A Novel Scalable and Effective Partitioning Approach for Big Data Reduction

Mohamed Malhat, Mohamed El-Menshawy, Hamdy Mousa, and Ashraf El-Sisi

Computer Science dept., Faculty of Computers and Information,
Menofia University, Egypt

{m.gmalhat@yahoo.com, Mohamed.elmenshawy@ci.menofia.edu.eg, hamdimmm@hotmail.com, ashrafelsisi@hotmail.com}

Abstract—The continuous increment of data size makes the traditional instance selection methods ineffective to reduce big training datasets in a single machine. Recent approaches to solving this technical problem partition the training dataset into subsets prior to apply the instance selection methods into each subset separately. However, the performance of the applied instance selection methods to subsets is negatively affected, especially when the number of partitioned subsets is increased. In this work, we propose a novel scalable and effective automated partitioning approach, called overlapped distance-based class-balance partitioning. This approach distributes the training dataset instances to the partitioned subsets based on a given distance metric and ensures the equal representation of data classes into partitioned subsets. Moreover, the instances might be assigned to two subsets once they satisfy the dynamic threshold. We implement and test empirically the scalability and effectiveness of the proposed approach using condensed nearest neighbor method over eight standard datasets. The proposed approach is compared empirically and analytically with stratification partitioning approach and a non-overlapped version from our approach with respect to 1) the reduction rate, classification accuracy, and effectiveness metrics, and 2) the scalability aspect, where the number of subsets is increased. The comparison results demonstrate that our approach is more scalable and effective than other partitioning approaches with respect to these standard datasets.

Keywords—Big data; Data Mining; Data Reduction; Instance Selection; Data Partitioning.

I. INTRODUCTION

The Knowledge Discovery in Databases (KDD) utilizes Data Mining (DM) algorithms and Machine Learning (ML) techniques to extract hidden patterns in raw data [1]. However, the DM algorithms and ML techniques are ineffective to process big data, generated in several application domains [2]. The term big data is used to describe the datasets that have the following characteristics: huge size (i.e., volume), rapid data generation (i.e., velocity), diversity of data types (i.e., variety), noisy and redundancy (i.e., veracity), and valuable patterns (i.e., value) [3-5]. Therefore, big raw data must be prepared prior to applying DM algorithms or ML techniques [6]. The preparing process is known as data preprocessing [1]. The data preprocessing contains a set of two methods: 1) data preparation; and 2) data reduction. The data preparation set contains the essential methods that must be applied to data in order to obtain applicable DM or ML results [2]. It specifically contains data cleaning methods, missing values imputation methods, noise identification methods, data transformation methods, data integration methods, and data normalization methods [1]. Called data reduction set contains a set of methods that can reduce data size via selecting most relevant instances (or features) and removing irrelevant data (e.g., noise, redundant, inconsistent, and superfluous) [7]. It contains instance selection methods, feature selection methods, and discretization methods. Instance Selection (IS) is the widest data reduction method used in the literature. The IS methods are used to search for the minimal subset of a given training dataset, which maintains the structure of the original training dataset [8]. The advantages of applying the IS methods in the KDD are decreasing the size of training dataset down, speeding up the mining or learning process, and improving the data quality [1, 8].

The continuous growth of data size makes the traditional IS methods unable to process training dataset in a single machine, due to memory limitations [9]. Therefore, new approaches are proposed that partition the training dataset into subsets and apply IS methods to each subset separately [10-12]. The approach in [10] uses random partitioning to partition a given training dataset into a group of manageable subsets. However, the performance of the applied IS method to the partitioned subsets is degraded, especially for class-imbalanced datasets. In order to overcome this limitation, the approaches in [11, 12] use stratification partitioning to ensure the equal distribution of data classes into subsets, while the instances of the same class are assigned randomly to subsets. The common feature of these approaches [10-12] is the random partitioning of the instances, which leads to a random representation of the instances in the partitioned subsets. This representation is insufficient for the employed IS method to get acceptable results, especially when highly scales up the number of subsets.

In this paper, we propose a novel scalable and effective automated approach for data partitioning called Overlapped Class-balance Distance-based Partitioning (OCDP). Our approach precisely focuses on improving the scalability and effectiveness of the employed IS methods applied to subsets. The scalability measures the ability of an IS method to obtain a good performance
regarding reduction rate and effectiveness, while the number of subsets is highly increased. The effectiveness measures the ability of an IS method to achieve a suitable balance between the reduction rate and classification accuracy metrics. The OCDP approach is an overlapped one as it allows instances to be assigned to two subsets if they satisfy the dynamic threshold. It is also a class-balance approach as it ensures the equal representation of data classes in the partitioned subsets. It is finally a distance-based approach as the instances are assigned to the nearest subset based on a given distance metric. The OCDP approach commences by finding the set of instances that belong to each class label in the training dataset. For each set of instances, it initializes the centroid of each subset by selecting a random instance. For each instance in the set of instances, the distances between the instance and the centroids of subsets are calculated. It then assigns the instance to the first nearest subset and updates the centroid of the subset. Finally, it checks if the instance may form a border between the first nearest subset and the second nearest subset using the dynamic threshold. If it is true, the instance is assigned to the second nearest subset.

