Expert Systems With Applications 149 (2020) 113297

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A new approach for instance selection: Algorithms, evaluation, and comparisons

Mohamed Malhat*, Mohamed El Menshawy, Hamdy Mousa, Ashraf El Sisi

Department of Computer Science, Faculty of Computers and Information, Menofia University, Egypt

ARTICLE INFO

Article history:
Received 20 October 2018
Revised 1 September 2019
Accepted 5 February 2020
Available online 7 February 2020

Keywords:
Big data
Data mining
Instance selection
Global density function
Time complexity

ABSTRACT

Several approaches for instance selection have been put forward as a primary step to increase the efficiency and accuracy of algorithms applied to mine big data. The instance selection task scales indeed big data down by removing irrelevant, redundant, and unreliable data, which, in turn, reduces the computational resources necessary for completing the mining task. The local density-based approaches are recently acknowledged as feasible approaches in terms of reduction rate, effectiveness, and computation time metrics. However, these approaches endure low classification accuracy results compared with other approaches.

In this manuscript, we propose a new layered and operational approach to address these limitations as well as advance the state-of-the-art by balancing among classification accuracy, reduction rate, and time complexity. We commence by designing a new algorithm (called GDIS) that selects most relevant instances using a global density and relevance functions. This enable us to consider a global view overall a data set to get a better classification accuracy results than current density-based approaches. We design another novel algorithm (called EGDIS), which maintains the effectiveness results of the GDIS algorithm while improving reduction rate results. Moreover, we compare our algorithms against three state-of-the-art algorithms to validate their performance. We develop a Java toolkit called ISTK on the top of the GDIS and EGDIS algorithms, the density-based approaches, and the state-of-the-art algorithms. We also develop a suitable user interface and its management and validation capabilities to ease-of-use and visualize results and data sets. We evaluate and test the performance of our algorithms in terms of four metrics (reduction rate, classification accuracy, effectiveness, and computation time) using twenty-four standard data sets and conduct an intensive set of experiments. The experimental results proved that the GDIS algorithm outperforms the density-based approaches in terms of classification accuracy and effectiveness, the EGDIS algorithm outperforms the density-based approaches in terms of reduction rate and effectiveness, and the GDIS and EGDIS algorithms outperform the state-of-the-art algorithms in terms of achieving a good results in both the effectiveness and computation time metrics. We finally test the scalability and compute experimentally the polynomial-time complexity of our algorithms.

1. Introduction

Data preprocessing is a fundamental step in the process of data mining, because the data acquisition process is loosely governed, leading to impossible data records (e.g., Gender: Male; Pregnant: Yes), missing values, and so forth (García, Luengo, & Herrera, 2015). Mining raw data without carefully addressing such issues can yield misleading results. Therefore, priori to running any mining or analysis, the quality of raw data must make first (Bolt, Leoni, & Aalst, 2016; Chen, Mao, & Liu, 2014). However, when there is a very great degree of irrelevant redundant information and noisy and unreliable data, the process of knowledge discovery is very difficult to carry on Zhang, Zhang, and Yang (2003). The data preprocessing methods can be classified into two main classes: data preparation and data reduction (García et al., 2015). The data preparation class incorporates methods of transforming (Hashem et al., 2015), integrating (Do & Rahm, 2007), normalizing (Hashem et al., 2015), and cleaning data (Kim, Choi, Hong, Kim, & Lee, 2003). The data reduction class includes methods of feature selection (Liu & Motoda, 1998), instance selection (Liu & Motoda, 2001), and discretization (Liu, Hussain, Tan, & Dash, 2002). Having performed the chain of data preprocessing methods, we expect a correct and useful data that can be further mined and analyzed by appropriate

* Corresponding author.
E-mail addresses: m.gmalhat@yahoo.com (M. Malhat), mohamed.elmenshawy@ci.menofia.edu.eg (M.E. Menshawy), hamdimmm@hotmail.com (H. Mousa), ashrafelsisi@hotmail.com (A.E. Sisi).

https://doi.org/10.1016/j.eswa.2020.113297
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

algorithms (Silva, Souza, & Motta, 2016). Unfortunately, current data mining algorithms cannot be applied directly to raw (or even preprocessed) data as the most of application domains and social media platforms (e.g., Twitter, Stack Overflow, and Facebook) produce a big and mismanaged data (Chen et al., 2014; Turner & Lambert, 2015). The National Institute of Standards and Technology (NIST) defines a big data as "the data of which the data volume, acquisition speed, or data representation limits the capacity of using traditional methods to conduct effective analysis" (Team, 2011). According to the 6V model (ur Rehman et al., 2016), a big data has six characteristics: (1) the volume characteristic defining the size of data is rapidly increase, (2) the velocity characteristic defining the generation of data is collected with respect to timeliness, (3) the variety characteristic defining a data contains various types, such as unstructured, semi-structured, and structured data, (4) the veracity characteristic defining a data contains irrelevant, missing, outlier, and redundant data, (5) the variability characteristic defining a data is not generated with the same speed and quality, and (6) the value characteristics defining the ability to extract a valuable information from data. Therefore, a big data becomes the driver of developing or improving the data preprocessing methods (Gantz & Reinsel, 2011).

The focus of the manuscript is on the data reduction class. On the one hand, the data reduction can be used to reduce the size of data to a manageable one, which leads to decreasing computational resources (time and space) needed for classifiers, especially within instance-based methods (or lazy learning methods) (Wilson & Martinez, 2000). On the other hand, the data reduction can be used to remove irrelevant and unreliable data. Thus, it increases the efficiency and accuracy of mining algorithms implemented, for example, in predictive tasks (Liu & Motoda, 2001). The data reduction is generally needed to understand data and used in various tasks including regression, classification, and supervised and unsupervised learning (Reinartz, 2002). Instance selection (IS), the most widely data reduction methods in the literature, is the process of selecting a subset of the original data set, the most representative examples, to achieve the original target of the mining task, when the whole data set is used (Liu & Motoda, 2001). Unlike data sampling, IS is an intelligent process, depending on the degree of noise or irrelevance in a data set and the mining task (García et al., 2015). Technically, IS identifies and removes the irrelevant instances from the training set and faces the trade-off between the reduction rate and classification accuracy metrics. Several methods proposed recently to solve the trading-off problem between these two metrics with a reliable, suitable, and accurate results, over high volume of data (Arnaiz-González, González-Rogel, Díez-Pastor, & López-Nozal, 2017; Carbonera & Abel, 2015; 2016; García, Cano, & Herrera, 2008; García-Osorio, de Haro-García, & García-Pedrajas, 2010; de Haro-García & García-Pedrajas, 2009; Liu, Wang, Wang, Lv, & Konan, 2017; Silva et al., 2016). The density-based approach (as we will show in Section 2) is one of the recent and promising approaches proposed to solve this trading-off problem. The local density-based instance selection algorithm (shortly, LDIS) (Carbonera & Abel, 2015) and the centroid density-based instance selection algorithm (shortly, CDIS) (Carbonera & Abel, 2016) are the two most popular algorithms in the literature of density-based approaches. These algorithms indeed process instances of each class separately by computing a local density function and finding local nearest neighbors for each instance. They then proceed to preserve only densest instances from each class. These algorithms are compared with five popular and standard IS methods: Edited Nearest-Neighbor (ENN) (Wilson, 1972); Decremental Reduction Optimization Procedure (DROP3) (Wilson & Martinez, 2000); Iterative Case Filtering (ICF) (Brighton & Mellish, 2002); Local Set-based Smoother (LSSm) (Leyva, González, & Pérez, 2015); and Local Set Border Selector (LSBo) (Leyva et al., 2015). The reduction rate, classification accuracy, and effectiveness metrics are used to evaluate the performance of the density-based algorithms (LDIS and CDIS)and the five selected IS methods. The effectiveness is used to determine the ability of an IS method to balance between reduction rate and classification accuracy. The results showed that density-based algorithms (LDIS and CDIS) are able to obtain a good reduction rate as well as a high effectiveness. However, current density-based algorithms have major drawbacks: (1) considering each class separately leads to removing relevant instances, which, in turn, decrease classification accuracy, (2) a high effectiveness is in favor of reduction rate, which is not preferable for most application domains, and (3) computation time is high for data sets which contain a low number of classes.

1.1. Motivations and contributions

Our *first* motivation is to propose a new operational approach to tackle the density-based approaches' drawbacks. Our main idea is to apply a density function to the whole training set, which, in turn, transforms the scope of the density function from local to global. A density function is used only for instances that have a class label like their k nearest neighbors to determine the densest instances. To apply our approach, we develop two global density-based instance selection algorithms called global density-based instance selection (shortly, GDIS) and enhanced global density-based instance selection (shortly, EGDIS). The GDIS algorithm applies a new function called relevance function besides a global density function. The relevance function is technically to determine the importance of instances that have at least one instance in their k nearest neighbors with a class label differs from the class label of the observed instance. The GDIS algorithm achieves a classification accuracy results that is better than the density-based approaches but with a decrease in a reduction rate. In order to address the decreasing in a reduction rate, we develop the EGDIS algorithm. The EGDIS algorithm uses another function called irrelevance function besides a global density function. The irrelevance function used to determine the number of instances in the k nearest neighbors of an instance i that may misclassify instance i. The EGDIS algorithm preserves only instances that may be misclassified using current k nearest neighbors of instances, while densest instances are preserved using a global density function. It then improves the results of the GDIS algorithm by enhancing the reduction rate and effectiveness.

