

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).

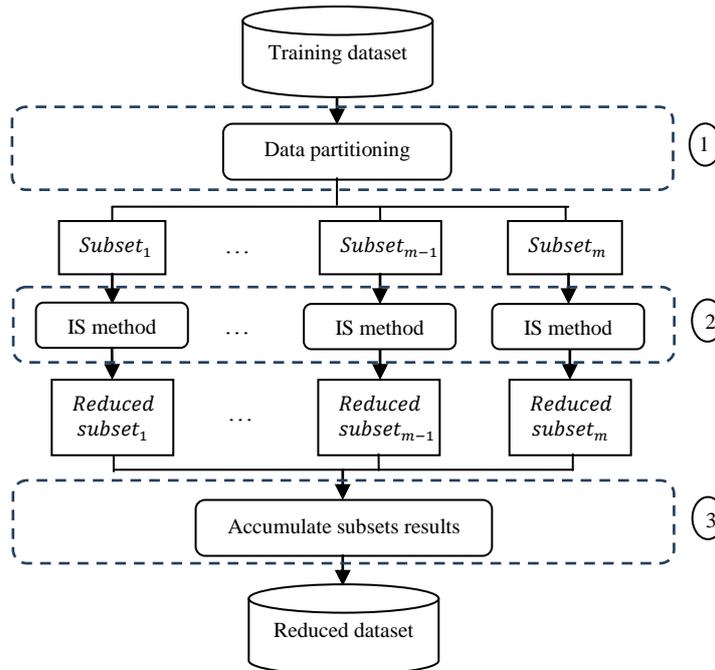


Fig. 1. The strategy of partitioning datasets prior to apply IS methods to big data

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. $L = \{l_1, l_2, \dots, l_p\}$ 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. $TR = \{x_1, x_2, \dots, x_n\}$ 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 $x_i = (x_{i1}, x_{i2}, \dots, x_{iw}, l_j)$. The x_{if} represents the f -th feature of the instance x_i and $1 \leq f \leq w$.
3. $S = \cup_{i=1}^m s_i$ 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 $t = \lfloor \frac{n}{m} \rfloor$.
4. $C = \{c_1, c_2, \dots, c_m\}$ 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. $D = \{d_1, d_2, \dots, d_m\}$ 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. $E = \{e_1, e_2, \dots, e_m\}$ 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 $x \in cs(l)$ (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.

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

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

$$c'_{aj} = \frac{x_{ij} + c_{aj}}{2} \quad (2)$$

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

```

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

```

The overlapping between subsets is only occurring for border instances. The dynamic threshold $d_{index_b} - d_{index_a} \leq d_{index_c} - d_{index_b}$ 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_{index_a} and subset s_{index_b} , and the centroid c_{index_a} is updated, while the centroid c_{index_b} is not updated, so as to reduce the number of overlapped instances. Intuitively, if we allow updating the centroid c_{index_b} , it will move toward the subset s_{index_a} . This moving will increase the instances that can be overlapped between subsets s_{index_a} and s_{index_b} in the next iterations. The instance violating the dynamic threshold is only added to the nearest subset s_{index_a} after updating the corresponding centroid c_{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.

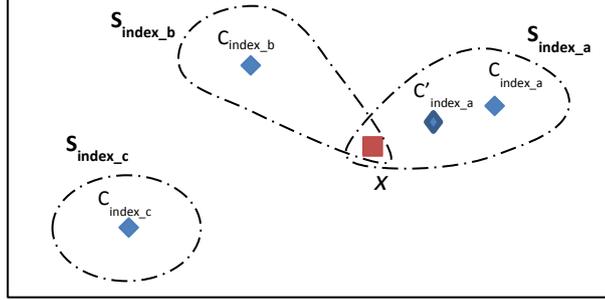


Fig. 2. Example of satisfying the dynamic threshold

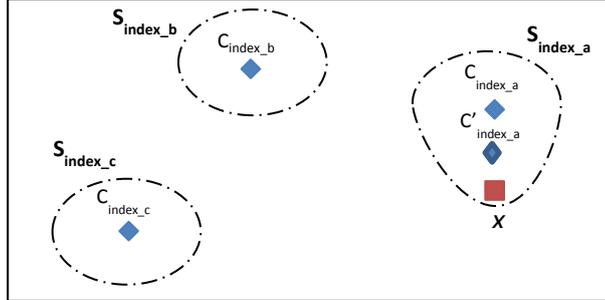


Fig. 3. Example violating the dynamic threshold

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 1. Description of dataset.