We use the Condensed Nearest Neighbor (CNN) method [13] as an IS method to evaluate and test the scalability and effectiveness of the OCDP approach using eight standard datasets. In order to assess the importance of overlapping, we develop a non-overlapped version from our approach called Class-balance Distance-based Partitioning (CDP). We compare the OCDP approach with the stratification partitioning used in [11, 12] and the developed CDP approach in terms of 1) reduction rate, classification accuracy, and effectiveness, and 2) scalability aspect. Our experimental results prove that the OCDP approach has a better reduction rate and effectiveness results than the stratification and CDP approaches. Moreover, the OCDP approach is able to obtain a good effectiveness results compared to other approaches when the number of subsets is increased. Therefore, the OCDP approach is more scalable and effective than the stratification and CDP approaches. The work will continue as follows. In Section II, the related work and their limitations are discussed. In Section III, we introduce some notations employed in the OCDP approach presented in the same section. In Section IV, the experimental results are reported and analyzed. In Section V, we introduce the conclusion and identify the directions of future work.

II. RELATED WORK

The big data generated in many application domains makes the traditional IS methods unable to process such data in a single machine due to memory limitation. Therefore, contributed approaches have been put forward in the literature [10-12], which adopt a partitioning strategy on the top of IS methods, as shown in Fig. 1. This strategy consists of three main processes:

1. The training dataset is partitioned into \( m \) manageable subsets, wherein subset size is acceptable for a single machine to process (layer 1 in Fig. 1).
2. The employed IS method is applied to each subset separately (layer 2 in Fig. 1).
3. The results of subsets are accumulated together to form one reduced set (layer 3 in Fig. 1).

![Fig. 1. The strategy of partitioning datasets prior to apply IS methods to big data](image-url)
The main differences between these approaches are: (1) how to partition the training dataset; (2) which an IS method can apply to subsets; and (3) how results of subsets are accumulated. In [10], they partition a training dataset into subsets randomly and Steady State Memetic Algorithm - Scale Factor Local Search in Differential Evolution (SSMA-SFLSDE) [14], Learning Vector Quantization (LVQ3) [15], Reduction by Space Partitioning (RSP3) [16], Decremental Reduction Optimization Procedure (DROP3) [17], and Fast Condensed Nearest Neighbor (FCNN) [18] methods are applied to the partitioned subsets. They apply joining, filtering, and fusion techniques, when accumulating results of subsets to avoid adding irrelevant instances in the final reduced set. However, these techniques are unable to recover the relevant instances removed due to random partitioning. The approaches in [11, 12] use the stratification partitioning to ensure the equal class distribution in the partitioned subset to reduce the number of relevant instances removed. The authors in [11] apply DROP3 [17] and FCNN [18] methods to the partitioned subsets, while the authors in [12] apply Reduced Nearest Neighbor (RNN) [19] and Edited Nearest Neighbor (ENN) [20] methods to the partitioned subsets. The obtained results are better than random partitioning, but still the performance of the IS method over whole training dataset is better than the performance of the IS method over the partitioned training dataset.

The proposed approaches [10-12] overcome the memory limitation of a single machine. Moreover, they reduce the computation time of the reduction process. However, the performance of the employed IS method is negatively affected. The approaches, depending on the random partitioning of a training dataset, lead to a random representation of instances and classes in subsets, which are insufficient to obtain acceptable performance for the employed IS method [10]. The performance is obviously degraded when the number of subsets is increased. The stratification partitioning only ensures the equal representation of data classes, wherein instances of the same classes are assigned to subsets randomly [11, 12].

III. THE PROPOSED APPROACH

Partitioning a training dataset into subsets becomes a mandatory step before applying IS methods to overcome the scalability of data. In this section, we start by giving some notations that are needed for the rest of the paper. Our proposed approach to overcome the limitations mentioned in Section II is then introduced in details.

A. Notations

Our denoted notations are as follows:

1. \( L = \{l_1, l_2, ..., l_p\} \) is the class label set of \( p \) labels, where each label \( l_i \in L \) represents a given class label, used to classify the instances in the training dataset.

2. \( TR = \{x_1, x_2, ..., x_n\} \) is the original training dataset of \( n \) instances, where each instance \( x_i \in TR \) is a tuple of \( w \) features and a class label \( l_j \in L \) such that \( x_i = (x_{1i}, x_{2i}, ..., x_{wi}, l_j) \). The \( x_{fi} \) represents the \( f \)-th feature of the instance \( x_i \) and \( 1 \leq f \leq w \).

3. \( S = \bigcup_{i=1}^{m} S_i \) is the set of \( m \) subsets, where \( m \) is an integer number (1 \( \leq m \leq n \)). Note that each subset \( S_i \) is a set of \( t \) instances where \( t \) represents the number of instances that each subset can hold and calculates by using \( t = \frac{n}{m} \).

4. \( C = \{c_1, c_2, ..., c_m\} \) is the set of \( m \) centroids, where each subset \( S_i \subseteq S \) has a centroid \( c_i \in C \) and \( c_i \) is a tuple of \( w \) features.

5. \( D = \{d_1, d_2, ..., d_m\} \) is the set of \( m \) distances, where each distance \( d_i \in D \) represents the distance between any instance \( x \in TR \) and a centroid \( c_i \in C \).

6. \( E = \{e_1, e_2, ..., e_m\} \) is a set of \( m \) integers numbers initialized by zero, where each \( e_i \in E \) counts the number of instances \( x \in TR \) that added to subset \( S_i \subseteq S \).

