Hindawi
Complexity
Volume 2018, Article ID 7404627, 13 pages
https://doi.org/10.1155/2018/7404627

WILEY | Hindawi

Research Article

An Approach to Data Reduction for Learning from Big Datasets: Integrating Stacking, Rotation, and Agent Population Learning Techniques

Ireneusz Czarnowski and Piotr Jędrzejowicz

Department of Information Systems, Gdynia Maritime University, Morska 83, 81-225 Gdynia, Poland

Correspondence should be addressed to Ireneusz Czarnowski; irek@am.gdynia.pl

Received 26 June 2018; Revised 3 September 2018; Accepted 16 September 2018; Published 5 November 2018

Academic Editor: Sergio Gómez

In the paper, several data reduction techniques for machine learning from big datasets are discussed and evaluated. The discussed approach focuses on combining several techniques including stacking, rotation, and data reduction aimed at improving the performance of the machine classification. Stacking is seen as the technique allowing to take advantage of the multiple classification models. The rotation-based techniques are used to increase the heterogeneity of the stacking ensembles. Data reduction makes it possible to classify instances belonging to big datasets. We propose to use an agent-based population learning algorithm for data reduction in the feature and instance dimensions. For diversification of the classifier ensembles within the rotation also, alternatively, principal component analysis and independent component analysis are used. The research question addressed in the paper is formulated as follows: does the performance of a classifier using the reduced dataset be improved by integrating the data reduction mechanism with the rotation-based technique and the stacking?

1. Introduction

Big data, so far, does not have a formal definition, although it is generally accepted that the concept refers to datasets that are too large to be processed using conventional data processing tools and techniques. Contemporary information systems produce data in huge quantities that are difficult to be measured [1]. It means that we already have found ourselves in the "big data era," and the question of how to solve large-scale machine learning problems is open and requires a lot of research efforts. Dealing with huge datasets poses a lot of the processing challenges. The big data sources including contemporary information systems and databases contain inherently complex data characterized by the well-known 5V properties: huge volume, high velocity, much variety, big variability, low veracity, and high value [2].

The big data applications involve four major phases: data generation, data management, data analytics, and data application. The data analytics is the most important phase, where the aim is to discover patterns from data. However, in the big data era, the task is not trivial and much more complicated than normal-sized data analytics [3]. This becomes especially troublesome in numerous critical domains like security, healthcare, finance, and environment protection, where obtaining a dependable knowledge of different processes and their properties is crucial to the social welfare.

Learning from data is an example of the most important data analytics problem, where machine learning algorithms are used. The aim of the machine learning is to expand algorithms that are able to learn through experience [4]. The algorithms, called learners, can improve their performance based on analysis of the collected data, which are called examples [5], and which are collected from the environment. Today, machine learning offers a wide range of tools and methods that can be used to solve a variety of data mining problems. Their common weakness is, however, the

so-called dimensionality curse, making them inefficient or even useless when solving large-scale problems. Thus, achieving scalability, low computational complexity, and efficient performance of the machine learning algorithms have become hot topics for the machine learning community.

Since traditional techniques used for analytical processing are not fit to effectively deal with the massive datasets, searching for new and better techniques, methods, and approaches suitable for big data mining is a hot area for the machine learning community. Considering the above facts and observing current trends in the machine learning research, it can be observed that among main contemporary challenges, the most important one is a search for improvements with respect to scalability and performance of the available algorithms. Among techniques for dealing with massive datasets are different parallel processing approaches aiming at achieving a substantial speed-up of the computation. Examples of such techniques are Hadoop and MapReduce techniques which have proven suitable for the computation and data intensive tasks [6].

The scalability and performance issues lead to the two simple questions: "how fast?" and "how large?," that is, how fast one can get a solution and how large is a dataset one can effectively deal with. In this paper, we focus on the question of "how large?," and we analyze approaches to deal with big data. In reference to a short discussion on fundamental strategies for big data analytics included in [3], the following approaches are currently considered as the most promising ones:

(i) Divide-and-conquer

(ii) Parallelization

(iii) Sampling

(iv) Granular computing

(v) Feature selection

Divide-and-conquer is a well-known strategy based on processing small chunks of data and then fusing separated results together.

Parallelization concerns dividing a large problem into several smaller ones which can be solved concurrently in parallel, producing, in the end, the final result.

Sampling is a well-known statistical technique based on the probability theory. The approach is based on identifying a relationship between the sample and the population. With the advent of the big data era, many new sampling techniques have emerged or have been modified including simple random sampling, systematic sampling, stratified sampling, cluster sampling, quota sampling, and minimum-maximum sampling [7].

Granular computing is a technique using granules to build an efficient computational model for complex applications in the big data environment. Examples of these granules are classes, clusters, subsets, groups, and intervals. From the implementation point of view, the technique reduces the data size through analyzing data at different levels of granularity [8].

Feature selection is a technique for dimensionality reduction in a feature space [9]. The aim of the feature selection is to obtain a representative subset of features that has fewer features in comparison to the original feature set. Several different techniques have been proposed for feature selection, so far. The feature extraction technique is one of the possible approaches.

The above-described strategies are in line with techniques proposed to achieve better scalability of the machine learning algorithms. In [10], such techniques were classified into the three categories. The first includes extensions and modification of the traditional machine learning tools. The second is based on the problem decomposition into a set of smaller or computationally less complex problems. The third involves using parallel processing where possible. In this paper, we use the idea of the problem decomposition. The paper is an extension of the earlier research results included in [11] and presented during the 2017 IEEE INISTA Conference. The extension involves an improvement of the stacking and rotation procedures allowing for either deterministic or random transformations in the feature space. The above option improves the performance of the procedure. The paper also refers to and offers some extensions of the research results included in other papers of the authors' [12–14].

The paper considers an approach dedicated to reducing the dimensionality in data, so this also means that it is dedicated to working with large datasets, with a view to enabling an efficient machine learning classification in terms of a high classification accuracy and an acceptable computation time. To achieve the above, the following techniques are used:

(i) Data reduction based on the prototype selection through the instance and feature selections from clusters of instances

(ii) Stacking

(iii) Rotation-based

(iv) Agent-based population learning algorithm for data reduction

The research question addressed in the paper is formulated as follows: does the performance of a classifier over the reduced dataset be improved by integrating the data reduction mechanism with the rotation-based technique and the stacking? In [11], to diversify the classifier ensembles, the rotation-based techniques using principal component analysis for feature selection have been implemented. In this paper, the alternatively independent component analysis method and feature selection based on an agent-based population learning algorithm implementation is used. We also propose to use an agent-based population learning algorithm for data reduction in the instance dimension. The techniques used have been integrated, and an adaptive approach to constructing the machine classifiers is proposed. The approach is validated experimentally by solving selected classification problems over benchmark datasets from UCI and the KEEL repositories [15, 16].