Our *second* motivation is to address the unsupervised classification problem of these twenty-four data sets and compare the performance of our algorithms and the density-based approaches. Moreover, we compare the performance of our algorithms and three state-of-the-art IS algorithms: (1) Condensed Nearest Neighbor (CNN) (Hart, 1968); (2) ENN (Wilson, 1972); and (3) ICF (Brighton & Mellish, 2002). The CNN algorithm is a standard IS algorithm that achieves a suitable results for most data sets, while the ENN algorithm is a standard IS algorithm that achieves a very good classification accuracy result for most data sets. The ICF algorithm is one of the recent algorithms that achieves a good results in both reduction rate and classification accuracy metrics, but it requires very high computation time. The results showed that: (1) the GDIS algorithm has a better classification accuracy and effectiveness results than the density-based approaches, (2) the EGDIS algorithm has a better reduction rate and effectiveness results than the density-based approaches, and (3) the GDIS and EGDIS algorithms are able to achieve better effectiveness results than the state-of-the-art algorithms in reasonable time. Our *final* motivation is to test the scalability and compute experimentally the time complexity of our algorithms, which are entirely missing in the literature.

The work will continue as follows. In Section 2, we review and discuss related work to identify their limitations. In Section 3, we introduce our approach and its algorithms from an algorithmic point of view. In Section 4, we briefly introduce the implementation of our toolkit. We also report and evaluate the experimental results of reducing twenty-four standard data sets. In Section 5, we conduct a set of experiments to evaluate our approach's scalability and compute its time complexity. In Section 6, we give an overall discussion to summarize the strengths and weakness of our approach. We conclude the paper and identify the directions of future work in the same section.

2. Related Work

In this section, we begin with presenting the García et al. (2015)'s taxonomy of the IS methods and review and compare the IS methods that adopt the density function (shortly, ISDF). We use the García et al.'s taxonomy to discriminate these methods and assess their performance.

García et al. categorized the IS methods in the literature based on three criteria: (1) the type of selection, (2) direction of search, and (3) evaluation of search.

1. The type of search criterion focuses on whether the IS method search to retain border points as in the condensation family, central points as in the edition family, or border and central points as in the hybrid family.
2. The direction of search criterion focuses on how the search process for a subset from a training set can be proceed either in a incremental, decremental, batch, mixed, or fixed way.
3. The kNN rule (Cover & Hart, 1967) is used to evaluate the search of an IS method with respect to either applying the filter or wapper quality criterion to a partial data or a complete training set, respectively.

The García et al.'s taxonomy of the IS methods is illustrated in Fig. 1. The condensation methods have ability to increase reduction rate with negative effect on classification accuracy. The Edition methods are able to increase classification accuracy with low reduction rate. Finally, the hybrid methods are achieving moderate reduction rate and moderate classification accuracy.

2.1. Reviewing and comparing ISDF methods

The basic idea of the ISDF algorithms is to select the instances that have a high concentration of instances near to them (densest instances). They search for the densest instances in each class of the data set separately. This local search strategy requires lower time complexity than the time complexity of methods that adopt a global search strategy (Carbonera & Abel, 2015; 2016). The LDIS is one of the ISDF algorithms that has a good results regrading the reduction rate and effectiveness (Carbonera & Abel, 2015). LDIS starts with splitting training set into a set of sets (i.e., one set for each class). For each class set, a local density function $Dens(i, P)$ is calculated using Eq. (1), adapted from Bai, Liang, Dang, and Cao (2012).

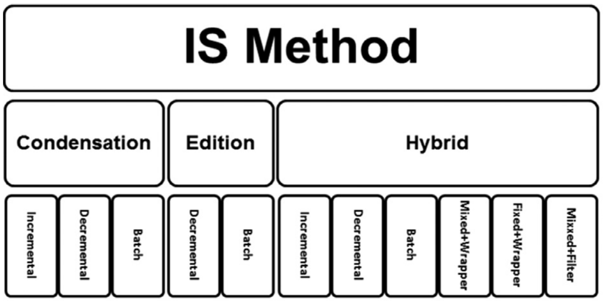


Fig. 1. The taxonomy of the IS methods.

$$Dens(i, P) = -\frac{1}{|P|} \sum_{j \in P} d(i, j) \tag{1}$$

Here, $P = \{i_0, i_1, \ldots, i_q\}$ is a set of q instances, i is a given instance (i.e., $i \in P$), and d is a given distance function between i and j (where $j \in P$). We can read Eq. (1) as follows: $Dens(i, P)$ provides the density of the instance i related to the set P of instances. When P is a subset of the whole data set which represents a class set, $Dens(i, P)$ then represents the local density of i. For each instance $i \in P$, then the local k nearest neighbors are determined (k is a user given). Finally, the instances whose densities are larger than or equal to the densities of its k nearest neighbors are preserved. Therefore, the LDIS algorithm preserves only the densest instances within each class.

Carbonera and Abel (2016) adopt a new density function (see Eq. (2)) and introduce a new algorithm called CDIS algorithm, which principally follows the same strategy of their LDIS algorithm (Carbonera & Abel, 2015). The main idea of the new density function is to give a higher density value to instances that have k nearest neighbors disposed around it.

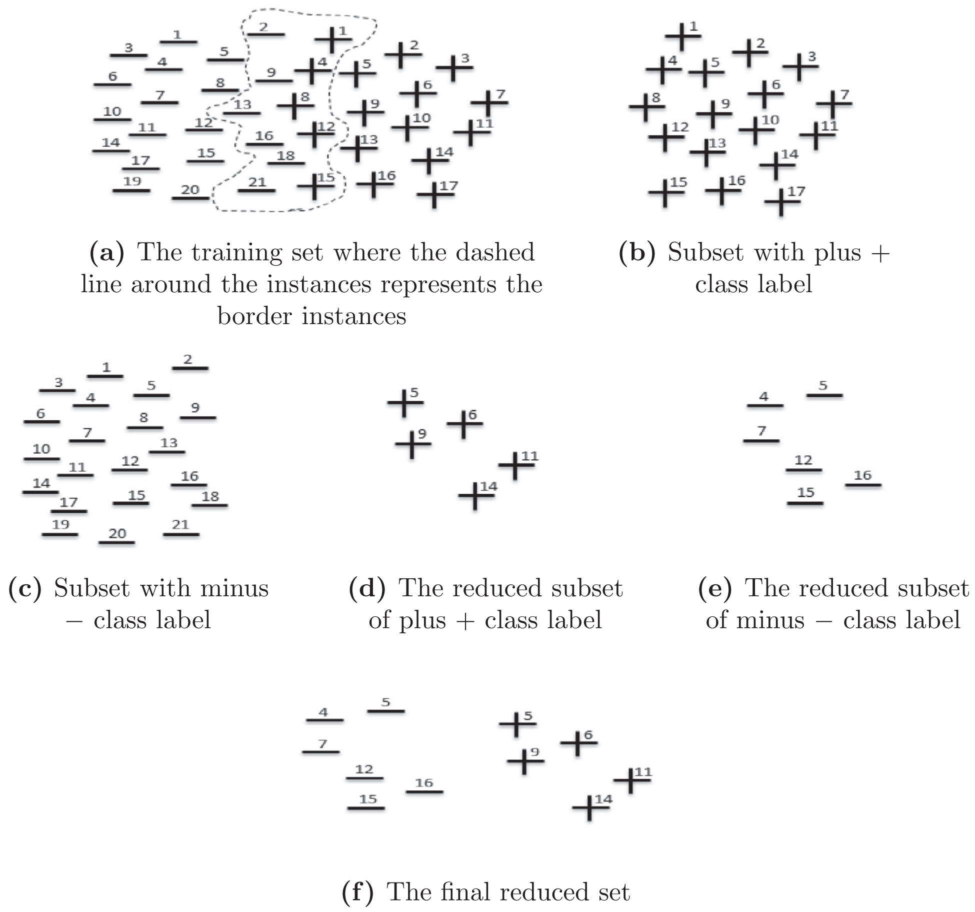
$$Dens(i, P) = \frac{\sum_{j \in P} \frac{1}{1+d(i,j)}}{1 + d(i, centroid(pkn(i, k)))} \tag{2}$$

Here, $\sum_{j \in P} \frac{1}{1+d(i,j)}$ is the summation of the multiplicative inverse of the distance between instances i and j. Note that one is added to denominator to avoid a division by zero. pkn is a function that takes instance i and a user given number k and returns the set of k nearest neighbors of i. $centroid$ is a function that takes the k nearest neighbors of instance i and returns their centroid. $d(i, centroid(pkn(i, k)))$ is the distance between an instance i and the centroid of its k nearest neighbors.

Hereafter, we assess the LDIS and CDIS algorithms using the García et al.'s taxonomy. The two algorithms are of the category of the Edition because they preserve the densest instances. The densest instances are instances surrounded with the large possible number of instances to give them a higher density value. Therefore, the densest instances are internal instances (internal points). The LDIS and CDIS algorithms start with empty reduced set, so they are incremental. Since the two algorithms use a kNN classifier to assess the accuracy of its performance on a complete training set, then they are wrapper based on the principle of Olvera-López, Carrasco-Ochoa, Martínez-Trinidad, and Kittler (2010). The LDIS and CDIS algorithms were compared with five well-performed and standard IS methods over 20 well-known data sets from UCI repository.[1] These standard methods are ENN (Wilson, 1972), DROP3 (Wilson & Martinez, 2000), ICF (Brighton & Mellish, 2002), LSSm (Leyva et al., 2015), and LSBo (Leyva et al., 2015). Table 1 shows the comparison results with respect to reduction rate, classification accuracy, and effectiveness metrics (Carbonera & Abel, 2016).

From the table, the results show that: (1) the LDIS and CDIS achieve the highest average reduction rate, (2) the LDIS and CDIS have the highest average effectiveness, and (3) the LSSm has the highest average classification accuracy. It is worth noting that the high effectiveness of the LDIS and CDIS algorithms is in favor of the reduction rate, rather than the classification accuracy. Based on García et al.'s taxonomy, the edition methods are able to increase classification accuracy with low reduction rate. However, the LDIS and CDIS (or ISDF algorithms in general) assessed as edition methods do not achieve the expected results, because they preserve a

[1] https://archive.ics.uci.edu/ml/datasets.html.