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

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.

$$Red. = 1 - |RS|/|TR| \quad (3)$$

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.

$$Acc. = |true_classified(TS)|/|TS| \quad (4)$$

The effectiveness is calculated as the product of the reduction rate and classification accuracy as given in Equation 5.

$$Eff. = Red. \times Acc. \quad (5)$$

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.

Dataset name	m	SP		CDP		OCDP	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
Ring	5	0.6755	0.0059	0.6753	0.0072	0.8126	0.0053
	10	0.6473	0.0074	0.6606	0.0043	0.8211	0.0060
	15	0.6352	0.0067	0.6507	0.0079	0.8263	0.0046
	20	0.6220	0.0040	0.6524	0.0053	0.8289	0.0072
Texture	5	0.8428	0.0038	0.8869	0.0051	0.9489	0.0022
	10	0.7915	0.0028	0.8854	0.0066	0.9500	0.0023
	15	0.7579	0.0041	0.8861	0.0067	0.9526	0.0028
	20	0.7281	0.0036	0.8816	0.0059	0.9553	0.0038
Opt-Digits	5	0.8530	0.0023	0.8757	0.0047	0.9480	0.0016
	10	0.8076	0.0038	0.8655	0.0041	0.9491	0.0025
	15	0.7767	0.0034	0.8666	0.0035	0.9514	0.0020
	20	0.7488	0.0044	0.8684	0.0038	0.9532	0.0024
Pen-Based	5	0.9110	0.0016	0.9324	0.0029	0.9711	0.0013
	10	0.8777	0.0020	0.9314	0.0027	0.9729	0.0012
	15	0.8532	0.0025	0.9331	0.0017	0.9740	0.0010
	20	0.8335	0.0022	0.9335	0.0028	0.9753	0.0012
Thyroid	5	0.8168	0.0023	0.9352	0.0029	0.9761	0.0014
	5	0.7835	0.0053	0.8085	0.0074	0.8703	0.0084
	10	0.7720	0.0058	0.8240	0.0080	0.8829	0.0076
	15	0.7666	0.0048	0.8469	0.0056	0.8971	0.0055
	20	0.7645	0.0053	0.8654	0.0070	0.8987	0.0024

	25	0.7623	0.0038	0.8755	0.0097	0.9090	0.0080
Phoneme	5	0.6765	0.0048	0.7463	0.0179	0.8250	0.0038
	10	0.6437	0.0043	0.8029	0.0175	0.8529	0.0057
	15	0.6316	0.0034	0.8346	0.0152	0.8555	0.0049
	20	0.6204	0.0069	0.8680	0.0140	0.8698	0.0073
	25	0.6096	0.0064	0.8792	0.0141	0.8719	0.0058
Sat-Image	5	0.7454	0.0047	0.8044	0.0092	0.8615	0.0014
	10	0.7215	0.0045	0.8227	0.0079	0.8749	0.0044
	15	0.7050	0.0036	0.8438	0.0104	0.8832	0.0032
	20	0.6938	0.0042	0.8549	0.0072	0.8905	0.0042
	25	0.6831	0.0036	0.8660	0.0071	0.8944	0.0035
Magic	5	0.6074	0.0039	0.6140	0.0053	0.7335	0.0032
	10	0.5973	0.0029	0.6284	0.0084	0.7527	0.0034
	15	0.5875	0.0031	0.6494	0.0116	0.7614	0.0022
	20	0.5825	0.0035	0.6781	0.0102	0.7723	0.0054
	25	0.5761	0.0036	0.7090	0.0084	0.7792	0.0031

TABLE 3. Classification accuracy 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.