The paper is organized as follows. A brief review of the stacking, rotation, and data reduction is included in the next section. The following section provides a detailed description of the proposed approach. Next, computational experiment carried out, including its plan and results, is described and discussed. The final section focuses on conclusions and ideas for further research.

2. Techniques for Improving Performance for Big Data Analytics

In this section, a brief review of the data reduction techniques, the rotation-based technique and the agent-based population learning algorithm (PLA), as a background for further consideration, is offered.

2.1. Data Reduction. Reducing the quantity of data aims at selecting pertinent information only as an input to the data mining algorithm. Thus, data reduction identifies and, eventually, leads to discarding information which is irrelevant or redundant. Ideally, after data reduction has been carried out, the user has to do with datasets of smaller dimensions representing the original dataset. It is also assumed that the reduced dataset carries the acceptable or identical amount of information as the original dataset.

Data reduction aim is not losing extractable information but to increase the effectiveness of the machine learning when the available datasets are large [4]. It is the most critical component in retrieving information from big data in many data mining processes [17].

Reducing data size may cover for the unwanted consequences of scaling up. Among such consequences, specialists list excessive memory requirements, increasing computational complexity and deteriorating learning performance [17].

In practice, data dimensionality reduction is concerned with selecting informative instances and features from the training dataset. In the literature on data reduction, quite often, instance and feature selections are addressed separately. There exist also approaches where both tasks are solved simultaneously as a dual selection problem [18]. Data reduction can be also merged with the problem of the prototype extraction.

The prototype extraction problem also aims at reducing the dimensionality of the training set by replacement of the existing instances by the extracted ones. Extracting prototypes from the original dataset may also include constructing new features. In such case, a smaller number of features are constructed from the original feature set through certain transformation operations [19]. A well-known tool for carrying such transformation is the principal component analysis (PCA) [20].

More on the data reduction problem as well as a review of the proposed approaches including instance selection, feature selection, and the dual dimension data reduction can be found among others in [2, 9, 14, 21–23].

Formally, the data reduction process aims at finding the reduced dataset S_{opt}, which is the subset of the original dataset D, such that the performance criterion of the machine learning algorithm L is maximized. From the above perspective, the performance of the classifier induced from the reduced dataset should be better or at least not worse than the classifier induced from the original dataset [24].

In this paper, the approach to data reduction is proposed as a tool for dimensionality reduction of the original dataset and is carried out in both dimensions (instance and feature). Moreover, in this paper, the implementation of data reduction is an example of the idea of data partitioning (as suggested in [10]), as well as an exemplification of the strategy of granular computing (in a sense proposed in [3]).

2.2. Stacked Generalization. Stacked generalization also known as stacking was proposed by Wolpert [25]. The technique was designed to improve classification algorithm performance.

Stacking is an example of a sampling strategy and is one of the ensemble learning techniques. The idea of stacking is based on combining the multiple classifications or regression models via a metaclassifier or a metaregressor.

In stacking, the base learners consist of different learning algorithms, so the stacking ensembles are often heterogeneous. Performance of the stacking-based classifiers is competitive in comparison with learners using bagging and boosting techniques. Besides, stacking allows for combining learners of the different types, which is not the case in bagging and boosting. Stacked generalization can be implemented using one of the two modes for combining, the so-called, base classifiers or combining their output. In the first mode, outputs from base classifiers are combined to obtain the final classification decision. In the second mode, base classifiers are used to construct the metamodel used for predicting unknown class labels.

In the vast literature on the stacked generalization, there are two basic approaches to combining base classifiers. The first one assumes combining, at a higher level, outputs from the base classifiers to obtain classification decision. Alternatively, at a higher level, base classifiers are integrated into the metamodel, subsequently used to predict unknown class labels.

In the standard stacking approach, at first q, different instance subsets of equal size are generated using a random generator. It is assumed that the subsets will be generated in such a way that assures relative proportion of instances from the different classes like it is observed in the original dataset. In the next step, omitting one of the subsets in each iteration, the so-called level-0 classifiers are generated from the remaining subsets. The process is repeated q times following the pattern of the q-fold cross-validation procedure. At each iteration, the omitted subset of instances is used to generate the so-called level-1 set of instances. Thus, the level-0 models produce predictions that form the input to the level-1 model. They are used to predict the class label for new instances with unknown class labels. In the approach, the metaclassifier in the form of relative weight for each level-0 classifier is created by assigning weights to classifiers proportional to their performance. The schema for metaclassifier induction has a form of the so-called leave-one-out cross-validation [25].

Thus, combining classifiers under the umbrella of stacking can be seen as follows. Supposing that there are q different learners $L_1, \ldots, L_q$ and q different training sets, $D_1, \ldots, D_q$, where $D = D_1 \cup D_2 \ldots \cup D_q$ and D is the original training set. Each learner is induced from training sets $D_1, \ldots, D_q$, respectively. As the result, we have the output hypotheses $h_1, \ldots, h_q$, where $\forall h_{i:i=1,\ldots,q} \in H$ and H is a hypothesis space, which is defined as a set of all possible hypothesis, that the learner can draw. Thus, the goal of stacking is to learn a well-combined classifier h such that the final classification will be computed from $h_1(x), \ldots, h_q(x)$ as shown in the equation:

$$h(x) = \sum_{i=1}^{q} w_i h_i(x), \tag{1}$$

where vector w represents the respective weights.

Different variants of stacking have been proposed so far. A review of the stacking algorithms is included, for example, in [25] or [26].

In this paper, the stacking technique used has been inspired by Skalak's proposal [27], where the prototype selection on the level-0 of the stacking is carried out as the mean for data reduction. Next, the outputs of the level-0 are used for generating the metaclassifier at the level-1. In this paper, we also assume that the data reduction is carried out through prototyping and that prototypes are selected from the clusters, which are induced during the carried out data analysis. Stacking plays the role of the sampling strategy paradigm and helps with achieving a diversification of the level-0 models.

2.3. Rotation-Based Technique. The rotation-based technique belongs to the family of the ensemble methods, while in turn, the ensemble methods can be seen as meta-algorithms. The rotation-based technique combines several machine learning techniques into one predictive model aiming at improving the machine learning performance. The rotation-based technique belongs to the class of the multiple classifier systems (MCS) described in [28]. The idea behind the rotation-based ensembles (RE) is to use the rotation operator to project or transform the original dataset into a new feature space. From such feature space, new features are extracted. To implement the approach, the following two steps are executed. First, the original dataset is projected into a new feature space. From such space, at the second step, feature subsets are selected, and base individual classifiers are induced. The procedure is expected to improve the classification accuracy as compared with the traditional approach. It is known that the approach is usually effective when classifying high dimensional data [29].

