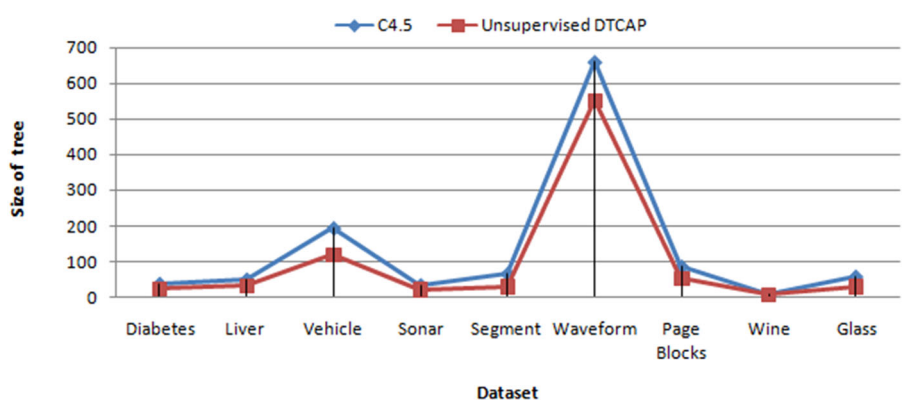

Fig. 5 Comparison of Tree size C4.5 (without reduction) with the proposed *Unsupervised DTCAP* method

decision making. If the number of leaves increased, then testing time is also increased. Figure 4 shows how the number of leaves gets decreased on scaled datasets obtained by *Unsupervised DTCAP*. This algorithm produces a fewer number of leaves for all datasets as shown in Fig. 4. Scaled data obtained from *Unsupervised DTCAP* creates optimized trees with reduced size. Comparison of size of tree for reduced dataset compared to the original dataset tree is shown in Fig. 5. It shows that an optimized tree created for the Vehicle and Waveform dataset. The results show that the accuracy of the C4.5 decision tree constructed

Table 5 Accuracy, Size and leaves comparison of C4.5 (without reduction) with the proposed *Supervised DTCAP* method

Dataset	Method	No. of instances	Accuracy	No. of leaves	Size of tree
Diabetes	C4.5	768	73.83	20	39
	Proposed	**500**	**76.40**	**14**	**27**
Liver	C4.5	345	63.67	26	51
	Proposed	**240**	**70**	**20**	**39**
Vehicle	C4.5	846	73.80	98	195
	Proposed	**480**	**73.54**	**51**	**101**
Sonar	C4.5	208	72.57	18	35
	Proposed	**130**	**73.07**	**8**	**15**

using the proposed method is better as compared to other methods. The main feature lies in the selection of data with thicken border data along with centroid instances.

Table 5 shows the results of the second proposed method *sup K-Means(D,K)* along with *OptScalec(D,K)* to find scaled data. The main idea is that the forming of small size clusters on supervised data can group correlated data and achieves more reduction. This reduced data increases computational accuracy. The *Supervised DTCAP* method is explained using the Diabetes dataset which consists of 768 instances. The diabetes dataset consists of 2 classes positive and negative. Out of 768 instances, 500 instances are of negative class and 268 instances are of the positive class. 768 instances are divided according to class. For 500 negative instances, *sup K-Means(D,K)* with *K* value as 200 is applied. Then, centers of 200 clusters are chosen using *OptScalec(D,K)* method as representatives. The same procedure is followed for positive class instances of the Diabetes dataset. Then, combined representative instances are applied to the decision tree to measure performance.