Dataset name	m	SP		CDP		OCDP	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
Ring	5	0.8330	0.0105	0.8304	0.0156	0.8328	0.0133
	10	0.8311	0.0046	0.8349	0.0061	0.8243	0.0094
	15	0.8322	0.0110	0.8341	0.0076	0.8234	0.0079
	20	0.8403	0.0106	0.8278	0.0086	0.8300	0.0121
	25	0.8326	0.0111	0.8292	0.0093	0.8232	0.0102
Texture	5	0.9822	0.0049	0.9705	0.0086	0.9480	0.0074
	10	0.9800	0.0052	0.9680	0.0054	0.9482	0.0092
	15	0.9856	0.0037	0.9660	0.0070	0.9475	0.0059
	20	0.9862	0.0055	0.9644	0.0075	0.9482	0.0079
	25	0.9876	0.0021	0.9669	0.0056	0.9465	0.0140
Opt-Digits	5	0.9781	0.0051	0.9767	0.0086	0.9464	0.0062
	10	0.9788	0.0040	0.9744	0.0066	0.9486	0.0061
	15	0.9827	0.0059	0.9733	0.0064	0.9464	0.0095
	20	0.9820	0.0075	0.9753	0.0070	0.9502	0.0110
	25	0.9847	0.0046	0.9733	0.0066	0.9379	0.0123
Pen-Based	5	0.9886	0.0031	0.9793	0.0065	0.9661	0.0073
	10	0.9886	0.0029	0.9796	0.0054	0.9623	0.0055
	15	0.9884	0.0037	0.9773	0.0048	0.9632	0.0065
	20	0.9885	0.0029	0.9753	0.0068	0.9605	0.0068
	25	0.9899	0.0023	0.9759	0.0031	0.9593	0.0062
Thyroid	5	0.8701	0.0089	0.8582	0.0189	0.8521	0.0232
	10	0.8542	0.0121	0.8415	0.0215	0.8329	0.0249
	15	0.8575	0.0089	0.8169	0.0164	0.7985	0.0190
	20	0.8519	0.0106	0.8024	0.0270	0.7940	0.0126
	25	0.8440	0.0151	0.8018	0.0238	0.7960	0.0222
Phoneme	5	0.8692	0.0166	0.8483	0.0181	0.8396	0.0121
	10	0.8705	0.0090	0.8161	0.0126	0.8198	0.0151
	15	0.8671	0.0087	0.8035	0.0187	0.8122	0.0140
	20	0.8638	0.0110	0.7874	0.0183	0.8050	0.0160
	25	0.8666	0.0154	0.7755	0.0280	0.8038	0.0156
Sat-Image	5	0.8838	0.0093	0.8741	0.0094	0.8605	0.0096
	10	0.8842	0.0064	0.8634	0.0089	0.8547	0.0113
	15	0.8841	0.0064	0.8570	0.0137	0.8429	0.0120
	20	0.8838	0.0090	0.8521	0.0104	0.8410	0.0103
	25	0.8824	0.0121	0.8472	0.0089	0.8421	0.0089
Magic	5	0.7695	0.0108	0.7668	0.0117	0.7403	0.0148
	10	0.7673	0.0088	0.7614	0.0105	0.7305	0.0119
	15	0.7655	0.0115	0.7556	0.0110	0.7256	0.0123
	20	0.7675	0.0128	0.7482	0.0125	0.7208	0.0119
	25	0.7675	0.0143	0.7440	0.0114	0.7217	0.0131

TABLE 4. Effectiveness 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.