Well-known example of the RE is the rotation forest (RF) algorithm. Rotation forest extends the idea of the random forest, which combines the bagging and the random subspace methods [30]. Random forest consists of a number of decision trees trained based on the example bootstraps sampled from the original training set. Each subset of the training dataset is modified by selecting randomly a subset of features. The RF procedure starts with the feature extraction from the input data followed by training of each decision tree in a different new rotated space. The process results in achieving, at the same time, a high individual accuracy and the required diversity among the ensemble members. Four feature extraction methods, principal component analysis (PCA), maximum noise fraction (MNF), independent component analysis (ICA), and local fisher discriminant analysis (LFDA), have been applied in the rotation forest [30, 31].

In [23], feature extraction and data transformation were based on the principal component analysis (PCA). How exactly to apply PCA depends on the user. One possible way is to apply it to a subset of features only. In such case, one has to split the original set of features to a number of subsets associating with each subset a subset of instances through the axis rotation [32]. The approach suffers from one drawback. Since PCA is a deterministic algorithm, it may generate the ensemble with members characterized by the identical set of features. To avoid such a situation, some diversification mechanisms like, for example, removing some instances from the dataset are often used [33].

The experimental results show that the rotation forest is on the average superior and can produce more accurate results than bagging, AdaBoost, random subspace, and random forest [29, 31].

In the proposed approach, generation of base classifiers through feature rotation has been integrated with stacking and data reduction. It is shown experimentally that such an integration assures better diversification of the base classifier ensemble and, consequently, better classification performance. Two approaches are applied to the feature space modification. In the first one, the original RF algorithm is used. In the second case, the feature space is modified through solving the respective optimization problem using the agent-based population learning algorithm (described in the next subsection).

2.4. Agent-Based Population Learning Algorithm. The agent-based population learning algorithm seems to be a promising tool for solving complex computational problems arising in the big data environment. During the last years, the idea of implementing the agent-based approaches for the big data analytics is a hot topic. Examples and exchange of ideas in the above respect can be found in a special issue of *Web Intelligence and Agent Systems: An International Journal* [34]. The subject has been also discussed during the international conferences (for example, Metaheuristics International Conference, IEEE/WIC/ACM International Conference on Intelligent Agent Technology). The implementation of the agent-based approach has been also a subject of the paper [35]. An agent-based paradigm and the example case study have been also discussed in the context of applying the big data analytics in retailing [36].

Recent advances in distributed problem solving and agent-based data mining confirm that both techniques can help work with data extracted from the distributed and online environments. Agent-based technologies offer a variety of mechanisms that can significantly reduce costs of processing a large volume of data and improve data processing

Generate the initial population of solutions (individuals) and store them in the common memory
Implement different improvement procedures executed by the optimization agents
Activate optimization agents
While (stopping criterion is not met) **do** {in parallel}
  Read randomly selected individual from the common memory
  Execute improvement algorithm
  Store the improved individual back in the common memory
End while
Take the best solution from the population as the final result.

Algorithm 1: Agent-based population learning algorithm.

quality. A number of emerging technologies have been proposed for processing huge datasets by employing the multiagent systems. For example, the agent-based techniques can help solve the information overload problems [37]. Furthermore, agent-based applications can be of help in evaluating the quality of big data [38].

In [35], as well as in the following papers of the authors (see, for example, [39, 40]), it has been shown that the agent-based approach can help in solving difficult optimization problems. It is well known that data reduction belongs to the class of the combinatorial optimization problems and as such is computationally difficult. Hence, to solve the problem, some metaheuristics or other approximate algorithms are required. Numerous approaches to data reduction through instance selection have been based on using genetic or evolutionary algorithms (see, for example, [15, 41–43]).

A brief review of different approaches for instance selection can be found [4]. A broad review of the evolutionary algorithm applications to feature selection is available in [44].

This paper deals with the implementation of the agent-based population learning algorithm (PLA) to data reduction. The agent-based population learning algorithm has been proposed in [39] and belongs to the family of metaheuristics.

The PLA has been already used for solving problems of learning from data [35]. In [14], the stacking ensemble approach has been proposed for the purpose of improving the quality of the agent-based data reduction algorithm. In [11], the implementation has been extended using the rotation-based techniques. In the mentioned paper, the goal was to find the effective classification tool, which uses data reduction and which guarantees the maximization of the classification quality criterion.

The agent-based population learning algorithm is based on the A-Team architecture. The A-Team concept was originally introduced in [45]. It was motivated by several approaches like blackboard systems and evolutionary algorithms, which have proven to be able to successfully solve some difficult combinatorial optimization problems.

The functionality of the algorithm based on the implementation of the agent-based population learning approach can be defined as the organized and incremental search for the best solution. The agent-based population learning algorithm involves a specialized team of agents working asynchronously and in parallel, executing various improvement procedures with a view to solving the problem at hand. Agents working in the A-Team achieve an implicit cooperation by sharing the population of solutions to the problem to be solved. A-Team can be also defined as a set of agents and a set of memories, forming a network in which every agent remains in a closed loop. Agents cooperate to construct, find, and improve solutions which are read from the shared common memory. More information on the PLA and more details on the implementation of A-Teams can be found in [39]. The pseudocode of the agent-based population learning approach is shown as Algorithm 1.

3. The Proposed Approach to Learning from Big Datasets

3.1. Problem Formulation. It is well known that data reduction belongs to the class of the combinatorial optimization problems and as such is computationally difficult. Hence, to solve the problem, some metaheuristics or other approximate algorithms are required. Numerous approaches to data reduction through instance selection have been based on using genetic or evolutionary algorithms (see, for example, [15, 42, 43]).

In this paper, to enable dealing with huge datasets and to make the learning process more effective, it has been decided to apply the dual data reduction, that is, a reduction in the feature and instance spaces. It has been assumed that the resulting classifier will perform better either in terms of the computational resources required or in terms of classification accuracy or in respect to both criteria. Formally, dual data reduction can be viewed as searching for the dataset S which is the subset of the set D and $|S| < |D|$ (possibly $S = S_{\text{opt}}$), where each data instance belonging to S is represented by the set of original or transformed features A' with $|A'| < |A|$.

The proposed approach is based on the integration of the data reduction and learning stages with a view to improving the final classifier performance. Such an integration allows introducing some adaptation mechanisms into the learning process. The idea has been described in a more detailed manner in [35]. Such integrated learning has proven effective for assuring the required diversification among prototypes using the stacking technique [13]. A general model of the integrated learning is shown in Figure 1.