7. \( cs: L \rightarrow 2^{TR} \) is a function that maps a given class label \( l_i \in L \) into a set \( CS \) of instances where \( CS \subseteq TR \). The set \( CS \) includes only the instances whose class label is \( l_i \). When \( CS = TR \), the instances of the training dataset have the same class label.

B. Automated Overlapped Class-Balance Distance-Based Partitioning (OCDP) Approach

The OCDP approach partitions a given training dataset \( TR \) into \( m \) overlapped subsets, while ensuring the (1) overlap border instances between subsets (i.e., overlapped approach); (2) equal representation of classes in subsets (i.e., class-balance approach); and (3) assignment of the instances to the nearest subset (i.e., distance-based approach). The complete steps of the OCDP approach are given in Algorithm 1. The OCDP approach accepts \( TR, m, \) and \( L \) as input and produces \( m \) overlapped subsets as output. It commences by entering a for loop over each class label \( l \in L \) (lines 2-21). In the loop, the OCDP approach initializes the centroid set \( C \) by assigning a random instance \( x \in cs(l) \) to each subset \( S \subseteq S \) (line 3), initializes \( E \) with \( m \) zero integers (line 4), and initializes \( subset_size \) variable with the number of instances that have class label \( l \) (\(|cs(l)|\)) divided by \( m \) (line 5). After
that, for each instance \( x \in \text{cs}(l) \) (lines 6-20), the distance set \( D \) is calculated. Each element \( d_y \in D \) is calculated using Equation 1 and represents a distance between instance \( x_i \) and centroid \( c_y \), where \( d_y \) and \( c_y \) are the \( y \) element in the \( D \) and \( C \) sets respectively.

\[
d_y(x_i, c_y) = \sum_{j=1}^{w} |x_{ij} - c_{yj}|
\]  

(1)

The indexes of the first three minimum distances in set \( D \) are assigned to variables \( \text{index}_a \) (line 8), \( \text{index}_b \) (line 9), and \( \text{index}_c \) (line 10), where \( 1 \leq a, b, c \leq m \). The instance \( x \) is added to the subset \( s_{\text{index}_a} \) (line 11) and \( e_{\text{index}_a} \) is incremented by one (line 12). The centroid \( c_{\text{index}_a} \) is updated to produce new centroid \( c'_{\text{index}_a} \) (line 13), such that each element \( c'_{\text{index}_a} \) is calculated using Equation 2, where \( x_{ij}, c_{\text{index}_a} \) are the \( j - \text{th} \) feature of \( x_i \) and \( c_{\text{index}_a} \) respectively.

\[
e_{aj}' = \frac{x_{ij} + c_{aj}}{2}
\]  

(2)

The dynamic overlapping threshold \( d_{\text{index}_b} - d_{\text{index}_a} \leq d_{\text{index}_c} - d_{\text{index}_b} \) is checked in lines 14-16, if true, instance \( x \) is added to \( s_{\text{index}_a} \) (line 15) with no update to \( c_{\text{index}_a} \) or \( e_{\text{index}_a} \). This threshold satisfies that instance \( x \) is closer to \( s_{\text{index}_b} \) than \( s_{\text{index}_c} \) and is a border instance between \( s_{\text{index}_a} \) and \( s_{\text{index}_b} \). In order to ensure that all subsets have approximately the same size (without counting overlapping), the condition \( e_{\text{index}_a} == \text{subset size} \) is checked (lines 17-19). If true, \( c_{\text{index}_a} \) is removed from \( C \) to not add any further instances to \( s_{\text{index}_a} \) (line 18). Finally, the resulted subsets \( S_i: \bigcup_{i=1}^{m} S_i = TR \) are returned in line 22.

**Algorithm 1** The Overlapped Class-balance Distance-based Partitioning (OCDP) algorithm

| Input | The training dataset \( TR \), the number \( m \) of subsets, and set \( L \) of class labels |
| Output: | The subsets \( S_i: \bigcup_{i=1}^{m} S_i = TR \) |

1. begin
2. \textbf{foreach} \( l \in L \) do
3. \hspace{1em} \( C \leftarrow \) load \( m \) random instances \( x \in \text{cs}(l) \)
4. \hspace{1em} \( E \leftarrow \) set of \( m \) integers initialized with zeros
5. \hspace{1em} \textbf{subset size} \( \leftarrow \left\lfloor \frac{|\text{cs}(l)|}{m} \right\rfloor \)
6. \hspace{1em} \textbf{foreach} \( x \in \text{cs}(l) \) do
7. \hspace{2em} \( D \leftarrow \) calculate distances between \( x \) and \( C \)
8. \hspace{2em} \( \text{index}_a \leftarrow \) get first minimum distance index from \( D \)
9. \hspace{2em} \( \text{index}_b \leftarrow \) get second minimum distance index from \( D \)
10. \hspace{2em} \( \text{index}_c \leftarrow \) get third minimum distance index from \( D \)
11. \hspace{2em} assign \( x \) to \( s_{\text{index}_a} \)
12. \hspace{2em} \( e_{\text{index}_a} \leftarrow e_{\text{index}_a} + 1 \)
13. \hspace{2em} update centroid \( c_{\text{index}_a} \)
14. \hspace{2em} \textbf{if} \( d_{\text{index}_b} - d_{\text{index}_a} \leq d_{\text{index}_c} - d_{\text{index}_b} \) \textbf{then}
15. \hspace{3em} assign \( x \) to \( s_{\text{index}_b} \)
16. \hspace{2em} \textbf{if} \( e_{\text{index}_a} == \text{subset size} \) \textbf{then}
17. \hspace{3em} remove centroid \( c_{\text{index}_a} \) from \( C \)
18. \hspace{2em} \textbf{end}
19. \hspace{1em} \textbf{end}
20. \hspace{1em} \textbf{end}
21. \textbf{return} \( S_i: \bigcup_{i=1}^{m} S_i = TR \)
22. end