(a) The training set where the dashed line around the instances represents the border instances

(b) Subset with plus + class label

(c) Subset with minus − class label

(d) The reduced subset of plus + class label

(e) The reduced subset of minus − class label

(f) The final reduced set

Fig. 2. An illustrative example of applying ISDF methods. The instances with + class label are numbered from 1 to 17 and instances with - class label are numbered from 1 to 21 in the training set to illustrate which instances are maintained in the reduced set.

Table 1
The comparison results of LDIS and CDIS algorithms and five standard IS methods using 20 data sets.

IS Method	Reduction Rate	Classification Accuracy	Effectiveness
ENN	0.18	0.81	0.12
Drop3	0.72	0.79	0.57
ICF	0.76	0.77	0.59
LSSm	0.08	0.82	0.06
LSBo	0.78	0.73	0.59
LDIS	0.84	0.76	0.64
CDIS	0.90	0.75	0.68

few number of the internal instances (i.e., densest instances), not all internal instances. Meanwhile, it is expected that the ISDF algorithms increase the reduction rate better than edition methods, however, it is not the case. An example of the ISDF algorithms will be introduced in Section 2.2 to illustrate these limitations in more details.

2.2. Current challenging issues

The ISDF methods endure low classification accuracy results, due to a local search for the densest instances. The local search indeed leads to missing an important instances (i.e., border instances) to be added into the reduced set. To illustrate this limitation and the above limitations, we consider the following example.

Example 1. Fig. 2 is an illustrative example for applying the ISDF methods to a certain training set. This training set contains a number of instances with two class labels (plus (+) and minus (-)), as shown in Fig. 2a. The ISDF methods start with splitting the training set into subsets (one subset for each class label). By applying this step, we have plus subset (see Fig. 2b) and minus subset (see Fig. 2c). For each class subset, the densest instances are then computed using the density function defined in Eq. (1) for the LDIS method or Eq. (2) for the CDIS method. The results of applying this step on each class subset are given in Fig. 2d and Fig. 2e, respectively. Finally, the reduced sets are combined togethers to produce the final reduced set, as shown in Fig. 2f.

By observing the final reduced set, we found that the ISDF methods focus on preserving the internal instances, which have higher density values than other instances and discarding the border instances. The border instances in fact represent a discrimination between different classes in the data set (see Fig. 2a for border instances surrounded with dashed line). The final reduced set is used to assess the classification accuracy of the ISDF methods by classifying the instances in the testing set. The instances in testing set that are close to the instances in the dashed line have a high probability to be misclassified, since the local search for densest instances in each class ignores the boundaries between classes. Therefore, the classification accuracy of the ISDF methods is negatively affected by the local search for the densest instances. Furthermore, the ISDF methods didn't consider all internal instances and preserve merely the densest instances from them, as shown in Fig. 2d and e. Therefore, the

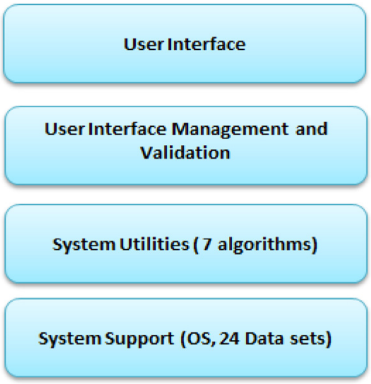

User Interface
User Interface Management and Validation
System Utilities (7 algorithms)
System Support (OS, 24 Data sets)

Fig. 3. A layered architecture pattern of the proposed approach.

reduction rate of the ISDF methods is better than other edition methods.

3. Proposed approach

Fig. 3 demonstrates the architecture of the proposed approach, which is deliberately arranged into separate four layers. Each layer solely depends on the services and facilities provided by the layer immediately underneath it. The separation and independence offered by the layered pattern allows changes to be handled locally. For example, changing an existing data view can be performed without any modifications to the underlying data set. This layered approach provides the incremental development as when a layer is developed, some of its services can be made ready to users. Moreover, our layered approach is changeable and portable, i.e., a new layer with a new functionality can substitute an existing one without modifying other elements of the system, provided that its interface is not changed. Even if a layer interface is changed, the adjacent layer is merely affected. From the figure, the lowest layer incorporates system support (operating system and twenty-four data sets, the details are given in Section 4.1). The next layer includes the system utilities (seven IS algorithms, see Section 3.1 for more details) used to select relevant instances. The third layer is mainly concerned with managing user interface and providing user input validation with the top layer offering user interface capabilities (the details are given in supplementary material). Any layer of these layers can be divided into two or more layers, if it is needed.

3.1. Developed algorithms

In this section, the developed GDIS and EGDIS algorithms are explained. For the readability purpose of our algorithms, we introduce the following notations.

1. $TS = \{i_1, i_2, \ldots, i_n\}$ is the original training data set that contains a set of n instances.
2. $RS = \{i_1, i_2, \ldots, i_m\}$ is the reduced training set, which contains a set of m instances such that $RS \subseteq TS$. RS produces as a result of applying an IS method on TS.
3. l: $TS \rightarrow L$, where L is a set of class labels, is a function that returns the corresponding class label $l_i \in L$ for a given instance $i \in TS$.
4. $knn : TS \times \mathrm{N}^+ \rightarrow 2^{TS}$ is a function that returns a set C, which contains the k nearest neighbors of a given instance $i \in TS$, excepting i itself, and a given $k \in \mathrm{N}^+$ (N^+ is a positive natural numbers). Notice that 2^{TS} is the powerset of TS, i.e., the set of all subsets of TS including the empty set and TS itself.
5. *relevance*: $TS \times C \rightarrow r$ is a function that returns the number of instances in a given set C that have a class label likes to a given instance $i \in TS$.

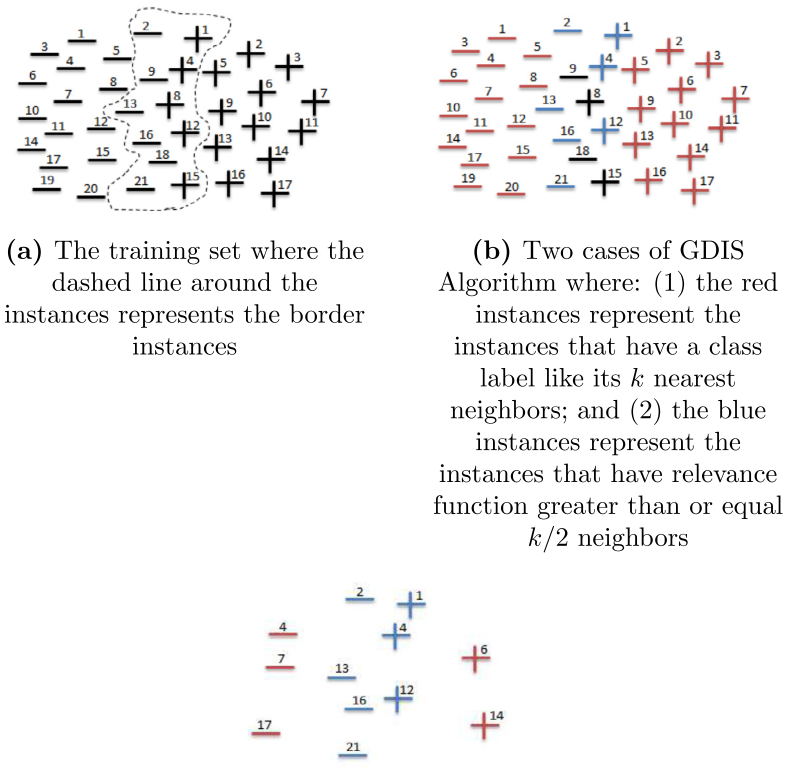
Algorithm 1: The global density-based instance selection algorithm (GDIS).

```
   Input : The training set $TS$ and the number $k$ of
           neighborhoods
   Output: The reduced set $RS$
 1 begin
 2     $RS \leftarrow \phi$
 3     foreach $i \in TS$ do
 4         $neighbors \leftarrow knn(i, k)$
 5         $x \leftarrow Relevance(i, neighbors)$
 6         if $x == k$ then
 7             $Densest \leftarrow false$
 8             foreach $neighbor \in neighbors$ do
 9                 if $Dens(i, TS) < Dens(neighbor, TS)$ then
10                     $Densest \leftarrow true$
11                 end
12             end
13             if $Densest == false$ then
14                 $RS \leftarrow RS \cup \{i\}$
15             end
16         else
17             if $x >= (k/2)$ then
18                 $RS \leftarrow RS \cup \{i\}$
19             end
20         end
21     end
22     return $RS$
23 end
```

6. *irrelevance*: $TS \times C \rightarrow ir$ is a function that returns the number of instances in a given set C that have a class label differs from a given instance $i \in TS$.

The GDIS algorithm (see Algorithm 1) takes the training set TS and the number k of nearest neighbors as input and outputs the reduced set RS. Particularly, the algorithm begins with initializing RS with the empty set (line 2). For each instance i in TS, the algorithm initializes two variables: (1) the `neighbors` variable with the k nearest neighbors of the instance i computed by the knn function (line 4); and (2) the `x` variable with the number of `neighbors` that only has a class label matches the label of the instance i using the relevance function (line 5). The if construct in lines 6-15 starts with checking if the x value equals k, the algorithm proceeds to initialize a variable (`densest` with *false*) and enters the loop over the neighbors of the instance i to determine the densest instance. Such a check intuitively ensures that all k nearest neighbors should have a class label similar to the label of the instance i. Inside the loop, the algorithm compares the density value of the instance i with the density value of each neighbor of i (line 9) and then sets *true* to `densest` whenever the density of any neighbor is greater than the density of i. By the end of the loop, if the `densest` value is false, then the instance i is added to RS, which means the instance i has a density value larger than its neighbors (line 14). In the else construct (where $x \neq k$), the algorithm compares the value of x with the half number of neighbors k (line 17). If the comparison is true, the instance i is added to RS (line 18). This intuitively means that the instance i is needed to classify more than half of its neighbors. Finally, the algorithm returns RS that contains the most relevant instances in TS (line 22).