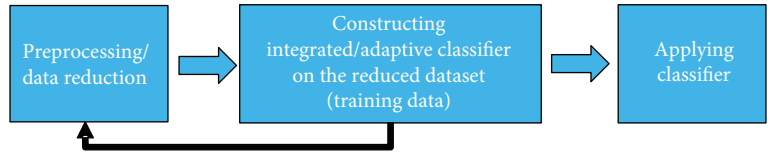

Figure 1: Integrated and adaptive learning from examples with data reduction.

Integrated and adaptive learning uses the positive feedback whereby more effective data reduction leads to a higher learning accuracy, and in return, higher learning accuracy results in even more effective data reduction.

Assume that the goal of learning from examples is to find a hypothesis h. The learner used to produce h requires the setting of some parameters decisive from the point of view of its performance. Let parameters g describe the way the training set should be transformed before training. Thus, it can be said that the goal of learning from examples is to find a hypothesis $h = L(D, g)$, where parameters g affect the learning process and influence the performance measure f. In such case, the learning task takes the following form:

$$h = \arg \max_{h \in H, g \in G} f(h = L(D, g)), \tag{2}$$

where G is the parameter space.

3.2. The Proposed Approach. In the proposed approach, it has been assumed that the learner is induced from prototypes. Prototypes, also referred to as reference instances, are represented by instances from the original dataset which have been selected in the evolutionary process. Before the selection process is activated, instances from the original dataset are grouped into clusters, and each cluster has its own reference instances in the final compact representation of the original dataset. In fact, each cluster has exactly one representative (reference instance) in the final dataset.

The above assumptions make a selection of the clustering algorithm crucial to the effectiveness of the resulting learner. We use two such algorithms—clustering guided by the similarity coefficient (SC) and the kernel-based C-means clustering algorithm (KFCM).

Similarity coefficient–based clustering was proposed in [40]. The algorithm assumes that for each instance from the original dataset, a similarity coefficient is calculated. Instances with identical coefficient are grouped into a cluster. The number of clusters is determined by the number of different similarity coefficients among the original dataset instances.

The second clustering algorithm—KFCM—was proposed to deal with problems caused by the noise and sensitivity to outliers characterizing the classic fuzzy C-means clustering algorithm. KFCM transforms input data into a higher dimensional kernel space through a nonlinear mapping [46]. The procedure has been already successfully used for the prototype selection [14].

To further increase chances for achieving a satisfactory performance of the learner induced over the reduced dataset, it has been decided to use the stacked generalization method using stratified sampling with replacement.

To improve performance and generalization ability of the prototype-based machine learning classification, it was decided to use the stacking technique. The implementation of the stacking technique in the discussed approach means that the process of classification with data reduction is carried out within the procedure that at first creates q different subsets of the training data using stratified sampling with replacement. All subsets are generated assuring relative proportion of the different classes as in the original dataset. However, to assure the required diversity, at first, q-1 training sets are split into the independent subsets with different feature subsets. Next, using q-1 subsets of the training sets, the process of the feature space modification is run.

Another diversifying factor is using the rotation technique or, alternatively, selecting features applying the population learning algorithm. In this paper, the first method is named as deterministic, while the second one as nondeterministic. In the case of the deterministic variant of the approach, based on rotation, two feature extraction techniques including principal component analysis (PCA) or independent component analysis (ICA) have been proposed.

After the above steps have been carried out, the learner is induced from the reduced (final) dataset transformed and diversified through applying stacking and rotation procedures. The process is executed by the set of agents cooperating and acting within the agent-based population learning algorithm. After the clusters have been produced followed by generation of the diversified subsets of the training data through stacking and rotation, potential solutions, forming their initial population, are generated through randomly selecting exactly one single instance from each of the considered clusters. Thus, a potential solution is represented by the set of prototypes, i.e., by the compact representations of the original dataset. A feasible solution to the data reduction problem is encoded as a string consisting of numbers of the selected reference vectors.

Selection procedure of the representation of instances through population-based search is carried out by the team of optimizing agents. Each agent is an implementation of the local search procedure and operates on individuals. The instance selection is carried out for each cluster, and removal of the remaining instances constitutes the basic step of the instance selection process. In case of feature selection, the potential solutions are improved by removing or adding an attribute to the solution that constitutes a basic step of the feature selection process. More precisely, the implemented improvement procedures include local search with the tabu list for instance selection, simple local search for instance selection, local search with the tabu list for feature selection,

Input: Dataset D with the feature set A; number of iterations q (i.e. the number of stacking folds); natural number T (defined by the user); *option* – the Boolean parameter determining the type of the transformation in the feature space (deterministic or nondeterministic)
Output: $h_{it(i:i=1,\ldots,q;t:t=1,\ldots T)}$ – set of the base classifiers
Begin
Allocate randomly instances from D into q disjoint subsets $D_1, \ldots, D_q$.
For $i = 1$ **to** q **do**
Let $D_i^{'} = D - D_i$
Partition randomly the feature set A into T subsets $\{A_{it}: t \leq T\}$ obtaining subsets D'_{it}, each with the identical number of features, smaller than the number of features in the original dataset.
For $t = 1$ **to** T **do**
Generate training set D'_{it} with features A_{it}, through bootstrapping with the size of 75% of the original dataset.
If *option* **then**
Run **PCA** or **ICA** over the transformed D'_{it} and produce new training datasets D''_{it}, with features A'_{it}, using the axis rotation;
Else
Run the **PLA** for feature selection on D'_{it} and produce new training datasets D''_{it} described on the set A'_{it}.
End If
Partition D'_{it} into clusters using the **KFCM** procedure or **SC** procedure.
Run **PLA** for the prototype selection obtaining $S'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ (i.e. subsets of the selected prototypes).
Induce base classifier h_{it} based on $S'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ using D_i with features A'_{it}, as the testing set.
End for
End for
Return $h_{it}, \ldots, h_{qT}$.
End.

Algorithm 2: Stacked generalization with rotation.

and simple local search for instance and feature selections. The detailed description and the background of these procedures can be found in [35].

To sum up, the optimizing agent task is to search for a better solution upon receiving a current one. To perform such search, each optimizing agent is equipped with some heuristic or local search algorithm which is activated immediately after the solution to be improved has been received. In case the agent is not able to find a better solution, the current one is returned. Otherwise, an improved solution is returned to the common memory. Quality of solutions also referred to as their fitness is evaluated through estimating the performance of the base classifier under evaluation. This performance is measured in terms of the classification accuracy provided the learner has been induced using instances and features of the reduced dataset.

From the implementation point of view, the above-described process of searching for the best results using the agent-based population learning algorithm is carried out in parallel for q independent data reduction problems. In each stream of such search, a different dataset is processed, and the independent set of the prototypes is selected.