The overlapping between subsets is only occurring for border instances. The dynamic threshold \( d_{\text{index}_b} - d_{\text{index}_a} \leq d_{\text{index}_c} - d_{\text{index}_b} \) is used to identify the border instances between the subsets. Fig. 2 shows an example for an instance \( x \) that satisfies the dynamic threshold. Therefore, the instance is added to the subset \( s_{\text{index}_a} \) and subset \( s_{\text{index}_b} \), and the centroid \( c_{\text{index}_a} \) is updated, while the centroid \( c_{\text{index}_b} \) is not updated, so as to reduce the number of overlapped instances. Intuitively, if we allow updating the centroid \( c_{\text{index}_b} \), it will move toward the subset \( s_{\text{index}_a} \). This moving will increase the instances that can be overlapped between subsets \( s_{\text{index}_a} \) and \( s_{\text{index}_b} \) in the next iterations. The instance violating the dynamic threshold is only added to the nearest subset \( s_{\text{index}_a} \) after updating the corresponding centroid \( c_{\text{index}_a} \) as shown in Fig. 3.
Regarding Fig. 1, we use the OCDP approach (i.e., first process) to partition a training dataset into \( m \) overlapped subsets and the employed IS method is applied into subsets separately (i.e., second process). After that, the results of subsets are collected together to form reduced set (i.e., third process). In order to remove redundant instances caused by overlapping, we add an extra process to Fig. 1, which applies the IS method again to the reduced set to get the final reduced set.

**IV. EXPERIMENTAL RESULTS**

We devote eight standard datasets from KEEL repository [21] in our experimentations. The number of instances (#instances), the number of features (#features), and the number of classes (#classes) for each dataset are given in Table 1. In order to validate and test the proposed approach, the instance selection CNN method is adopted [13]. The CNN method, in fact, belongs to the condensation family and is a standard method for the most contributed condensation methods in the literature (see, for example, TCNN [22], MCNN [23], GCNN [24], and FCNN [18]).

<table>
<thead>
<tr>
<th>Dataset name</th>
<th># instances</th>
<th># features</th>
<th># classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>7400</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Texture</td>
<td>5500</td>
<td>40</td>
<td>11</td>
</tr>
<tr>
<td>Opt-Digits</td>
<td>5620</td>
<td>64</td>
<td>10</td>
</tr>
<tr>
<td>Pen-Based</td>
<td>10992</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>Thyroid</td>
<td>7200</td>
<td>21</td>
<td>3</td>
</tr>
<tr>
<td>Phoneme</td>
<td>5404</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Sat-Image</td>
<td>6435</td>
<td>36</td>
<td>7</td>
</tr>
<tr>
<td>Magic</td>
<td>19020</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

The reduction rate (Red.), the classification accuracy (Acc.), and the effectiveness (Eff.) metrics are used to measure the performance of the CNN method applied to the partitioned subsets. The reduction rate is calculated using Equation 3, where \( |RS| \) is the number of instances in the reduced set and \( |TR| \) is the number of instances in the original training dataset.

\[
Red. = 1 - \frac{|RS|}{|TR|}
\]  

(3)

The classification accuracy is calculated using Equation 4, where \( true\_classified(TS) \) is the number of correctly classified instances in the testing dataset using \( RS \) and \( |TS| \) is the total number of instances in the testing dataset.

\[
Acc. = \frac{|true\_classified(TS)|}{|TS|}
\]  

(4)

The effectiveness is calculated as the product of the reduction rate and classification accuracy as given in Equation 5.
The 10-Fold Cross Validation (10-FCV) scheme is used to partition dataset into training dataset and testing dataset. The reported results are the average (Avg.) and standard deviation (Std.) results of 10-FCV. The k-NN classifier [25] with k=1 is used to assess the classification accuracy of the CNN method. Our OCDP approach is compared with the Stratification Partitioning (SP) [11, 12] and developed CDP approaches. The CDP approach follows the same procedure of OCDP, except it doesn’t allow instances to be overlapped into two subsets (lines 14-16 in Algorithm 1). We implemented the three approaches and the CNN method using Java. Our experiments are performed using the following specification: the processor is Intel(R) Core-i5, 2.5 GHz, 4 GB RAM, and Windows 7.