To have an optimum algorithm in terms of minimum possible calculations, we in line 5 calculate x to only apply the density function to the instances that have $x = k$ and meanwhile modify the range of the density function in Eq. (1) with the one in Eq. (3) to



(a) The training set where the dashed line around the instances represents the border instances

(b) Two cases of GDIS Algorithm where: (1) the red instances represent the instances that have a class label like its k nearest neighbors; and (2) the blue instances represent the instances that have relevance function greater than or equal $k/2$ neighbors

(c) Reduced set

Fig. 4. An illustrative example of using the GDIS algorithm. The instances with + class label are numbered from 1 to 17 and instances with - class label are numbered from 1 to 21 in the training set to illustrate which instances are maintained in the reduced set.

get the global density function.

$$Dens(i, TS) = -\frac{1}{|TS|}\sum_{j\in TS} d(i, j) \tag{3}$$

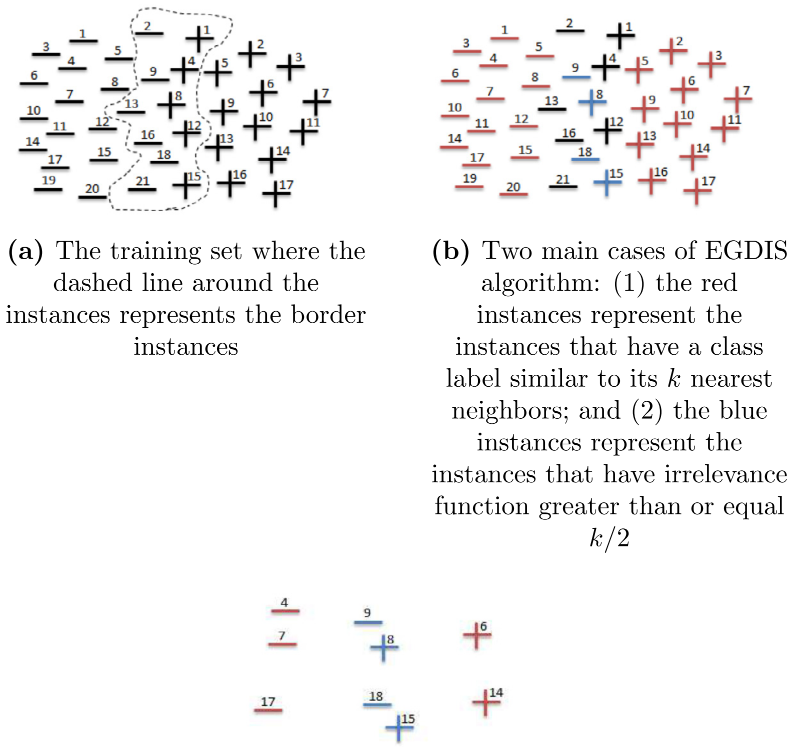
Our modification is to indeed compute the density of the instance i relatively to the whole training set TS of instances, instead of the set of instances of each class separately. This modification overcomes the low classification accuracy limitation of the ISDF algorithms (e.g., LDIS (Carbonera & Abel, 2015) and CDIS (Carbonera & Abel, 2016)) and maintains and improves the effectiveness metric, as we will show in Section 4.4.

Example 2. We use the training set shown in Fig. 4 as an illustrative example to test the correctness of the main cases in Algorithm 1 where k is equal 3. This set contains a number of instances with two class labels (+ and - signs), as shown in Fig. 4a. There are two main cases in Algorithm 1: (1) instances that have a class label similar to their k nearest neighbors represented as red instances in Fig. 4b (lines 6-15), and (2) instances that have relevance function greater than or equal $k/2$ neighbors represented as blue instances in Fig. 4b (lines 17-19). For the first case, the global density function in Eq. (3) is applied to the instances and the densest instances are added to the reduced set RS, as shown in Fig. 4c. For the second case, the instances are added to the reduced set RS, as shown in Fig. 4c.

The main power of the GDIS algorithm is to maintain instances in the reduced set RS that represent the border between different classes in the training set. These instances have a positive impact on the classification accuracy results. However, the algorithm decreases the reduction rate to maintain such instances. For this reason, we develop another algorithm called EGDIS algorithm (see Algorithm 2) to improve the reduction rate of the GDIS algorithm.

Algorithm 2: The enhanced global density-based instance selection algorithm (EGDIS).

```
Input  : The training set $TS$ and the number $k$ of
         neighborhoods
Output: The reduced set $RS$
begin
    $RS \leftarrow \phi$
    foreach $i \in TS$ do
        $neighbors \leftarrow knn(i, k)$
        $x \leftarrow Irrelevance(i, neighbors)$
        if $x == 0$ then
            $Densest \leftarrow false$
            foreach $neighbor \in neighbors$ do
                if $Dens(i, TS) < Dens(neighbor, TS)$ then
                    $Densest \leftarrow true$
                end
            end
            if $Densest == false$ then
                $RS \leftarrow RS \cup \{i\}$
            end
        else
            if $x >= (k/2)$ then
                $RS \leftarrow RS \cup \{i\}$
            end
        end
    end
    return $RS$
end
```



(a) The training set where the dashed line around the instances represents the border instances

(b) Two main cases of EGDIS algorithm: (1) the red instances represent the instances that have a class label similar to its k nearest neighbors; and (2) the blue instances represent the instances that have irrelevance function greater than or equal $k/2$

(c) The reduced set

Fig. 5. An illustrative example of using EGDIS algorithm. The instances with + class label are numbered from 1 to 17 and instances with - class label are numbered from 1 to 21 in the training set to illustrate which instances are maintained in the reduced set.

The EGDIS algorithm is similar to the GDIS algorithm from line 1 to line 15, except lines 5 and 6. In line 5, it initializes the variable x with the number of `neighbors` that have a class label differs from the instance i and results from applying the irrelevance function, given the instance i and `neighbors`. In line 6, the algorithm checks if $x = 0$ or not, meaning that no instance in the `neighbors` of the instance i has a class label differs from the instance i. In lines 16-20, it proceeds by comparing the irrelevance function of the instance i with the number of neighbors k divided by two. The instance i is added to RS only if the irrelevance function is greater than k divided by two. Finally, the algorithm returns RS that contains the most relevant instances in TS (line 22). By this modification, only instances that might be misclassified are added to RS. This leads to an increase in the reduction rate and little decrease in the classification accuracy.

Example 3. We use the training set in Fig. 5 as an illustrative example to clarify the main cases in Algorithm 2 where k is equal 3. Fig. 5a shows the training set that contains a number of instances with two class labels (+ and - signs). Algorithm 2 has two main cases: (1) instances that have a class label similar to the class label of their k nearest neighbors, represented as red instances in Fig. 5 (lines 6-15), and (2) instances that have a irrelevance function greater than or equal k/2, represented as blue instances in Fig. 5b (lines 17-19). The global density function in Eq. (3) is applied to the instances of the first case and the densest instances are added to the reduced set RS, as shown in Fig. 5c. The instances of the second case are added to the reduced set RS, as shown in Fig. 5c.

4. Implementation and experimental results

In this section, we briefly introduce the implementation of the proposed layered approach that produces a novel toolkit. We then introduce testbed, data sets, evaluation metrics, and comparisons through a set of experiments.

4.1. Implementations

We started with implementing our algorithms developed in Section 3.1 and the ISDF algorithms introduced in Section 2 using Java Archive. We then implemented three state-of-the-art IS algorithms (CNN, ENN, and ICF) to validate and assess the performance of our algorithms. We then implemented a suitable user interface (the first layer in our approach) and a user interface management and validation components (the second layer in our approach) using Java Archive as well. We finally combined all these implementations in an integrated open source toolkit, called Instance Selection Toolkit (ISTK) and is available on-demand, which is discussed in supplementary material.

4.2. Testbed and data sets

We perform an intensive set of 150 experiments using our ISTK: 7 algorithms × 18 data sets + 4 algorithms (our algorithms and ISDF algorithms) × 6 data sets. We exclude the state-of-the-art algorithms from being applied to six data sets because of their large sizes, which make these algorithms unable to process such data sets in acceptable time, specially ICF algorithm. The experiments are conducted with the following testbed.

1. Software and Hardware: Java 8, NetBeans IDE 8.2, LabTop with the following specification: processor is Intel(R) Core(TM) i5, 2.5GHz, 4GB of RAM, and Windows 7.
2. Validation procedure: the data sets are partitioned using the 10-FCV procedure (90% training set and 10% testing set). The procedure is repeated 10 times with different blocks for test-

Table 2
A summary of information about the data sets.

Name	#Inst.	#Attr.	#Clas.
German	1000	20	2
Contraceptive	1473	9	3
Yeast	1484	8	10
Car	1728	6	4
Titanic	2201	3	2
Segment	2310	20	7
Splice	3190	60	3
Chess	3196	36	2
Abalone	4174	8	28
Spam-base	4597	57	2
Banana	5300	2	2
Phoneme	5404	5	2
Page-blocks	5472	10	5
Texture	5500	40	11
Opt-digits	5620	64	10
Sat-image	6435	36	7
Thyroid	7200	21	3
Two-norm	7400	20	2
Ring	7400	20	2
Coil2000	9822	85	2
Nursery	12690	8	5
Fars	100968	29	8
Census	142521	41	3
LDPA	164860	7	11
Skin	245057	3	2

ing sets and then average results are reported for reduction rate and classification accuracy.