The process of searching for solutions is iterative with q iterations. In each iteration, the reference instances are selected, and the respective decision tree is induced. Such tree plays the role of the base classifier. Its evaluation is carried out using the subset of instances which, in the current iteration, has been removed from the original dataset with a view to serving as the temporary testing set. The procedure produces a number of heterogeneous base classifiers forming an ensemble. The final decision as to the unknown class label is taken at the upper level of the stacking scheme through a majority vote. Pseudocode of the proposed scheme for stacked generalization with rotation is shown as Algorithm 2.

The diversity of the obtained set of the base classifiers is assured by application of stacking and rotation methods resulting in varying the training sets at the learning stage. The majority voting paradigm leads to the final decision as to the class label of the considered instance. It is computed as shown in

$$h = \arg \max_{h_{it} \in H,\, g_{it} \in G} \sum_{i=1}^{q} \sum_{t=1}^{T} w_{it} f(h_{it} = L(D_{it}, g_{it})), \qquad (3)$$

where g_{it} are the reduced instances produced by stacking and rotation procedures for $D'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)} \subset D$, $h_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$ are output hypotheses induced from training sets $D'_{it(i:i=1,\ldots,q;t:t=1,\ldots,T)}$, respectively, and w_{it} represents weights of the base classifiers induced at respective stacking levels.

The framework of the considered approach is shown in a graphic form in Figure 2.

However, the proposed approach is based on decomposition and involves the stacking and rotation, which can be carried out at random or in a deterministic way in feature space; the complexity of the approach is the sum of

(i) complexity depending on the number of iteration, i.e., the number of stacking folds—q

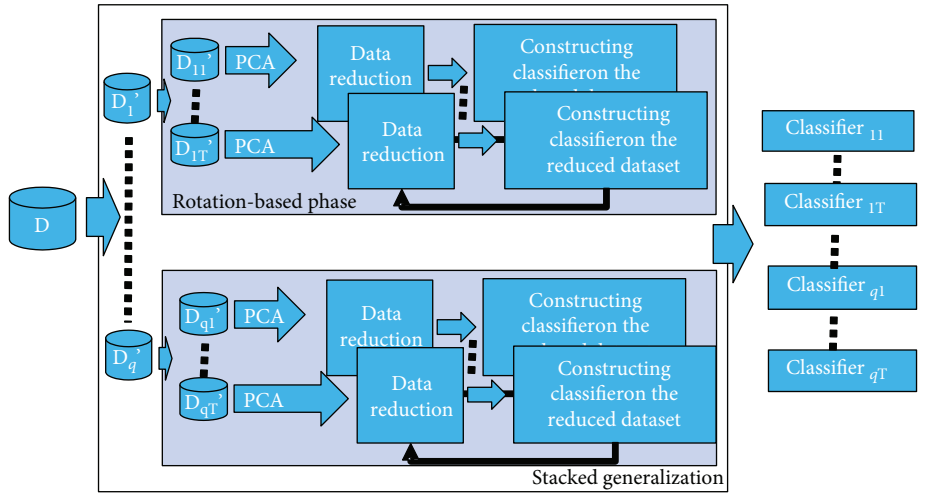

Figure 2: A framework of the proposed approach based on the PCA feature extraction.

(ii) the complexity of the selected rotation procedure, i.e.,

(a) the computational complexity of SC: $O(nN) + O(N \log N)$

(b) the computational complexity of KFCM: $O(tN^2n)$ [46]

(c) the computational complexity of the PLA, which especially depends on the computational complexity of the implemented improvement procedures,

where n denotes the number of features, N denotes the number of instances in the dataset, and t is the so-called number of eigenvalues

(iii) the complexity of the PLA implemented for the prototype selection—the complexity also depends on the computational complexity of the implemented improvement procedures

(iv) the complexity of the machine learning algorithm used to induce base classifier

4. Computational Experiment

4.1. Computational Experiment Setting. To validate the proposed approach, an extensive computational experiment has been planned and carried out. The experiment goals include searching for answers to the following questions:

(i) Can the instance selection be strengthened by using the rotation-based technique and the stacking?

(ii) How competitive is the proposed approach in comparison with the performance of the state-of-the-art classifiers?

(iii) Does the proposed approach produces, on average, better results than those produced by the earlier version of the algorithm introduced in [11], as well in [12–14, 35]?

In experiment, six following versions of the proposed algorithm have been considered:

(i) ABDRStEr$_{PCA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and PCA for feature extraction—introduced in [11]

(ii) ABDRkfStEr$_{PCA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and PCA for feature extraction—introduced in [11]

(iii) ABDRStEr$_{ICA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and ICA for feature extraction—a new version of the algorithm

(iv) ABDRkfStEr$_{ICA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and ICA for feature extraction—a new version of the algorithm

(v) ABDRStEr$_{PLA}$: agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning and PLA for feature extraction—a new version of the algorithm

(vi) ABDRkfStEr$_{PLA}$: agent-based data reduction based on the KFCM with stacking rotation ensemble learning and PLA for feature extraction—a new version of the algorithm

Among other algorithms proposed by the authors and compared in this paper are

(i) ABInDRkfStE: agent-based integrated data reduction based on the KFCM with the stacking ensemble learning—introduced in [13]

Table 1: Characteristics of the datasets used in this paper.

Dataset	Source of data	Instances	Attributes	Classes	Best reported results classification accuracy
Heart	[15]	303	13	2	90.0% [15]
Diabetes	[15]	768	8	2	77.34% [15]
WBC	[15]	699	9	2	97.5% [15]
ACredit	[15]	690	15	2	86.9% [15]
GCredit	[15]	1000	20	2	77.47% [15]
Sonar	[15]	208	60	2	97.1% [15]
Shuttle	[15]	58,000	9	7	95.6% [42]
Connect-4	[16]	67,557	42	3	—
Magic	[16]	19,020	10	2	—
Census	[16]	142,521	41	3	—

(ii) ABInDRStE: agent-based integrated data reduction based on the similarity coefficient with the stacking ensemble learning—introduced in [13]

(iii) ABDRkfStE: agent-based data reduction based on the KFCM with stacking ensemble learning and without feature selection—introduced in [14]

(iv) ABDRStE: agent-based data reduction based on the similarity coefficient with stacking ensemble learning and without feature selection—introduced in [14]

(v) ABIS: agent-based instance selection—proposed in [35]

(vi) ABDRE: agent-based data reduction with ensemble with RM-RR (random move and replace randomly strategy)—proposed in [12]

(vii) ABDRE with RM-RW (random move and replaces first worst strategy): proposed in [12]

All proposed and all above-mentioned algorithms belong to the family of the integrated-based learning paradigm.