Table 2, Table 3, and Table 4 show the Avg. and Std. results of the reduction rate, classification accuracy, and effectiveness respectively over the eight employed datasets for the three partitioning approaches using different number of subsets (m), where m is 5, 10, 15, 20, and 25. From Table 2, the CDP approach has better reduction rate results than the SP approach for all employed datasets. For example, the reduction rate of the Texture dataset when m = 5 is 0.8869 for CDP and 0.8428 for SP. The CDP approach assigns the instances to the nearest subsets based on the defined distance in Equation 1. Therefore, similar instances are grouped in the same subset, which allows CNN method to maximize the reduction rate results. Our OCDP approach achieves a highly better reduction rate results than the SP and CDP approaches for all employed datasets. For example, the reduction rate results of the Phoneme dataset when m = 5 are 0.6765 for SP, 0.7463 for CDP, and 0.8250 for OCDP. The OCDP approach overlaps the border instances between two subsets besides grouping similar instances in the same subset. Therefore, the CNN method maintains only these border instances to classify the instances in the two subsets, which maximize the reduction rate results than the CDP approach. From Table 3, the SP and CDP approaches have slightly better classification accuracy results than the OCDP approach. This is because of the high reduction rate results achieved using OCDP. For example, the classification accuracy results of the Pen-Based dataset when m = 5 are 0.9886 for SP, 0.9793 for CDP, and 0.9661 for OCDP. The effectiveness measures the ability of an IS method to achieve the best trade-off between the reduction rate and classification accuracy metrics. The high effectiveness results are produced when we achieve good results in both metrics. The high reduction rate and low classification accuracy give low effectiveness and vice versa. Therefore, we take the effectiveness metric as a benchmark to compare the partitioning approaches. From Table 4, the effectiveness results of our OCDP approach are obviously better than the SP and CDP approach for the eight employed datasets. For example, the effectiveness results of the Thyroid dataset when m = 5 are 0.7416 for OCDP, 0.6818 for SP, and 0.6939 for CDP. From these results, we conclude that our OCDP approach is more effective (i.e., achieves the best trade-off between the reduction rate and classification accuracy metrics) than the SP and CDP approaches.

TABLE 2. Reduction rate results over the eight employed datasets using different number of subsets (m). Notice that the bolded numbers represent the best result for each dataset with respect to number of subsets.