Figure 6 shows the performance comparison of *Supervised DTCAP* on a different dataset. It illustrates that the Supervised method gives better performance for Diabetes and Sonar datasets. As shown in Table 5, this method performs well for a small and medium-size dataset. Results of *Supervised DTCAP* show that there is considerable improvement of the decision tree in terms of prediction accuracy, size and number of leaves. The algorithm gives a comparable performance on the datasets like diabetes, liver, sonar and vehicle. The results of proposed approaches *OptScaleb(D,K)*and *OptScalec(D,K)* with different UCI datasets are obtained and compared with the other data reduction techniques. The experiments can be performed on the categorical datasets also. The main achievement of research work is that the methods proposed with aim of pre-processing data for decision

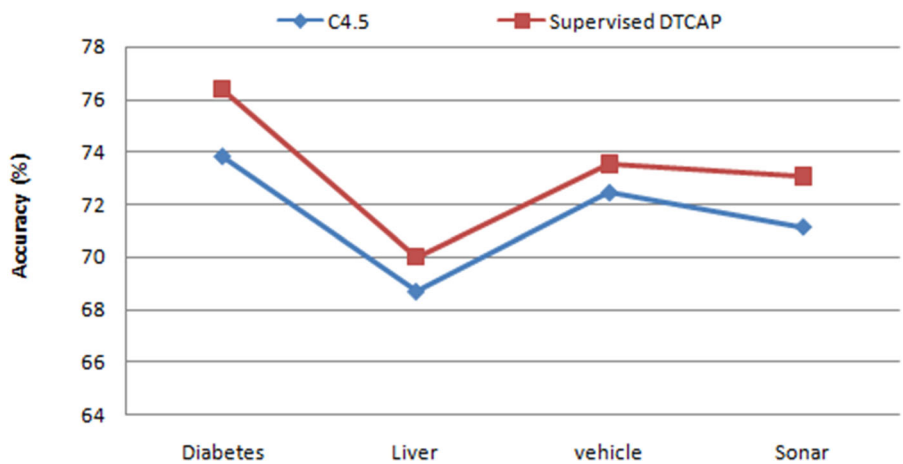

Fig. 6 Accuracy comparison of C4.5 (without reduction) with the proposed *Supervised DTCAP* method

tree are simple, easy and effective as compared to existing methods. Both methods *(Un Supervised DTCAP and Supervised DTCAP)* aims to improve decision tree classifiers. It creates efficient decision trees as compared to the original data set trained decision trees.

Conclusions

Decision tree classifier works better on the preprocessed data in terms of computational accuracy and size of the dataset obtained from the UCI dataset. The main aim of the research is to reduce the dataset which increases the performance of the C4.5 decision tree in terms of time and space. In this paper, two innovative, simple and easy techniques for selecting instances from the datasets are proposed which not only increases the accuracy of the C4.5 decision tree but also reduces its size. The *Unsupervised DTCAP* algorithm first forms clusters of unsupervised data and opt for the representative thicken border instances. The *Supervised DTCAP* algorithm first forms a large number of small clusters of supervised data and opt the centroid instances. From experimental work, it is concluded that the proposed *Supervised and Unsupervised DTCAP* gives the reduced dataset which increases the prediction accuracy of the decision tree from 1 to 9 %. The scaled data will reduce the building time and size of a decision tree.

As the future work we propose to use other supervised data correlation techniques to scale big data and apply scaled to improve performance of decision tree forest.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Alvar AS, Abadeh MS (2016) Efficient instance selection algorithm for classification based on fuzzy frequent patterns. In: 2016 IEEE 17th international symposium on computational intelligence and informatics (CINTI), pp 000319–000324

Angiulli F (2005) Fast condensed nearest neighbor rule. In: Proceedings of the 22nd international conference on machine learning, pp 25–32

Bailey K (1994) Numerical taxonomy and cluster analysis. In: Typologies and Taxonomies, vol 34, pp 24

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. CRC Press, Cambridge

Cavalcanti GD, Ren TI, Pereira CL (2013) ATISA: adaptive threshold-based instance selection algorithm. Expert Syst Appl 40(17):6894–6900

Chao S, Chen L (2005) Feature dimension reduction for microarray data analysis using locally linear embedding. In: Proceedings of the 3rd Asia-Pacific bioinformatics conference, vol 1, pp 211–217