Computational experiment results produced by the proposed approach have been also compared with some other approaches based on different ensemble techniques (AdaBoost, bagging, and random subspace) proposed in [11]. In the experiment, several benchmark datasets from the UCI and KEEL repositories [15, 16] have been used (for details see Table 1). The criterion for fitness evaluation has been the classification accuracy (Acc.) understood as the correct classification ratio. The experiment involved several runs. The number of stacking folds has been set from 3 to 10, respectively. The number of bootstraps has been set to 4. For each considered dataset, the experiment plan has required 10 repetitions over the 10-cross-validation (10-C-V) scheme induced using C.45 or CART algorithms. Each set of the 10-C-V of runs has repeated for 50 times.

For experiment where searching for a solution has been carried out by A-Teams, the following A-Team parameters have been used: population size (40) and stopping criterion (100 iterations without an improvement or one minute of computations without such improvement). In the case of bagging and random subspace, the size of bags has been set to 50% of the original training set. The number of base models in ABDRE with RM-RR and ABDRE with RM-RW has been set to 40.

4.2. Experiment Results. Classification accuracies produced by the investigated approaches using all considered data sets are shown in Table 2. The results have been reported as averages over all runs of each algorithm and for each problem, respectively.

In general, in the case of the proposed approach, results shown refer to the number of stacking folds producing the best results. Among the proposed models, best performers, on the average, are approaches using the integrated learning paradigm and stacking (Algorithms 1–16). This conclusion is valid independent of the clustering procedure used. Only in one case, we notice better results obtained by AdaBoost (please see the result for Connect-4). It can be observed that the proposed algorithms are competitive in comparison to others, among them to the DROP4 algorithm. This observation answers positively the second question asked at the beginning of this section.

The experiment also confirms that the rotation technique can improve the quality of results (the rotation technique has been implemented within algorithms from 1 to 12). Although the algorithms based on the rotation assured the best results in four cases, we can conclude that the rotation can improve the learning based on instance selection with stacking. On the other side, observing all algorithms proposed by the authors, among them their earlier versions, we can conclude that the instance selection can be strengthened by using the rotation-based technique and the stacking, which answers positively the first question asked at the beginning of this section.

The aim of the paper was also to verify the benefits from the diversification by the rotation technique, and two deterministic methods have been used (i.e., PCA and ICA). Alternatively, selecting features applying the population

Table 2: Classification accuracy (%) and comparison of different classifiers.

#	Algorithm	Heart	Diabetes	WBC	ACredit	GCredit	Sonar	Shuttle	Connect-4	Magic	Census
1	$ABDRStEr_{PCA}$ (C4.5) [11]	92.4	79.15	**98.25**	91.31	**80.42**	90.02	98.15	47.95	72.5	**83.54**
2	$ABDRkfStEr_{PCA}$ (C4.5) [11]	92.78	80.12	97.04	92.61	79.03	89.54	98.65	48.04	72.05	82.67
3	$ABDRStEr_{ICA}$ (C4.5)	91.21	78.21	96.2	91.6	78.2	87.01	98.5	47.62	71.9	81.52
4	$ABDRkfStEr_{ICA}$ (C4.5)	90.91	80.3	97.04	90.48	77.16	86.5	98.2	46.7	70.56	80.3
5	$ABDRStEr_{PLA}$ (C4.5)	91.94	80.23	95.21	91.24	78.69	88.6	97.84	46.8	71.69	81.62
6	$ABDRkfStEr_{PLA}$ (C4.5)	92.05	79.21	94.47	92	80.1	88.14	98.63	47.56	72.23	82.6
7	$ABDRStEr_{PCA}$ (CART)	90.56	78.21	96.21	95.2	78.65	87.5	97.45	45.21	71.86	82.15
8	$ABDRkfStEr_{PCA}$ (CART)	91.52	79.05	97.06	**96.4**	79.8	86.9	97.63	46.63	70.03	82.42
9	$ABDRStEr_{ICA}$ (CART)	90.72	78	95.28	94.26	77.76	85.71	98.05	45.77	72	81.62
10	$ABDRkfStEr_{ICA}$ (CART)	91.32	79.11	95.98	92.42	76.58	86.65	97.17	46.34	70.69	80.45
11	$ABDRStEr_{PLA}$ (CART)	90.06	80.6	96.02	91.3	77.91	86	98.26	46.04	71.62	82.41
12	$ABDRkfStEr_{PLA}$ (CART)	91.3	79.33	95.72	91.61	77.08	85.59	98.48	46.9	72.5	81.98
13	ABInDRkfStE [13]	**93.01**	**80.71**	98.08	92.04	78.45	90.57	98.41	46.98	**72.59**	82.68
14	ABInDRStE [13]	92.87	79.84	98.13	91.89	80.24	**91.15**	98.73	47.23	71.84	82.05
15	ABDRkfStE [14]	90.45	75.15	96.91	90.78	77.41	80.42	**99.66**	46.07	71.6	81.57
16	ABDRStE [14]	92.12	79.12	96.91	91.45	80.21	85.63	98.75	46.14	71.08	81.07
17	ABDRE with RM-RR [12]	92.84	80.4	96.4	90.8	78.2	83.4	97.51	45.67	70.96	81.65
18	ABDRE with RM-RW [12]	90.84	78.07	97.6	89.45	76.28	81.75	97.74	44.6	69.84	80.45
19	ABIS [35]	91.21	76.54	97.44	90.72	77.7	83.65	95.48	45.02	70.02	81.69
20	AdaBoost	82.23	73.55	63.09	91.05	73.01	86.09	96.13	**54.16**	68.57	80.6
21	Bagging	79.69	76.37	95.77	85.87	74.19	76.2	95.27	44.68	70.69	80.04
22	Random subspace method	84.44	74.81	71.08	82.14	75.4	85.18	92.81	43.58	70.56	79.05
23	C 4.5	77.8	73	94.7	84.5	70.5	76.09	95.6	45.89	69.13	80.61
24	SVM	81.5	77	97.2	84.8	72.5	90.4	—	—	—	—
25	DROP4 [20]	80.9	72.4	96.28	84.78	—	82.81	—	—	—	—

learning algorithm (PLA) have been implemented. These diversification techniques have been implemented within algorithms from 1 to 12. Analyzing the experiment results, we can observe that in nine cases out of ten, the best results have been obtained by PCA. Only in one case, the best result has been obtained using the PLA. Comparing results obtained using PLA and ICA allows one to observe that in seven cases, the better results have been obtained by PLA. In three cases, the better results have been assured by ICA. Thus, it can be observed that the rotation technique based on the PCA performs better than using PLA and ICA, even if PLA outperforms ICA.

The performance of the proposed approach has been also evaluated with respect to the kind of method used for inducing the base classifier. The computational experiment results show that the C4.5 as a machine learning tool used for ensemble induction assured better generalization than algorithm CART.