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>m</th>
<th>SP Avg.</th>
<th>SP Std.</th>
<th>CDP Avg.</th>
<th>CDP Std.</th>
<th>OCDP Avg.</th>
<th>OCDP Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ring</td>
<td>5</td>
<td>0.6755</td>
<td>0.0059</td>
<td>0.6753</td>
<td>0.0072</td>
<td>0.8126</td>
<td>0.0053</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.6473</td>
<td>0.0074</td>
<td>0.6606</td>
<td>0.0043</td>
<td>0.8211</td>
<td>0.0060</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.6352</td>
<td>0.0067</td>
<td>0.6507</td>
<td>0.0079</td>
<td>0.8263</td>
<td>0.0046</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.6220</td>
<td>0.0040</td>
<td>0.6524</td>
<td>0.0053</td>
<td>0.8289</td>
<td>0.0072</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.6122</td>
<td>0.0053</td>
<td>0.6532</td>
<td>0.0062</td>
<td>0.8314</td>
<td>0.0034</td>
</tr>
<tr>
<td>Texture</td>
<td>5</td>
<td>0.8428</td>
<td>0.0038</td>
<td>0.8869</td>
<td>0.0051</td>
<td>0.9489</td>
<td>0.0022</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.7915</td>
<td>0.0028</td>
<td>0.8854</td>
<td>0.0066</td>
<td>0.9500</td>
<td>0.0023</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.7579</td>
<td>0.0041</td>
<td>0.8861</td>
<td>0.0067</td>
<td>0.9526</td>
<td>0.0028</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.7281</td>
<td>0.0036</td>
<td>0.8816</td>
<td>0.0059</td>
<td>0.9553</td>
<td>0.0038</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.7022</td>
<td>0.0041</td>
<td>0.8844</td>
<td>0.0063</td>
<td>0.9554</td>
<td>0.0019</td>
</tr>
<tr>
<td>Opt-Digits</td>
<td>5</td>
<td>0.8530</td>
<td>0.0023</td>
<td>0.8757</td>
<td>0.0047</td>
<td>0.9480</td>
<td>0.0016</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.8076</td>
<td>0.0038</td>
<td>0.8655</td>
<td>0.0041</td>
<td>0.9491</td>
<td>0.0025</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.7767</td>
<td>0.0034</td>
<td>0.8666</td>
<td>0.0035</td>
<td>0.9514</td>
<td>0.0020</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.7488</td>
<td>0.0044</td>
<td>0.8684</td>
<td>0.0038</td>
<td>0.9532</td>
<td>0.0024</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.7263</td>
<td>0.0045</td>
<td>0.8643</td>
<td>0.0035</td>
<td>0.9541</td>
<td>0.0020</td>
</tr>
<tr>
<td>Pen-Based</td>
<td>5</td>
<td>0.9110</td>
<td>0.0016</td>
<td>0.9324</td>
<td>0.0029</td>
<td>0.9711</td>
<td>0.0013</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.8777</td>
<td>0.0020</td>
<td>0.9314</td>
<td>0.0027</td>
<td>0.9729</td>
<td>0.0012</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.8532</td>
<td>0.0025</td>
<td>0.9331</td>
<td>0.0017</td>
<td>0.9740</td>
<td>0.0010</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.8335</td>
<td>0.0022</td>
<td>0.9335</td>
<td>0.0028</td>
<td>0.9753</td>
<td>0.0012</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.8168</td>
<td>0.0023</td>
<td>0.9352</td>
<td>0.0029</td>
<td>0.9761</td>
<td>0.0014</td>
</tr>
<tr>
<td>Thyroid</td>
<td>5</td>
<td>0.7835</td>
<td>0.0053</td>
<td>0.8065</td>
<td>0.0074</td>
<td>0.8703</td>
<td>0.0084</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.7720</td>
<td>0.0058</td>
<td>0.8240</td>
<td>0.0080</td>
<td>0.8829</td>
<td>0.0076</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.7666</td>
<td>0.0048</td>
<td>0.8469</td>
<td>0.0056</td>
<td>0.8971</td>
<td>0.0055</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.7645</td>
<td>0.0053</td>
<td>0.8654</td>
<td>0.0070</td>
<td>0.8987</td>
<td>0.0024</td>
</tr>
<tr>
<td>Dataset name</td>
<td>m</td>
<td>Avg.</td>
<td>Std.</td>
<td>Avg.</td>
<td>Std.</td>
<td>Avg.</td>
<td>Std.</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>Ring</td>
<td></td>
<td>5</td>
<td>0.8330</td>
<td>0.01105</td>
<td>0.8504</td>
<td>0.0156</td>
<td>0.8328</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8311</td>
<td>0.00346</td>
<td>0.8349</td>
<td>0.0061</td>
<td>0.8243</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.8322</td>
<td>0.01110</td>
<td>0.8341</td>
<td>0.0076</td>
<td>0.8234</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.8403</td>
<td>0.01106</td>
<td>0.8278</td>
<td>0.0086</td>
<td>0.8300</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.8326</td>
<td>0.01111</td>
<td>0.8292</td>
<td>0.0093</td>
<td>0.8232</td>
</tr>
<tr>
<td>Texture</td>
<td></td>
<td>5</td>
<td>0.9822</td>
<td>0.00497</td>
<td>0.9705</td>
<td>0.0086</td>
<td>0.9480</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.9800</td>
<td>0.00352</td>
<td>0.9680</td>
<td>0.0054</td>
<td>0.9482</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.9856</td>
<td>0.00357</td>
<td>0.9660</td>
<td>0.0070</td>
<td>0.9475</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.9862</td>
<td>0.00555</td>
<td>0.9644</td>
<td>0.0075</td>
<td>0.9482</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.9876</td>
<td>0.00291</td>
<td>0.9669</td>
<td>0.0056</td>
<td>0.9465</td>
</tr>
<tr>
<td>Opt-Digits</td>
<td></td>
<td>5</td>
<td>0.9781</td>
<td>0.00515</td>
<td>0.9767</td>
<td>0.0086</td>
<td>0.9464</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.9788</td>
<td>0.00400</td>
<td>0.9744</td>
<td>0.0066</td>
<td>0.9486</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.9827</td>
<td>0.00359</td>
<td>0.9733</td>
<td>0.0064</td>