3. Parameters of the algorithms: the parameters of the ISDF algorithms and the state-of-the-art algorithms are as optimally chosen by their authors with $k = 3$ and we choose the same value for our algorithms. For evaluating the classification accuracy, we consider K-Nearest Neighbor classifier with $k = 1$ for all employed algorithms.
4. Data sets: we used twenty-four standard data sets taken from different business domains with different sizes ranging from a small, medium, large, to a very large volume: German, Contraceptive, Yeast, Car, Segment, Splice, Chess, Abalone, Spam-base, Banana, Phoneme, Page-blocks, Texture, Opt-digits, Sat-image, Thyroid, Ring, Two-norm, Coil2000, Nursery, Fars, Census, Localization Data for Person Activity (LDPA), and Skin Segmentation (Skin). These data sets are well-known and widely adopted in the literature. The LDPA and Skin Segmentation data sets are available at the UCI Repository,[2] while other data sets are available at the KEEL Repository.[3] Table 2 introduces some important information about the employed data sets.

4.3. Evaluation metrics

Three metrics are used to evaluate and compare the performance of the developed algorithms with the performance of the state-of-the-art algorithms and the ISDF algorithms. Good algorithm must essentially aim to strikingly balance amongst all of those metrics. These metrics are:

1. Reduction rate (Red.): it measures the storage reduction achieved by an IS method. Eq. (4) shows how to calculate the reduction rate.

$$Red. = 1 - \frac{|RS|}{|TS|} \tag{4}$$

2. Classification accuracy (Acc.): it measures the ability of a classification algorithms to classify data correctly using reduced set RS. It is computed as the division of the number of the instances correctly classified in testing set by total number of instances in testing set, as given in Eq. (5).

$$Acc. = \frac{sucess(Test)}{|Test|} \tag{5}$$

3. Effectiveness (Eff.): it measures the ability of an IS algorithm to balance between the reduction rate and the classification accuracy. It is computed as the product of reduction rate and classification accuracy, as given in Eq. (6).

$$Eff. = Red. \times Acc. \tag{6}$$

4.4. Experimental results and comparisons

Tables 3–5 show the classification accuracy, reduction rate, and effectiveness respectively for the state-of-the-art algorithms (CNN, ENN, and ICF), the ISDF algorithms (LDIS and CDIS), and the proposed algorithms (GDIS and EGDIS). Our proposed algorithms are compared to the ISDF algorithms using the twenty-four data sets in Table 2, while our algorithms are compared to the state-of-the-art algorithms using only the first eighteen data sets in Table 2. The reason behind that is the high computation time reported by the state-of-the-art algorithms to process large data sets. Therefore, we compute two averages in Tables 3–5: (1) average 1 is the average results for each algorithm over the eighteen data sets, and (2) average 2 is the average results for our algorithms and ISDF algorithms over the twenty-four data sets.

By analyzing results, the ENN algorithm is better than other algorithms regarding the classification accuracy (see Table 3) over 18 data sets. The ENN achieves better results for 11 data sets, while other algorithms achieve better results only for 7 data sets. It also reports the highest average classification accuracy (0.8057) over 18 data sets. The average classification accuracy results of the GDIS and ICF algorithms are approximately equal and they are better than the ISDF algorithms and the CNN algorithm. The GDIS algorithm outperforms the ICF algorithm in 12 data sets, while the ICF algorithm outperforms the GDIS algorithm in 6 data sets. The average classification accuracy of the GDIS algorithm (0.8015) is better than the average classification accuracy results of the ISDF algorithms (LDIS=0.7636 and CDIS=7789) over 24 data sets. Moreover, the GDIS algorithm outperforms the ISDF algorithms in 18 data sets, while the ISDF algorithms outperform the GDIS algorithm in 6 data sets.

The average reduction rate of the LDIS algorithm (0.8650) is higher than the average reduction rate of the state-of-the-art algorithms (CNN = 0.6904, ENN = 0.1847, and ICF = 0.8459), CDIS algorithm (0.8446), and GDIS algorithm (0.8453) over 18 data sets (see Table 4). The GDIS algorithm has better average reduction rate than the CDIS algorithm over 24 data sets. However, the LDIS algorithm achieves better reduction rate for 15 data sets, while the GDIS achieves better reduction rate only for 9 data sets. From Table 5, we found that the average effectiveness of the GDIS algorithm (0.6609) is approximately equal to the average effectiveness of the ICF algorithm (0.6687) over 18 data sets. Moreover, the average effectiveness of the GDIS algorithm (0.6780) is higher than the average effectiveness of the ISDF algorithms (LDIS = 0.6477 and CDIS = 0.6406) over 24 data sets.

To get a high effectiveness, which is a critical metric for most of application domains, our motivation of introducing the EGDIS algorithm is to improve the reduction rate results of the GDIS algorithm, while maintaining or improving its effectiveness results. On the one hand, EGDIS has an average reduction rate (0.8815) higher than the GDIS's average reduction rate (0.8476) over 24 data sets, as shown in Table 4. On the other hand, the improvement of the EGDIS reduction rate has a little impact on the classification accuracy and no negative impact on the effectiveness. The average

[2] https://archive.ics.uci.edu/ml/datasets.php.

[3] http://sci2s.ugr.es/keel/category.php?cat=clas.

Table 3
Results of classification accuracy over 24 data sets. Notice that the bolded numbers represent the best classification accuracy for each data set.

Name	CNN	ENN	ICF	LDIS	CDIS	GDIS	EGDIS
German	0.6540	**0.7370**	0.6950	0.6280	0.6250	0.6670	0.6630
Contraceptive	0.4108	0.4623	0.4582	0.4814	0.4882	**0.4895**	0.4698
Yeast	0.4791	0.5640	0.5438	0.4966	0.4960	**0.5707**	0.5013
Car	**0.8993**	0.8727	0.8831	0.7431	0.7743	0.8825	0.7477
Segment	0.9472	**0.9550**	0.9177	0.9074	0.9260	0.9277	0.9074
Splice	**0.7254**	0.6981	0.6934	0.7144	0.7072	0.7075	0.5639
Chess	0.9183	0.9462	**0.9553**	0.7710	0.7844	0.7669	0.7550
Abalone	0.1931	**0.2453**	0.2408	0.1945	0.1933	0.2415	0.1991
Spam-base	0.8571	**0.8928**	0.7248	0.7531	0.7657	0.7938	0.7611
Banana	0.8202	0.8904	0.8858	0.8619	0.8815	0.8891	**0.8947**
Phoneme	0.8494	**0.8781**	0.8484	0.8292	0.8558	0.8579	0.8294
Page-blocks	0.8712	**0.9607**	0.9099	0.9344	0.9393	0.9527	0.9454
Texture	0.9620	**0.9869**	0.9589	0.9367	0.9605	0.9533	0.9469
Opt-digits	0.9617	**0.9867**	0.9596	0.9819	0.9794	0.9776	0.9744
Sat-image	0.8715	**0.9030**	0.8564	0.8682	0.8720	0.8861	0.8744
Thyroid	0.8439	**0.9381**	0.9339	0.8615	0.8903	0.9311	0.9304
Two-norm	0.8809	**0.9561**	0.9054	0.9320	0.9354	0.9388	0.9469
Ring	0.8239	0.6269	**0.8350**	0.5697	0.6072	0.7064	0.5116
Coil2000	–	–	–	0.9088	0.8934	0.9369	**0.9370**
Nursery	–	–	–	0.5477	**0.7006**	0.6798	0.6400
Fars	–	–	–	0.7393	0.7411	**0.7626**	0.7488
Census	–	–	–	0.9166	0.9175	0.9433	**0.9441**
LDPA	–	–	–	0.7551	0.7632	**0.7750**	0.7543
Skin	–	–	–	**0.9992**	0.9956	0.9991	0.9991
Average 1	0.7761	0.8057	0.7892	0.7481	0.7601	0.7856	0.7457
Average 2	–	–	–	0.7636	0.7789	0.8015	0.7686

Table 4
Results of reduction rate over 24 data sets. Notice that the bolded numbers represent the best reduction rate for each data set.

Name	CNN	ENN	ICF	LDIS	CDIS	GDIS	EGDIS
German	0.4672	0.2759	0.8514	0.8699	0.8014	0.7996	**0.9173**
Contraceptive	0.2907	0.5503	0.7634	**0.9079**	0.8801	0.8631	0.8809
Yeast	0.3201	0.4656	**0.8582**	0.8075	0.7993	0.8476	0.8434
Car	0.7637	0.0700	0.3698	0.8609	0.8555	0.6826	**0.8899**
Segment	0.8684	0.0448	0.8218	**0.8730**	0.8724	0.8024	0.8366
Splice	0.6284	0.2291	0.6920	0.8088	0.7870	0.7442	**0.9210**
Chess	0.8179	0.0352	0.6236	0.9436	0.9276	0.9092	**0.9618**
Abalone	0.0822	0.7835	0.8685	0.7741	0.7743	**0.9421**	0.5596
Spam-base	0.7480	0.1069	**0.9717**	0.8998	0.8986	0.8400	0.9203
Banana	0.7695	0.1152	**0.9239**	0.8801	0.8045	0.8562	0.8987
Phoneme	0.7241	0.1118	0.8587	0.8637	0.8442	0.7958	**0.8805**
Page-blocks	0.9118	0.0408	**0.9667**	0.8662	0.8547	0.8614	0.8817
Texture	**0.9079**	0.0120	0.8778	0.9033	0.8658	0.8498	0.8712
Opt-digits	0.9184	0.0120	**0.9368**	0.7685	0.7693	0.8394	0.8475
Sat-image	0.7925	0.0907	**0.9300**	0.9166	0.8987	0.8288	0.8710
Thyroid	0.8190	0.0608	**0.9766**	0.9021	0.8646	0.8871	0.9161
Two-norm	0.8436	0.0353	0.9590	0.8594	0.8488	0.9358	**0.9668**
Ring	0.7534	0.2838	0.9755	0.8637	0.8555	0.9312	**0.9851**
Coil2000	–	–	–	0.8975	**0.9018**	0.8023	0.8456
Nursery	–	–	–	0.6847	0.7703	0.8421	**0.8759**
Fars	–	–	–	0.8079	0.8059	0.8439	**0.8954**
Census	–	–	–	0.8689	0.8730	0.8582	**0.8852**
LDPA	–	–	–	0.7686	0.7791	0.8060	**0.8303**
Skin	–	–	–	0.6807	0.4331	0.9738	**0.9740**
Average 1	0.6904	0.1847	0.8459	0.8650	0.8446	0.8453	**0.8805**
Average 2	–	–	–	0.8449	0.8236	0.8476	**0.8815**