The question of the performance of the proposed methods can be also formulated with respect to the kind of the clustering methods. As it has been mentioned before, the clustering algorithm can be crucial to the effectiveness of the resulting learner. In this case, the computational experiment results show that the most effective is clustering guided by the similarity coefficient (SC). SC has been six times more effective in comparison to the kernel-based C-means clustering algorithm. We also observe that the agent-based data reduction based on the similarity coefficient with stacking rotation ensemble learning has been more effective when the PCA for feature extraction was used.

To confirm and verify the obtained results, Friedman and Iman-Davenport's nonparametric ranking test has been carried out for comparison of the results. Results have been ranked, and the ranking of the results has been computed assigning to the best of them rank 1 and rank 23 to the worst one (the statistical analysis does not include results for SVM and DROP4). Figure 3 depicts average weights for each compared algorithm obtained by Friedman's test.

The tests have been carried out under the following hypotheses:

(i) H_0—null hypothesis: all of the 23 compared algorithms are statistically equally effective regardless of the kind of the problem

(ii) H_1—alternative hypothesis: not all algorithms are equally effective

Both analyses have been carried out at the significance level of 0.05. The respective value of the x^2 statistics for

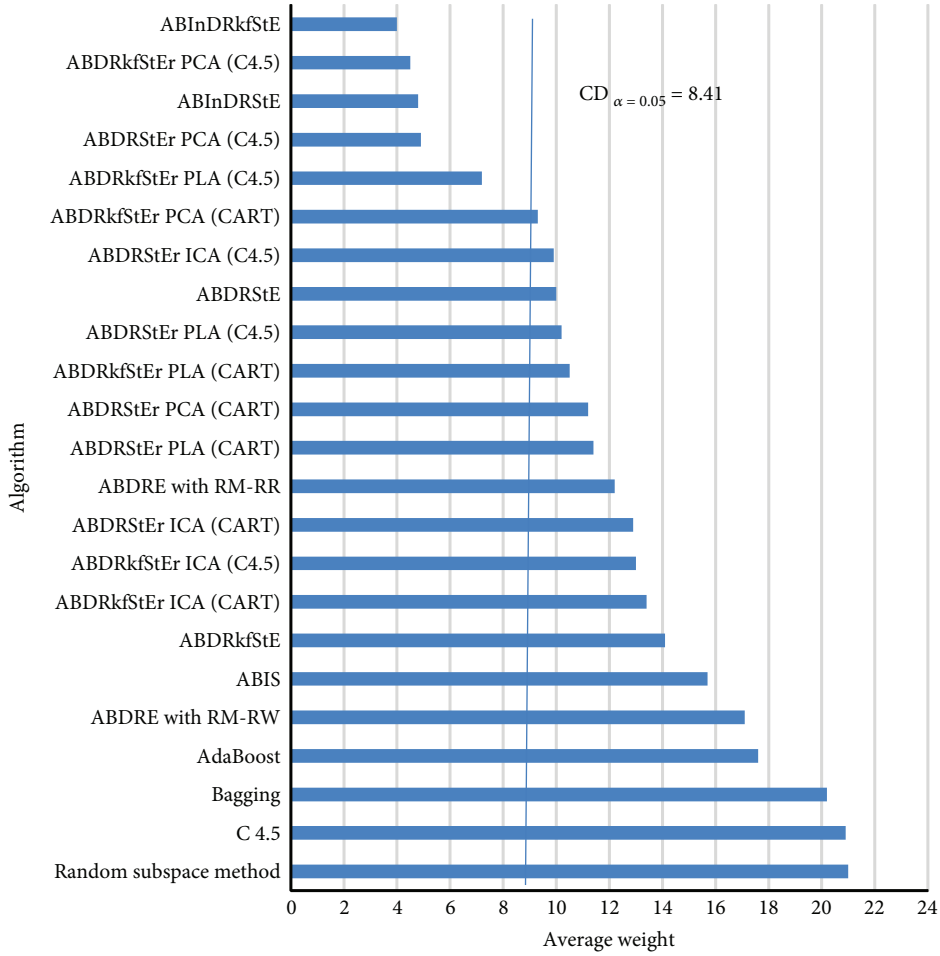

Figure 3: The average Friedman test weights and Bonferroni-Dunn's graphics corresponding to the obtained ranking.

Friedman's test with 23 algorithms and 10 instances of the considered problems is 124.8826087; the value of χ^2 of the distribution is equal to 33.92443847 for 22 degrees of freedom. The respective value F_F of Iman-Davenport's test is 9.763455; the critical value of $F(22.198)$ degrees of freedom is 1.59630281. For both tests, the p values are lower than the considered significance level $\alpha = 0.05$; thus, there are significant differences among the analyzed results, and the null hypothesis should be rejected. This means that not all algorithms are equally effective regardless of the kind of problem which instances are being solved.

A post hoc statistical analysis, based on Bonferroni-Dunn's test, to detect significant differences between the compared algorithms has been carried out. The critical difference (CD) of Bonferroni-Dunn's procedure is shown in Figure 3. The vertical cut line represents the threshold for the best performing algorithms. These bars which exceed the threshold are associated with algorithms displaying the worst performance with respect to the first five algorithms (ABInDRkfStE, $\text{ABDRkfStEr}_{\text{PCA}}$ (C4.5), ABInDRStE, $\text{ABDRStEr}_{\text{PCA}}$ (C4.5), and $\text{ABDRkfStEr}_{\text{PLA}}$ (C4.5)). These algorithms are better than the other versions with $\alpha = 0.05$.

To sum up the results of the statistical analysis, it can be concluded that the best results have been obtained

(i) by data reduction algorithms based on stacking and without rotation transformation in the feature space

(ii) by data reduction algorithms with stacking rotation ensemble learning and based on PCA for feature extraction independently on the cluster method used in the process of data reduction; however, when the KFCM has been used, the PLA was preferred

(iii) by data reduction based on integrated learning, which confirms our previous observation

The important factor of the research is that the proposed approach is based on decomposition by stacking, and the process of learning on the decomposition strategy is assured by the multiple agent system. It should be also underlined that the success of the learning process of the learning based on the PLA algorithm depends on the improvement procedures employed by the optimization agents.

5. Conclusions

The main scientific contribution of the paper is to propose an improvement to the core procedure of the proposed data reduction approach. The procedure integrates stacked generalization and rotation-based methods. The proposed algorithm allows for either deterministic or random transformations in the feature space. This feature was not available in the earlier algorithm proposed in [11]. It has been shown experimentally that the above option improves the performance of the procedure. The paper contributes also by proposing and evaluating a family of the hybrid classifiers based on data reduction, stacking, feature space rotation, and multiple agent environments. The proposed approach can be applied to mine huge datasets owing to quite radical data reduction mechanism and inherent parallelization typical for the multiple agent systems. It has been experimentally shown that merging stacking, rotation-based ensemble techniques, and data reduction with machine classification may bring the added value with respect to the accuracy of the classification process.