Dataset name	m	SP		CDP		OCDP	
		Avg.	Std.	Avg.	Std.	Avg.	Std.
Ring	5	0.5627	0.0099	0.5608	0.0139	0.6767	0.0118
	10	0.5379	0.0070	0.5515	0.0045	0.6768	0.0077
	15	0.5286	0.0096	0.5427	0.0084	0.6804	0.0096
	20	0.5226	0.0054	0.5400	0.0066	0.6879	0.0097
	25	0.5097	0.0099	0.5416	0.0073	0.6845	0.0080
Texture	5	0.8278	0.0045	0.8608	0.0079	0.8996	0.0075
	10	0.7756	0.0049	0.8570	0.0062	0.9007	0.0077
	15	0.7470	0.0040	0.8559	0.0084	0.9026	0.0061
	20	0.7181	0.0042	0.8502	0.0080	0.9058	0.0087
	25	0.6935	0.0043	0.8551	0.0084	0.9043	0.0128
Opt-Digits	5	0.8343	0.0035	0.8553	0.0097	0.8972	0.0052
	10	0.7904	0.0037	0.8433	0.0057	0.9003	0.0051
	15	0.7633	0.0046	0.8434	0.0076	0.9005	0.0088
	20	0.7353	0.0074	0.8469	0.0050	0.9057	0.0093
	25	0.7152	0.0052	0.8412	0.0074	0.8948	0.0117
Pen-Based	5	0.9006	0.0032	0.9130	0.0063	0.9381	0.0067
	10	0.8677	0.0037	0.9124	0.0060	0.9362	0.0057
	15	0.8433	0.0037	0.9120	0.0056	0.9381	0.0059
	20	0.8240	0.0034	0.9104	0.0062	0.9368	0.0072
	25	0.8086	0.0034	0.9126	0.0035	0.9364	0.0061
Thyroid	5	0.6818	0.0069	0.6939	0.0179	0.7416	0.0232
	10	0.6594	0.0085	0.6934	0.0200	0.7355	0.0266
	15	0.6574	0.0078	0.6918	0.0123	0.7163	0.0173
	20	0.6513	0.0089	0.6943	0.0217	0.7136	0.0110
	25	0.6434	0.0104	0.7018	0.0171	0.7236	0.0255
Phoneme	5	0.5879	0.0113	0.6328	0.0114	0.6927	0.0108
	10	0.5603	0.0061	0.6552	0.0176	0.6992	0.0140
	15	0.5477	0.0073	0.6706	0.0189	0.6947	0.0103
	20	0.5359	0.0086	0.6834	0.0158	0.7002	0.0162
	25	0.5283	0.0095	0.6816	0.0208	0.7009	0.0143
Sat-Image	5	0.6587	0.0073	0.7030	0.0068	0.7413	0.0080
	10	0.6380	0.0074	0.7103	0.0093	0.7477	0.0103
	15	0.6233	0.0063	0.7231	0.0131	0.7445	0.0100
	20	0.6132	0.0074	0.7284	0.0054	0.7489	0.0106
	25	0.6028	0.0096	0.7337	0.0088	0.7532	0.0070
Magic	5	0.4673	0.0052	0.4708	0.0085	0.5429	0.0098
	10	0.4583	0.0050	0.4784	0.0096	0.5499	0.0084
	15	0.4497	0.0078	0.4906	0.0101	0.5524	0.0098
	20	0.4470	0.0084	0.5073	0.0073	0.5566	0.0085
	25	0.4421	0.0079	0.5274	0.0093	0.5623	0.0095

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.