classification accuracy of the GDIS algorithm is 0.8015 over 24 data sets, while the average classification accuracy of the EGDIS algorithm is 0.7686 over 24 data sets, as shown in Table 3. Finally, the effectiveness of the EGDIS algorithm is better than the GDIS algorithm. The average effectiveness of the EGDIS algorithm is 0.6849 over 24 data sets, while the average effectiveness of the GDIS algorithm is 0.6780 over 24 data sets, as shown in Table 5.

By having a high results in one metric (reduction rate or classification accuracy), we don't conclude that an algorithm is the best compared to other algorithms. The effectiveness metric represents the ability of an IS method to balance between the reduction rate and the classification accuracy. The high reduction rate with low classification accuracy gives low effectiveness and vice versa. For instance, assume we have an algorithm its reduction rate is 0.98 and classification accuracy is 0.43. This algorithm produces an effectiveness equal 0.42. The high value of effectiveness indicates a high reduction rate and high classification accuracy. Therefore, we can take the effectiveness metric in Eq. (6) to compare between the algorithms. The ICF algorithm and our proposed algorithms have the best average effectiveness over eighteen data sets (see Table 5), while our algorithms have the best average effectiveness over twenty-four data sets compared to the ISDF algorithms. We

Table 5
Results of effectiveness over 24 data sets. Notice that the bolded numbers represent the best effectiveness for each data set.

Name	CNN	ENN	ICF	LDIS	CDIS	GDIS	EGDIS
German	0.3056	0.2033	0.5918	0.5463	0.5009	0.5333	**0.6082**
Contraceptive	0.1194	0.2544	0.3498	**0.4370**	0.4296	0.4225	0.4138
Yeast	0.1534	0.2626	0.4667	0.4010	0.3965	**0.4837**	0.4228
Car	**0.6868**	0.0611	0.3266	0.6397	0.6624	0.6024	0.6654
Segment	0.8225	0.0428	0.7542	0.7921	**0.8078**	0.7444	0.7591
Splice	0.4559	0.1599	0.4798	**0.5778**	0.5566	0.5266	0.5194
Chess	0.7511	0.0333	0.5957	0.7275	**0.7276**	0.6972	0.7261
Abalone	0.0159	0.1922	0.2091	0.1506	0.1497	**0.2275**	0.1114
Spam-base	0.6411	0.0954	**0.7043**	0.6777	0.6881	0.6668	0.7004
Banana	0.6311	0.1026	**0.8185**	0.7586	0.7092	0.7612	0.8041
Phoneme	0.6150	0.0982	0.7286	0.7162	0.7225	0.6827	**0.7303**
Page-blocks	0.7943	0.0392	**0.8796**	0.8094	0.8029	0.8206	0.8335
Texture	**0.8734**	0.0118	0.8417	0.8462	0.8316	0.8101	0.8249
Opt-digits	0.8832	0.0119	**0.8990**	0.7546	0.7534	0.8206	0.8258
Sat-image	0.6907	0.0819	**0.7965**	0.7958	0.7836	0.7344	0.7616
Thyroid	0.6911	0.0571	**0.9121**	0.7772	0.7697	0.8260	0.8523
Two-norm	0.7432	0.1787	0.8683	0.8010	0.7940	0.8785	**0.9154**
Ring	0.6207	0.0337	**0.8146**	0.4921	0.5194	0.6577	0.5040
Coil2000	–	–	–	**0.8156**	0.8057	0.7517	0.7923
Nursery	–	–	–	0.3750	0.5397	**0.5724**	0.5605
Fars	–	–	–	0.5972	0.5973	0.6436	**0.6705**
Census	–	–	–	0.7964	0.8010	0.8095	**0.8357**
LDPA	–	–	–	0.5804	0.5947	0.6246	**0.6263**
Skin	–	–	–	0.6801	0.4312	0.9730	**0.9731**
Average 1	0.5830	0.1067	**0.6687**	0.6500	0.6447	0.6609	0.6655
Average 2	–	–	–	0.6477	0.6406	0.6780	**0.6849**

Table 6
Wilcoxon signed-rank test results for classification accuracy.

Algorithm 1	Algorithm 2	$w+$	$w-$	Test statistic	Critical value	Status
GDIS	LDIS	289	-11	11	91	Accept H_1
GDIS	CDIS	261	-39	39	91	Accept H_1
GDIS	CNN	102	-69	69	47	Accept H_0
GDIS	ENN	46	-125	46	47	Accept H_1
GDIS	ICF	116	-55	55	47	Accept H_0
EGDIS	LDIS	209	-91	91	91	Accept H_0
EGDIS	CDIS	127	-173	127	91	Accept H_0
EGDIS	GDIS	23	-277	23	91	Accept H_1
EGDIS	CNN	72	-99	72	47	Accept H_0
EGDIS	ENN	3	-168	3	47	Accept H_1
EGDIS	ICF	52	-119	52	47	Accept H_0

can conclude that the ICF algorithm and our proposed algorithms have a better ability to balance between the reduction rate and classification accuracy metrics than other algorithms.

4.5. Statistical analysis

We use the Wilcoxon signed-rank test (recommended in García et al. (2015)) to determine if there is a statistical significant difference between each pairs of algorithms. The GDIS algorithm is statistically compared to the state-of-the-art algorithms and the ISDF algorithms, while the EGDIS algorithm is statistically compared to the state-of-the-art algorithms, the ISDF algorithms, and the GDIS algorithm. We consider two hypotheses (Hypotheses H_0 and H_1) with significance level $\alpha = 0.1$:

Hypothesis H_0. There is no difference between two algorithms.

Hypothesis H_1. There is a difference between two algorithms.

We report the Wilcoxon signed-rank test results in Tables 6–8 for classification accuracy, reduction rate, and effectiveness respectively. In the tables, $W+$ indicates the sum of the positive ranks, $W-$ indicates the sum of the negative ranks, test statistic is the minimum absolute value of $W+$ and $W-$, critical value is the distribution value for Wilcoxon using α and number of data sets (note that the number of data sets is 24 when comparing our algorithms and the ISDF algorithms, while the number of data sets is 18 when comparing our algorithms and the state-of-the-art algorithms), and status represents which hypothesis is accepted (if test statistic $<$ critical value, we accept H_1, otherwise we accept H_0). From Tables 6–8, we have the following remarks:

1. There is no significant difference between our GDIS algorithm and the CNN and ICF algorithms in term of classification accuracy.
2. Our GDIS algorithm improves the classification accuracy results of the ISDF algorithms.
3. There is no significant difference between our GDIS algorithm, the ISDF algorithms, and the ICF algorithm in terms of reduction rate and effectiveness.
4. Our GDIS algorithm outperforms the CNN and ENN algorithms in terms of reduction rate and effectiveness.
5. There is no significant difference between our EGDIS algorithm, the ISDF algorithms, the CNN algorithm, and the ICF algorithm in term of classification accuracy.
6. The ENN and GDIS algorithms outperforms the EGDIS algorithm in term of classification accuracy.
7. Our EGDIS algorithm outperforms the ISDF algorithms, the GDIS algorithm, the CNN algorithm, and the ENN algorithm in terms of reduction rate and effectiveness.
8. There is no significant difference between our EGDIS algorithm and the ICF algorithm in terms of reduction rate and effectiveness.

5. Scalability and time complexity

Table 9 shows the computation time in seconds for the state-of-the-art algorithms, the ISDF algorithms, and our algorithms over 24 data sets. By analyzing the reported results in Table 9, we found that: (1) the ISDF algorithms, the CNN algorithm, and the ENN algorithm require less computation time than other algorithms for most data sets, (2) our developed algorithms require extremely lower computation time than the ICF algorithm, (3) the computation time of the ISDF algorithms is clearly better for the data sets containing a large number of classes (e.g., Abalone and Sat-image data sets), (4) the computation time of the developed algorithms

Table 7
Wilcoxon signed-rank test results for reduction rate.

Algorithm 1	Algorithm 2	*w*+	*w*−	Test statistic	Critical value	Status
GDIS	LDIS	136	−164	136	91	Accept H_0
GDIS	CDIS	170	−130	130	91	Accept H_0
GDIS	CNN	147	−24	24	47	Accept H_1
GDIS	ENN	171	0	0	47	Accept H_1
GDIS	ICF	65	−106	65	47	Accept H_0
EGDIS	LDIS	219	−81	81	91	Accept H_1
EGDIS	CDIS	248	−52	52	91	Accept H_1
EGDIS	GDIS	274	−26	26	91	Accept H_1
EGDIS	CNN	161	−10	10	47	Accept H_1
EGDIS	ENN	170	−1	1	47	Accept H_1
EGDIS	ICF	90	−81	81	47	Accept H_0

Table 8
Wilcoxon signed-rank test results for effectiveness.