<td>0.9464</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.9820</td>
<td>0.00755</td>
<td>0.9753</td>
<td>0.0070</td>
<td>0.9502</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.9847</td>
<td>0.00446</td>
<td>0.9733</td>
<td>0.0066</td>
<td>0.9539</td>
</tr>
<tr>
<td>Pen-Based</td>
<td></td>
<td>5</td>
<td>0.9886</td>
<td>0.00311</td>
<td>0.9793</td>
<td>0.0065</td>
<td>0.9623</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.9886</td>
<td>0.00297</td>
<td>0.9796</td>
<td>0.0054</td>
<td>0.9623</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.9884</td>
<td>0.00307</td>
<td>0.9773</td>
<td>0.0048</td>
<td>0.9632</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.9885</td>
<td>0.00297</td>
<td>0.9753</td>
<td>0.0068</td>
<td>0.9605</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.9899</td>
<td>0.00223</td>
<td>0.9759</td>
<td>0.0031</td>
<td>0.9593</td>
</tr>
<tr>
<td>Thyroid</td>
<td></td>
<td>5</td>
<td>0.8701</td>
<td>0.00389</td>
<td>0.8582</td>
<td>0.0139</td>
<td>0.8521</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8542</td>
<td>0.01021</td>
<td>0.8415</td>
<td>0.0133</td>
<td>0.8329</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.8575</td>
<td>0.00889</td>
<td>0.8169</td>
<td>0.0164</td>
<td>0.7985</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.8519</td>
<td>0.01061</td>
<td>0.8024</td>
<td>0.0270</td>
<td>0.7940</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.8440</td>
<td>0.01515</td>
<td>0.8018</td>
<td>0.0238</td>
<td>0.7960</td>
</tr>
<tr>
<td>Phoneme</td>
<td></td>
<td>5</td>
<td>0.8692</td>
<td>0.01666</td>
<td>0.8483</td>
<td>0.0181</td>
<td>0.8396</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8705</td>
<td>0.00900</td>
<td>0.8161</td>
<td>0.0126</td>
<td>0.8198</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.8671</td>
<td>0.00871</td>
<td>0.8035</td>
<td>0.0187</td>
<td>0.8122</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.8638</td>
<td>0.01100</td>
<td>0.7874</td>
<td>0.0183</td>
<td>0.8050</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.8666</td>
<td>0.01545</td>
<td>0.7755</td>
<td>0.0280</td>
<td>0.8038</td>
</tr>
<tr>
<td>Sat-Image</td>
<td></td>
<td>5</td>
<td>0.8838</td>
<td>0.00933</td>
<td>0.8741</td>
<td>0.0094</td>
<td>0.8605</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.8842</td>
<td>0.00646</td>
<td>0.8634</td>
<td>0.0089</td>
<td>0.8547</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.8841</td>
<td>0.00646</td>
<td>0.8570</td>
<td>0.0137</td>
<td>0.8429</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.8838</td>
<td>0.00900</td>
<td>0.8521</td>
<td>0.0104</td>
<td>0.8410</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.8824</td>
<td>0.01211</td>
<td>0.8472</td>
<td>0.0089</td>
<td>0.8421</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>0.7695</td>
<td>0.01081</td>
<td>0.7668</td>
<td>0.0117</td>
<td>0.7403</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.7673</td>
<td>0.00888</td>
<td>0.7614</td>
<td>0.0105</td>
<td>0.7305</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>0.7655</td>
<td>0.01115</td>
<td>0.7556</td>
<td>0.0110</td>
<td>0.7256</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>0.7675</td>
<td>0.01228</td>
<td>0.7482</td>
<td>0.0125</td>
<td>0.7208</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25</td>
<td>0.7675</td>
<td>0.01430</td>
<td>0.7440</td>
<td>0.0114</td>
<td>0.7217</td>
</tr>
</tbody>
</table>
Since the CNN method is one of the condensation methods, then its main motivation is to improve the reduction rate while not degrading the effectiveness. Therefore, we analyze the scalability on the reduction rate and effectiveness metrics when the number of subsets is increased. The scalability is analyzed using 5, 10, 15, 20, and 25 subsets. The reduction rate results of the CNN method using the SP, CDP, and OCDP approaches over the eight employed datasets are demonstrated in Fig. 4. The reduction rate results of the SP approach are extremely decreased when the number of subsets is increased. For example, the reduction rate results of the Texture dataset are 0.8428 when \( m = 5 \) and 0.7022 when \( m = 25 \). The CDP and OCDP approaches maintain or may improve the reduction rate results when number of subsets is increased. For example, the reduction rate results of the Pen-based dataset are 0.9321 and 0.9352 for CDP when \( m = 5 \) and \( m = 25 \) respectively and 0.9711 and 0.9761 for OCDP when \( m = 5 \) and \( m = 25 \) respectively. Finally, the effectiveness results of the SP approach are obviously decreased when the number of subsets is increased for all employed datasets, as shown in Fig. 5. For example, the effectiveness results of the Ring dataset are 0.5627 when \( m = 5 \) and 0.5097 when \( m = 25 \). The CDP and OCDP approaches maintain or may increase the effectiveness results when number of subsets is increased. For example, the effectiveness results of the Magic dataset are 0.4708 and 0.5247 for CDP when \( m = 5 \) and \( m = 25 \) respectively and 0.5429 and 0.5623 for OCDP when \( m = 5 \) and \( m = 25 \) respectively. From these comparison results, we conclude that the CDP and OCDP approaches are more robust and scalable against high number of subsets than the SP approach, but our OCDP approach has better reduction rate and effectiveness results than the CDP approach.