Chen G, Cheng Y, Xu J (2008) Cluster reduction support vector machine for large-scale data set classification. In: 2008 IEEE Pacific-Asia workshop on computational intelligence and industrial application, vol 1, pp 8–12

Chou CH, Kuo BH, Chang F (2006) The generalized condensed nearest neighbor rule as a data reduction method. In: 18th international conference on pattern recognition (ICPR'06), vol 2. IEEE, pp 556–559

Czarnowski I (2012) Cluster-based instance selection for machine classification. Knowl Inf Syst 30(1):113–133

Dua D and Graff C (2019) UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA. http://archive.ics.uci.edu/ml

Gates G (1972) The reduced nearest neighbor rule (Corresp.). IEEE Trans Inf Theory 18(3):431–433

Han J, Kamber M (2006) Data mining: concepts and techniques, 3rd edn. Morgan Kaufmann Publishers, Burlington, pp 223–357

Hart P (1968) The condensed nearest neighbor rule (Corresp.). IEEE Trans Inf Theory 14(3):515–516

Hernandez-Lea P, Carrasco-Ochoa JA, Martínez-Trinidad JF, Olvera-Lopez JA (2013) InstanceRank based on borders for instance selection. Pattern Recognit 46(1):365–375

Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31(8):651–66

Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. Appl Stat 29:119–127

Lumini A, Nanni L (2006) A clustering method for automatic biometric template selection. Pattern Recognit 39:495–497

Marchiori E (2008) Hit miss networks with applications to instance selection. J Mach Learn Res 9:997–1017

Nikolaidis K, Goulermas JY, Wu QH (2011) A class boundary preserving algorithm for data condensation. Pattern Recognit 44(3):704–715

Olvera-López JA, Carrasco-Ochoa JA, Martínez-Trinidad JF (2010) A new fast prototype selection method based on clustering. Pattern Anal Appl 13(2):131–141

Ougiaroglou S, Evangelidis G (2012) Efficient dataset size reduction by finding homogeneous clusters. In: Proceedings of the fifth Balkan conference in informatics. ACM, pp 168–173

Pechenizkiy M, Tsymbal A, Puuronen S (2006) Local dimensionality reduction and supervised learning within natural clusters for biomedical data analysis. IEEE Trans Inf Technol Biomed 10(3):533–539

Peng K, Leung VC, Huang Q (2018) Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access 6:11897–11906

Phinyomark A, Pornchai P, Chusak L (2012) Feature reduction and selection for EMG signal classification. Expert Syst Appl 39(8):7420–7431

Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):119–127

Quinlan JR (1993) Programming for machine Learning. Morgan Kaufman, San Francisco

Randall WD, Martinez TR (2000) Reduction techniques for instance-based learning algorithms. Mach Learn 38(3):257–286

Sanguinetti G (2008) Dimensionality reduction of clustered data sets. IEEE Trans Pattern Anal Mach Intell 30(3):535–540

Sathyadevan S, Nair RR (2015) Comparative analysis of decision tree algorithms: ID3, C4.5 and random forest. In: Computational intelligence in data mining, vol 1. Springer, New Delhi, pp 549–562

Tang T, Chen S, Zhao M, Huang W, Luo J (2019) Very large-scale data classification based on K-means clustering and multi-kernel SVM. Soft Comput 23(11):3793–3801

Thorndike RL (1953) Who belongs in the family? Psychometrika 18:267–276. https://doi.org/10.1007/BF02289263

Toussaint GT, Foulsen RS (1979) Some new algorithms and software implementation methods for pattern recognition research. In: COMPSAC 79. Proceedings. Computer software and The IEEE computer society's third international applications conference, pp 59–63

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybernet 2(3):408–421

Yodjaiphet A, Theera-Umpon N, Auephanwiriyakul S (2015) Instance reduction for supervised learning using input-output clustering method. J Cent South Univ 22(12):4740–4748