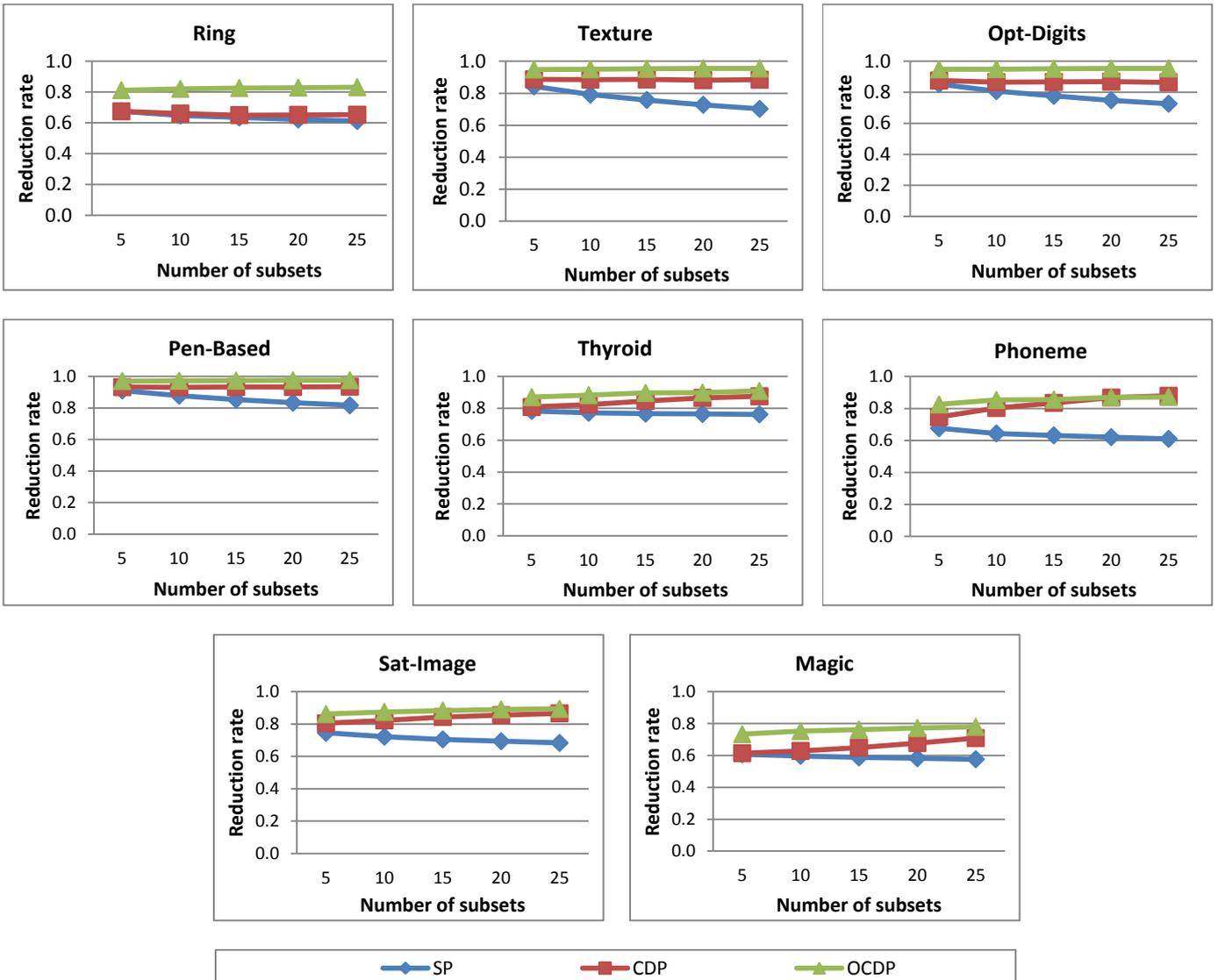


Fig. 4. The reduction rate of the eight employed datasets for the three partitioning approaches with a different number of subsets.