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>m</th>
<th>Avg.</th>
<th>Std.</th>
<th>Avg.</th>
<th>Std.</th>
<th>Avg.</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SP</td>
<td></td>
<td>CDP</td>
<td></td>
<td>OCDP</td>
<td></td>
</tr>
<tr>
<td>Ring</td>
<td>5</td>
<td>0.5627</td>
<td>0.0099</td>
<td>0.5608</td>
<td>0.0139</td>
<td>0.6767</td>
<td>0.0118</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.5379</td>
<td>0.0070</td>
<td>0.5515</td>
<td>0.0045</td>
<td>0.6768</td>
<td>0.0077</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.5286</td>
<td>0.0096</td>
<td>0.5427</td>
<td>0.0084</td>
<td>0.6804</td>
<td>0.0096</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.5226</td>
<td>0.0054</td>
<td>0.5400</td>
<td>0.0066</td>
<td>0.6879</td>
<td>0.0097</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.5097</td>
<td>0.0099</td>
<td>0.5416</td>
<td>0.0073</td>
<td>0.6845</td>
<td>0.0080</td>
</tr>
<tr>
<td>Texture</td>
<td>5</td>
<td>0.5278</td>
<td>0.0345</td>
<td>0.8608</td>
<td>0.0079</td>
<td>0.8996</td>
<td>0.0075</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.7756</td>
<td>0.0499</td>
<td>0.8570</td>
<td>0.0062</td>
<td>0.9007</td>
<td>0.0077</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.7470</td>
<td>0.0440</td>
<td>0.8559</td>
<td>0.0084</td>
<td>0.9026</td>
<td>0.0061</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.7181</td>
<td>0.0042</td>
<td>0.8502</td>
<td>0.0080</td>
<td>0.9058</td>
<td>0.0087</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.6935</td>
<td>0.0043</td>
<td>0.8551</td>
<td>0.0084</td>
<td>0.9043</td>
<td>0.0128</td>
</tr>
<tr>
<td>Opt-Digits</td>
<td>5</td>
<td>0.8543</td>
<td>0.0033</td>
<td>0.8555</td>
<td>0.0093</td>
<td>0.8972</td>
<td>0.0052</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.7904</td>
<td>0.0037</td>
<td>0.8433</td>
<td>0.0057</td>
<td>0.9003</td>
<td>0.0051</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.7633</td>
<td>0.0046</td>
<td>0.8434</td>
<td>0.0076</td>
<td>0.9005</td>
<td>0.0088</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.7353</td>
<td>0.0074</td>
<td>0.8469</td>
<td>0.0050</td>
<td>0.9057</td>
<td>0.0093</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.7152</td>
<td>0.0052</td>
<td>0.8412</td>
<td>0.0074</td>
<td>0.8948</td>
<td>0.0117</td>
</tr>
<tr>
<td>Pen-Based</td>
<td>5</td>
<td>0.9046</td>
<td>0.0032</td>
<td>0.9130</td>
<td>0.0065</td>
<td>0.9381</td>
<td>0.0067</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.8677</td>
<td>0.0037</td>
<td>0.9124</td>
<td>0.0060</td>
<td>0.9362</td>
<td>0.0057</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.8433</td>
<td>0.0037</td>
<td>0.9120</td>
<td>0.0056</td>
<td>0.9381</td>
<td>0.0059</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.8240</td>
<td>0.0034</td>
<td>0.9104</td>
<td>0.0062</td>
<td>0.9384</td>
<td>0.0072</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.8086</td>
<td>0.0034</td>
<td>0.9126</td>
<td>0.0035</td>
<td>0.9364</td>
<td>0.0061</td>
</tr>
<tr>
<td>Thyroid</td>
<td>5</td>
<td>0.6818</td>
<td>0.0069</td>
<td>0.6939</td>
<td>0.0179</td>
<td>0.7416</td>
<td>0.0232</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.6594</td>
<td>0.0085</td>
<td>0.6934</td>
<td>0.0200</td>
<td>0.7355</td>
<td>0.0266</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.6574</td>
<td>0.0078</td>
<td>0.6918</td>
<td>0.0123</td>
<td>0.7163</td>
<td>0.0173</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.6513</td>
<td>0.0089</td>
<td>0.6943</td>
<td>0.0217</td>
<td>0.7136</td>
<td>0.0110</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.6434</td>
<td>0.0104</td>
<td>0.7018</td>
<td>0.0171</td>
<td>0.7236</td>
<td>0.0255</td>
</tr>
<tr>
<td>Phoneme</td>
<td>5</td>
<td>0.5879</td>
<td>0.0113</td>
<td>0.6328</td>
<td>0.0114</td>
<td>0.6927</td>
<td>0.0108</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.5603</td>
<td>0.0064</td>
<td>0.6552</td>
<td>0.0176</td>
<td>0.6992</td>
<td>0.0140</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.5477</td>
<td>0.0073</td>
<td>0.6706</td>
<td>0.0189</td>
<td>0.6947</td>
<td>0.0103</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.5359</td>
<td>0.0086</td>
<td>0.6834</td>
<td>0.0158</td>
<td>0.7002</td>
<td>0.0162</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.5283</td>
<td>0.0095</td>
<td>0.6816</td>
<td>0.0208</td>
<td>0.7009</td>
<td>0.0143</td>
</tr>
<tr>
<td>Sat-Image</td>
<td>5</td>
<td>0.6587</td>
<td>0.0073</td>
<td>0.7030</td>
<td>0.0068</td>
<td>0.7413</td>
<td>0.0080</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.6380</td>
<td>0.0074</td>
<td>0.7103</td>
<td>0.0093</td>
<td>0.7477</td>
<td>0.0103</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.6233</td>
<td>0.0063</td>
<td>0.7231</td>
<td>0.0131</td>
<td>0.7445</td>
<td>0.0100</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.6132</td>
<td>0.0074</td>
<td>0.7284</td>
<td>0.0054</td>
<td>0.7489</td>
<td>0.0106</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.6028</td>
<td>0.0096</td>
<td>0.7337</td>
<td>0.0088</td>
<td>0.7532</td>
<td>0.0070</td>
</tr>
<tr>
<td>Magic</td>
<td>5</td>
<td>0.4673</td>
<td>0.0052</td>
<td>0.4708</td>
<td>0.0085</td>
<td>0.5429</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.4583</td>
<td>0.0050</td>
<td>0.4784</td>
<td>0.0096</td>
<td>0.5499</td>
<td>0.0084</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.4497</td>
<td>0.0078</td>
<td>0.4906</td>
<td>0.0101</td>
<td>0.5524</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.4470</td>
<td>0.0084</td>
<td>0.5073</td>
<td>0.0073</td>
<td>0.5566</td>
<td>0.0085</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.4421</td>
<td>0.0079</td>
<td>0.5274</td>
<td>0.0093</td>
<td>0.5623</td>
<td>0.0095</td>
</tr>
</tbody>
</table>
Fig. 4. The reduction rate of the eight employed datasets for the three partitioning approaches with a different number of subsets.
V. CONCLUSION AND FUTURE WORK

The traditional IS methods are unable to handle the big data resulted from many application fields due to memory limitation of a single machine. Other contributed approaches in the literature proposed recently to partition training dataset (either using random or stratification manner) into manageable subsets and apply the IS methods to these subsets individually. Therefore, they enable IS methods to overcome the size of big data. However, the performance of the employed IS methods is negatively affected due to partitioning, especially when the number of partitioned subsets is highly increased. The main contribution lies in proposing a novel scalable and effective automated approach for partitioning a training dataset. The instances are allocated to the nearest subsets based on the defined distance measure while ensuring the equal representation of classes in subsets. The instances might be overlapped in two subsets only if it satisfies the dynamic threshold. We compare the proposed approach with the SP and CDP approaches using eight standard datasets and the CNN method with respect to the reduction rate, classification accuracy, and effectiveness metrics. The results prove that our OCDP approach has the ability to achieve a better reduction rate and effectiveness results than the SP and CDP approaches. Moreover, our OCDP approach maintains a good reduction rate and effectiveness results when the number of subsets is increased. In future work, we plan to analyze the impact of our OCDP approach on other IS methods that belong to other families, such as ENN [20] and iterative case filtering [26] methods.
REFERENCES