Algorithm 1	Algorithm 2	*w*+	*w*−	Test statistic	Critical value	Status
GDIS	LDIS	193	−107	107	91	Accept H_0
GDIS	CDIS	201	−99	99	91	Accept H_0
GDIS	CNN	132	−39	39	47	Accept H_1
GDIS	ENN	171	0	0	47	Accept H_1
GDIS	ICF	64	−107	64	47	Accept H_0
EGDIS	LDIS	227	−73	73	91	Accept H_1
EGDIS	CDIS	217	−83	83	91	Accept H_1
EGDIS	GDIS	218	−82	82	91	Accept H_1
EGDIS	CNN	140	−31	31	47	Accept H_1
EGDIS	ENN	171	0	0	47	Accept H_1
EGDIS	ICF	75	−96	75	47	Accept H_0

Table 9
The computation time in seconds of the employed algorithms over 24 data sets.

Name	CNN	ENN	ICF	LDIS	CDIS	GDIS	EGDIS
German	4.8	1.5	22.9	8.7	13.4	14.6	14.1
Contraceptive	6.8	2.1	6.9	9.6	17.5	28.7	27.8
Yeast	7.6	2.4	16.5	2.6	2.0	3.2	3.1
Car	4.0	2.8	38.7	2.9	3.2	3.7	3.5
Segment	4.6	6.6	878.7	40.6	60.6	283.8	283.3
Splice	145.7	84.8	501.3	39.8	53.9	119.7	107.8
Chess	28.5	19.0	245.9	51.9	68.7	97.8	94.8
Abalone	64.1	16.0	21.1	7.3	5.9	15.9	17.2
Spam-base	79.0	60.7	6312.9	1269.9	1671.9	2342.3	2310.8
Banana	18.6	15.6	3258.0	74.2	114.5	155.1	154.0
Phoneme	31.6	21.1	6118.6	218.3	328.4	364.9	361.7
Page-blocks	15.4	40.0	49569.5	53.9	72.9	66.8	63.9
Texture	28.2	53.6	14023.7	273.9	389.1	2852.0	2899.9
Opt-digits	69.0	180.3	28066.6	21.2	24.3	360.1	346.3
Sat-image	69.9	79.1	18503.8	591.8	755.8	2944.7	2954.3
Thyroid	66.1	83.6	41153.5	152.3	175.0	172.3	174.2
Ring	157.5	128.5	76398.1	86.1	107.5	205.3	221.0
Two-norm	132.7	146.7	23022.2	88.3	110.1	237.4	258.3
Coil2000	–	–	–	12478.4	15853.0	14148.2	14004.9
Nursery	–	–	–	177.0	289.0	409.0	399.9
Fars	–	–	–	9285.0	11148.1	38782.5	40769.8
Census	–	–	–	99734.7	131364.1	113319.3	126815.9
LDPA	–	–	–	4731.5	6460.2	35407.6	38821.2
Skin	–	–	–	34507.9	46982.1	56427.0	68014.8

(GDIS and EGDIS) is close to the computation time of the CDIS algorithm for the data sets that contain two or three classes (e.g., Phoneme and Thyroid), and (5) the CDIS algorithm requires a high computation time for data sets containing a large number of instances and features (see, for example, the results of the Coil2000 data set that contains 85 features).

We exclude the state-of-the-art algorithms from the scalability and time complexity study due to two main reasons: (1) the low reduction rate and effectiveness results for the CNN and ENN algorithms compared to other algorithms (see Sections 4.4 and 4.5) and (2) the extremely high computation time reported by the ICF algorithm to operate on large data sets (see Table 9). Therefore, we only evaluate the scalability and compute the time complexity for the ISDF algorithms and our algorithms (Algorithms 1 and 2) by conducting five experiments for each employed data set. Particularly, we consider 20% of the data set in the first experiment and 40% of the data set in the second experiment. We continue this scaling-up percentage till we get 100% of the data set in the last experiment. We then draw the computation times (Y's values) as a function of the data set size (X's values) for each data set in Figs. 6–9. Technically, we used the values of *X* and *Y* pairs for each data set as a training set to get the best fit linear regression func-

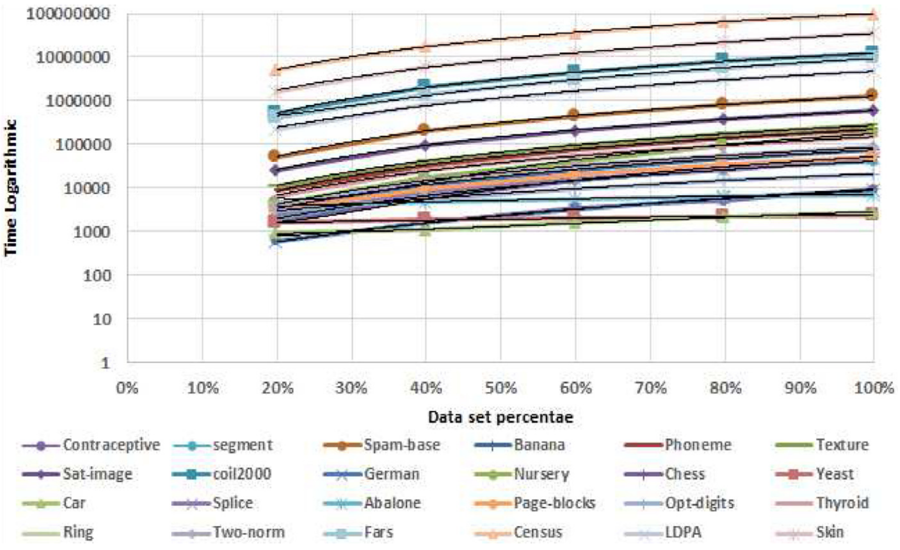


Fig. 6. The computation time in milliseconds of LDIS over 24 data sets with polynomial trends.

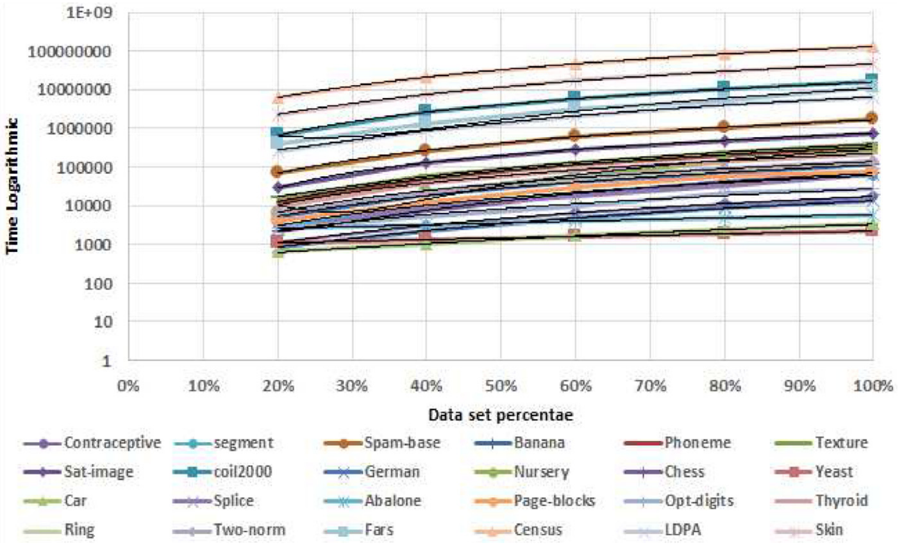


Fig. 7. The computation time in milliseconds of CDIS over 24 data sets with polynomial trends.

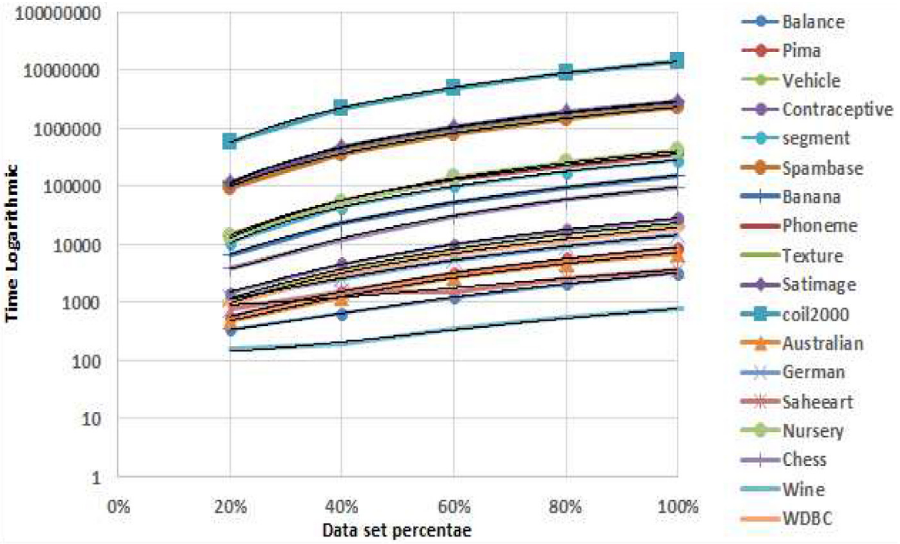


Fig. 8. The computation time in milliseconds of GDIS over 24 data sets with polynomial trends.