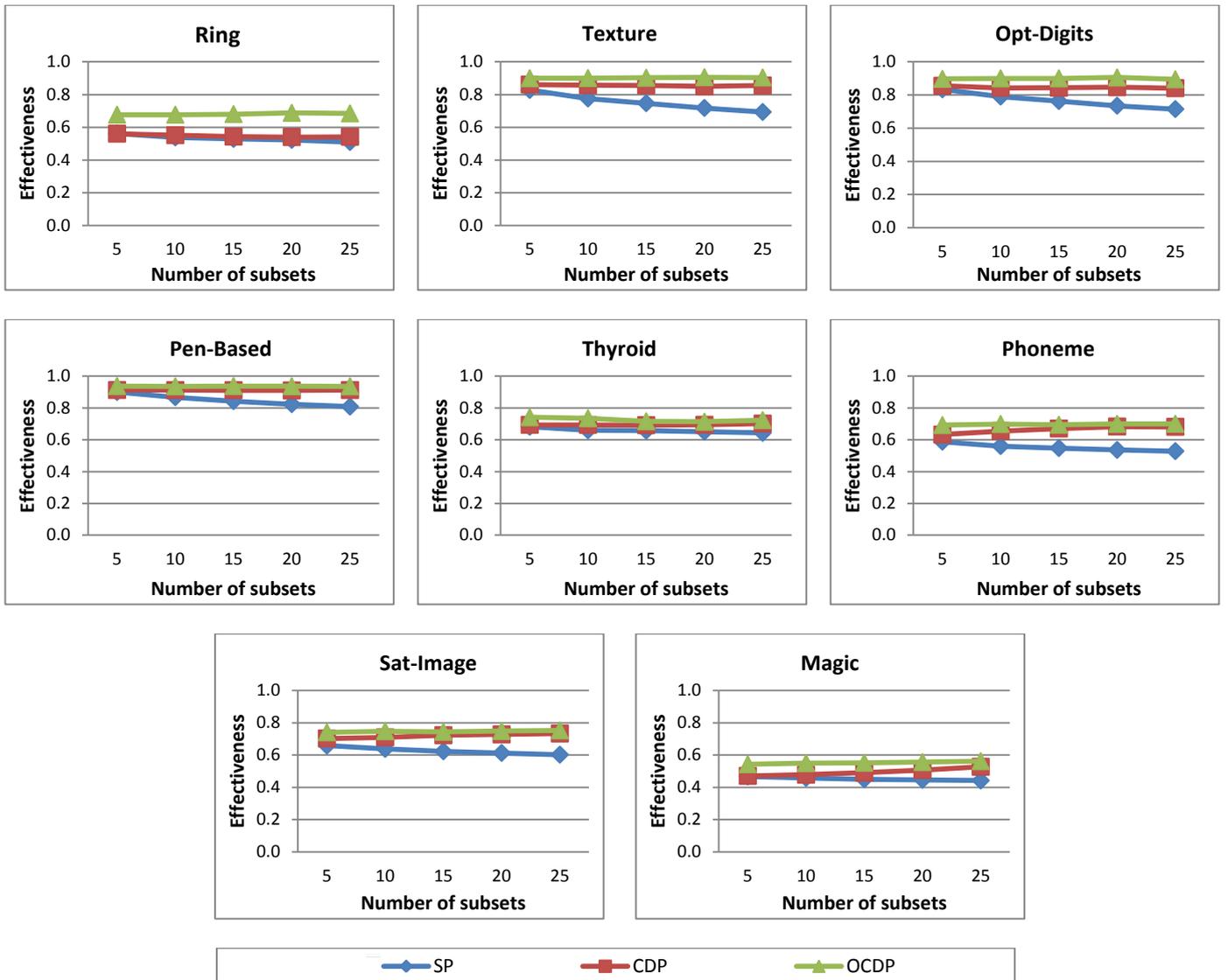


Fig. 5. The effectiveness of the eight employed datasets for the three partitioning approaches with a different number of subsets.

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

1. S. Garcia, Juli, and F. Herrera, *Data Preprocessing in Data Mining*, 1st ed.: Springer International Publishing, 2015.
2. M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," *Mobile Networks and Applications*, vol. 19, pp. 171-209, 2014.
3. P. Zikopoulos, and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, 1st ed.: McGraw-Hill Osborne Media, 2011.
4. J. Gantz and D. Reinsel, "Extracting value from chaos," IDC iView, Technical report, 2011.
5. O. R. Team, "Big data now: current perspectives from O'Reilly Radar," O'Reilly Media, Technical report, 2011.
6. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 1st ed.: Morgan Kaufmann, 2000.
7. M. H. Rehman et al., "Big Data Reduction Methods: A Survey," *Data Science and Engineering*, vol. 1, pp. 265-284, 2016.
8. H. Liu and H. Motoda, *Instance Selection and Construction for Data Mining*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.
9. E. Leyva, A. González, and R. Pérez, "Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective," *Pattern Recognition*, vol. 48, pp. 1523-1537, 2015.
10. I. Triguero et al., "MRPR: A MapReduce solution for prototype reduction in big data classification," *Neurocomputing*, vol. 150, Part A, pp. 331-345, 2015.
11. J. Derrac, S. Garcia, and F. Herrera, "Stratified prototype selection based on a steady-state memetic algorithm: a study of scalability," *Memetic Computing*, vol. 2, pp. 183-199, 2010.
12. M. Malhat, M. El Menshawy, H. Mousa, and A. E. Sisi, "Improving instance selection methods for big data classification," in *13th International Computer Engineering Conference (ICENCO)*, pp. 213-218, 2017.
13. P. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, pp. 515-516, 1968.
14. I. Triguero, S. García, F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44 (4), pp. 901-916, 2011.
15. T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78 (9), pp. 1464-1480, 1990.
16. J. Sánchez, "High training set size reduction by space partitioning and prototype abstraction," *Pattern Recognition*, vol. 37 (7), pp. 1561-1564, 2004.
17. D. Wilson, T. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*. Vol. 38, pp. 257-286, 2000.
18. F. Angiulli, "Fast nearest neighbor condensation for large data sets classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19 (11), pp. 1450-1464, 2007.
19. G. Gates, "The Reduced Nearest Neighbor Rule," *IEEE Transaction Information Theory*, vol. 18, no. 3, pp. 431-433, 1972.
20. D. Wilson, "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 3, pp. 408-421, 1972.
21. A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera J. Alcalá-Fdez. (2018, October) KEEL-dataset repository. [Online]. <https://sci2s.ugr.es/keel/datasets.php>
22. I. Tomek, "Two Modifications of CNN," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, pp. 769-772, 1976.
23. V. Susheela Devi and M. Narasimha Murty, "An incremental prototype set building technique," *Pattern Recognition*, vol. 35, pp. 505-513, 2002.
24. F. Chang, C. Lin, and C. Lu, "Adaptive Prototype Learning Algorithms: Theoretical and Experimental Studies," *Journal of Machine Learning Research*, vol. 7, pp. 2125-2148, 2006.
25. T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.
26. H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data Mining and Knowledge Discovery*, vol. 6, pp. 153-172, 2002.