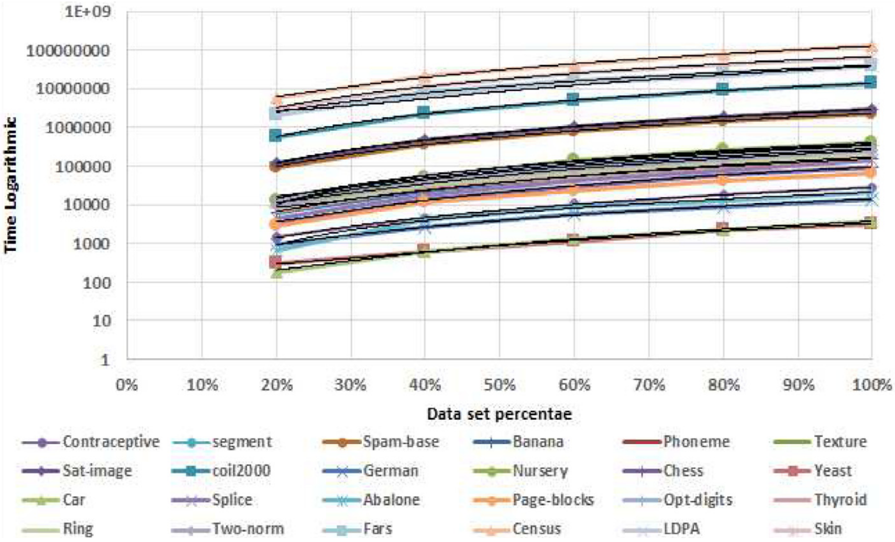


Fig. 9. The computation time in milliseconds of EGDIS over 24 data sets with polynomial trends.

tion/model, as shown in Figs. 6-9. The obtained regression function (also called polynomial trend) is of kind polynomial with degree two. For example, Eq. (7) can be used to compute the computation time in milliseconds (Y) given a data set size (X) for EGDIS algorithm over Chess data set.

$$Y = 113320X^2 - 22243X + 3555.4 \tag{7}$$

Indeed, all the computed polynomial models have a small p-value ($p < 0.0001$), which rejects the null hypothesis and proves that there is a strong association between the predictor (X) and the response (Y). Also, its t-distribution is approximately normal distribution. Moreover, we computed the R^2 statistic and we found that R^2 is close or exactly equal to one in all polynomial trends, indicating that a large proportion of the variability in Y can be explained by the regression. All these computations (p-value, t-statistic, and R^2) prove the best fitness of the obtained polynomial models. For the readability purpose, we removed polynomial functions and R^2 values from Figs. 6–9.

6. Overall discussion and conclusion

The research community is normally focused on satisfying one or two criteria (reduction rate, classification accuracy, effectiveness, and computation time). However, we changed by somehow this acceptable principle to satisfy all these criteria in one approach by improving the performance of the ISDF algorithms. We technically converted the scope of the density function from local (for each class) to global (for the whole data set). By doing so, the points between classes (i.e., border points) were considered that have a good impact on the classification accuracy results. The GDIS algorithm (the novel and the first contribution) applies a relevance function on these points to determine which points are added to reduced set beside the densest instance. The results showed an improvement in the classification accuracy and effectiveness metrics compared to the ISDF algorithms. The algorithm improves the classification accuracy of 18 data sets from 24 employed data sets compared to the ISDF algorithms and has a better average classification accuracy (0.8015) than the ISDF algorithms (LDIS = 0.7636 and CDIS = 0.7789) over 24 data sets. It achieves an average effectiveness (0.6780) which is higher than the average effectiveness of the ISDF algorithms (LDIS = 0.6477 and CDIS = 0.6406) over 24 data sets.

In order to improve the reduction rate and effectiveness results of the GDIS algorithm, the EGDIS algorithm is developed (the novel and the second contribution). It uses another function called irrelevance function beside the global density function. The irrelevance function adds only instances that may be misclassified beside densest instances to the reduced set. The results of EGDIS algorithm showed an improvement in the average reduction rate (0.8815) compared to the GDIS algorithm (0.8476) and the ISDF algorithms (LDIS = 0.8449 and CDIS = 0.8236) over 24 data sets. It achieves an average classification accuracy (0.7686) approximately equal to the classification accuracy of the LDIS algorithm (0.7636) over 24 data sets. It also has the highest average effectiveness (0.6849) compared to GDIS algorithm (0.6780) and ISDF algorithms (LDIS = 0.6477 and CDIS = 0.6406) over 24 data sets. We conclude that our algorithms achieve a better balance between reduction rate and classification accuracy than the ISDF algorithms (the novel and the third contribution). Moreover, we compared our algorithms with three standard and well known algorithms (CNN, ENN, and ICF) over 18 data sets to assess their performance. The results showed that our algorithms outperform the CNN and ENN algorithms in terms of reduction rate and effectiveness. The average effectiveness of our algorithms is approximately equal to the average effectiveness results of the ICF algorithm, but our algorithms require extremely lower computation time than ICF. We also performed a statistical analysis using Wilcoxon signed-rank test between our algorithms and other employed algorithms. The statistical analysis proved that: (1) our GDIS algorithm is better than the ISDF algorithms in term of classification accuracy, and (2) our EGDIS algorithm is better than the ISDF algorithms in terms of reduction rate and effectiveness.

The time complexity (the novel and the fourth contribution) of the developed algorithms (GDIS and EGDIS) is polynomial of order two that is a positive result. The empirical and theoretical computations of time complexity are missing entirely in the literature. For data sets that contain only two classes, the computation time of developed algorithms is approximately equal to the computation time of the CDIS algorithm. The computation time of our algorithms is notably higher than the computation time of the ISDF algorithms (LDIS and CDIS) for data sets containing a large number of classes. The overall conclusion is that the computation time of current ISDF algorithms and our algorithms is still increasing when the size of the data set is going larger, as normally expected.

There are several directions of future work. We plan to continue and propose a new density function and enhance results by adding improvements to the proposed algorithms. Moreover, we plan to implement our algorithms in a parallel framework to consider big data sets used recently in a cloud-based system to compare our results (Sinnott & Voorsluys, 2016).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.eswa.2020.113297.

Credit authorship contribution statement

Mohamed Malhat: Data curation, Software, Writing - original draft, Writing - review & editing, Visualization. **Mohamed El Menshawy:** Methodology, Formal analysis, Writing - review & editing. **Hamdy Mousa:** Investigation, Validation. **Ashraf El Sisi:** Conceptualization, Supervision, Project administration.

References

Arnaiz-González, Á., González-Rogel, A., Díez-Pastor, J.-F., & López-Nozal, C. (2017). Mr-dis: Democratic instance selection for big data by mapreduce. *Progress in Artificial Intelligence, 6*(3), 211–219.

Bai, L., Liang, J., Dang, C., & Cao, F. (2012). A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications, 39*(9), 8022–8029.

Bolt, A., Leoni, M., & Aalst, W. M. (2016). Scientific workflows for process mining: Building blocks, scenarios, and implementation. *International Journal on Software Tools for Technology Transfer, 18*(6), 607–628.

Brighton, H., & Mellish, C. (2002). Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery, 6*(2), 153–172.

Carbonera, J. L., & Abel, M. (2015). A density-based approach for instance selection. *IEEE 27th International conference on tools with artificial intelligence (ICTAI)*.

Carbonera, J. L., & Abel, M. (2016). A novel density-based approach for instance selection. *IEEE 28th International conference on tools with artificial intelligence (ICTAI)*.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications, 19*(2), 171–209.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

Do, H.-H., & Rahm, E. (2007). Matching large schemas: Approaches and evaluation. *Information Systems, 32*(6), 857–885.

Gantz, J., & Reinsel, D. (2011). Extracting value from chaos. *Technical Report*. IDC iView.

García, S., Cano, J. R., & Herrera, F. (2008). A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition, 41*(8), 2693–2709.

García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (1st ed.). Springer International Publishing.

García-Osorio, C., de Haro-García, A., & García-Pedrajas, N. (2010). Democratic instance selection: A linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence, 174*(5), 410–441.

de Haro-García, A., & García-Pedrajas, N. (2009). A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Mining and Knowledge Discovery, 18*(3), 392–418.

Hart, P. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory, 14*(3), 515–516.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of big data on cloud computing: Review and open research issues. *Information Systems, 47*, 98–115.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. (2003). A taxonomy of dirty data. *Data Mining and Knowledge Discovery, 7*(1), 81–99.

Leyva, E., González, A., & Pérez, R. (2015). Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. *Pattern Recognition, 48*(4), 1523–1537.

Liu, C., Wang, W., Wang, M., Lv, F., & Konan, M. (2017). An efficient instance selection algorithm to reconstruct training set for support vector machine. *Knowledge-Based Systems, 116*(Supplement C), 58–73.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery, 6*(4), 393–423.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Norwell, MA, USA: Kluwer Academic Publishers.

Liu, H., & Motoda, H. (2001). *Instance selection and construction for data mining*. Norwell, MA, USA: Kluwer Academic Publishers.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review, 34*(2), 133–143.

Ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah, T. Y., & Khan, S. U. (2016). Big data reduction methods: A survey. *Data Science and Engineering, 1*(4), 265–284.

Reinartz, T. (2002). A unifying view on instance selection. *Data Mining and Knowledge Discovery, 6*(2), 191–210.

Silva, D. A., Souza, L. C., & Motta, G. H. (2016). An instance selection method for large datasets based on Markov geometric diffusion. *Data & Knowledge Engineering, 101*, 24–41.

Sinnott, R. O., & Voorsluys, W. (2016). A scalable cloud-based system for data-intensive spatial analysis. *International Journal on Software Tools for Technology Transfer, 18*(6), 587–605.

Team, R. O. (2011). Big data now: current perspectives from OReilly Radar. *Technical Report*. OReilly Media.

Turner, K. J., & Lambert, P. S. (2015). Workflows for quantitative data analysis in the social sciences. *International Journal on Software Tools for Technology Transfer, 17*(3), 321–338.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-2*(3), 408–421.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning, 38*(3), 257–286.

Zhang, S., Zhang, C., & Yang, Q. (2003). Data preparation for data mining. *Applied Artificial Intelligence, 17*(5-6), 375–381.