Future research will concentrate on searching for more effective local search procedures employed by the optimization agents. It is also envisaged to investigate different learners and different strategies with respect to the decision making within the classification ensemble. Finally, it would be also interesting to detect experimentally scaling up barriers for the proposed approaches.

Data Availability

The data used in this study are available at

(i) UCI Machine Learning Repository—http://archive.ics.uci.edu/ml/index.php

(ii) KEEL-dataset repository—http://sci2s.ugr.es/keel/datasets.php

Disclosure

The paper includes an extension of the research results presented earlier during the 2017 IEEE INISTA Conference. The earlier presented approach has been modified, new datasets have been used, and a broader analysis of the results has been carried out.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] N. García-Pedrajas and A. de Haro-García, "Scaling up data mining algorithms: review and taxonomy," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 71–87, 2012.

[2] M. H. U. Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, "Big data reduction methods: a survey," *Data Science and Engineering*, vol. 1, no. 4, pp. 265–284, 2016.

[3] X. Wang and Y. He, "Learning from uncertainty for big data: future analytical challenges and strategies," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 2, no. 2, pp. 26–31, 2016.

[4] S.-W. Kim and B. J. Oommen, "A brief taxonomy and ranking of creative prototype reduction schemes," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 232–244, 2003.

[5] T. Mitchell, *Machine Learning*, McGraw-Hill, New York, 1997.

[6] I. Triguero, M. Galar, H. Bustince, and F. Herrera, "A first attempt on global evolutionary unsampling for imbalanced big data," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2054–2061, San Sebastian, Spain, June 2017.

[7] S. L. Lohr, *Sampling: Design and Analysis*, Cengage. Learning, Boston, MA, USA, 2nd edition, 2009.

[8] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*, CRC Press/Francis Taylor, Boca Raton, FL, 2013.

[9] Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 551–577, 2017.

[10] N. García-Pedrajas and A. de Haro-García, "Boosting instance selection algorithms," *Knowledge-Based Systems*, vol. 67, pp. 342–360, 2014.

[11] I. Czarnowski and P. Jędrzejowicz, "Stacking and rotation-based technique for machine learning classification with data reduction," in *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 55–60, Gdynia, Poland.

[12] I. Czarnowski and P. Jędrzejowicz, "Agent-based data reduction using ensemble technique," in *Computational Collective Intelligence. Technologies and Applications*, C. Badica, N. T. Nguyen, and M. Brezovan, Eds., pp. 447–456, Springer, Berlin, Heidelberg, 2013.

[13] I. Czarnowski and P. Jędrzejowicz, "An approach to machine classification based on stacked generalization and instance selection," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4836–4484, Budapest, Hungary, October 2016.

[14] I. Czarnowski and P. Jędrzejowicz, "Learning from examples with data reduction and stacked generalization," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 2, pp. 1401–1411, 2017.

[15] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, 2007, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[16] J. Alcalá-Fdez, A. Fernández, J. Luengo et al., "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, 2011.

[17] A. A. Yıldırım, C. Özdoğan, and D. Watson, "Parallel data reduction techniques for big datasets," in *Big Data Management, Technologies, and Applications*, W.-C. Hu and N. Kaabouch, Eds., pp. 72–93, IGI Global, 2014.

[18] J. Derrac, N. Verbiest, S. García, C. Cornelis, and F. Herrera, "On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection," *Soft Computing*, vol. 17, no. 2, pp. 223–238, 2013.

[19] J. C. Bezdek and L. I. Kuncheva, "Nearest prototype classifier designs: an experimental study," *International Journal of Intelligence Systems*, vol. 16, no. 12, pp. 1445–1473, 2001.

[20] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithm," in *Machine Learning*, D. R. Wilson and T. R. Martinez, Eds., vol. 38, no. 3pp. 257–286, Kluwer Academic Publishers, 2000.

[21] Á. Arnaiz-González, J. F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Instance selection of linear complexity for big data," *Knowledge-Based Systems*, vol. 107, pp. 83–95, 2016.

[22] C. Liu, W. Wang, M. Wang, F. Lv, and M. Konan, "An efficient instance selection algorithm to reconstruct training set for support vector machine," *Knowledge-Based Systems*, vol. 116, pp. 58–73, 2017.

[23] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[24] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[25] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[26] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 271–289, 1999.

[27] D. B. Skalak, *Prototype Selection for Composite Neighbor Classifiers*, University Of Massachusetts Amherst, 1997, https://web.cs.umass.edu/publication/docs/1996/UM-CS-1996-089.pdf.

[28] J. A. Benediktsson, J. Chanussot, and M. Fauvel, "Multiple classifier systems in remote sensing: from basics to recent developments," *Lecture Notes in Computer Science*, vol. 4472, pp. 501–512, 2007.

[29] J. Xia, J. Chanussot, P. Du, and X. He, "Rotation-based ensemble classifiers for high-dimensional data," in *Fusion in Computer Vision*, B. Ionescu, J. Benois-Pineau, T. Piatrik, and G. Quénot, Eds., pp. 135–160, Springer, 2014.

[30] R. Blaser and P. Fryzlewicz, "Random rotation ensembles," *The Journal of Machine Learning Research*, vol. 2, pp. 1–15, 2015.

[31] J. Xia, *Multiple classifier systems for the classification of hyperspectral data, [Ph. D. Thesis]*, University de Grenoble, 2014.

[32] I. H. Witten and D. J. Merz, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2nd edition, 2005.

[33] I. Czarnowski and P. Jędrzejowicz, "Cluster-dependent rotation-based feature selection for the RBF networks initialization," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, Gdynia, Poland, June 2015.

[34] H. L. Zhang and H. C. Lau, "Agent-based problem-solving methods in big data environment," *Web Intelligence and Agent Systems: An International Journal*, vol. 12, pp. 343–345, 2014.

[35] I. Czarnowski, "Distributed learning with data reduction," in *Transactions on Computational Collective Intelligence IV*, pp. 3–121, Springer, 2011.

[36] F. D. Ahmed, A. N. Jaber, and M. B. A. Majid, "Agent-based big data analytics in retailing: a case study," in *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pp. 67–72, Kuantan, Malaysia, August 2015.

[37] A. Amato, B. Di Martino, and S. Venticinque, "Agent-based decision support for smart market using big data," *Lecture Notes in Computer Science*, vol. 8286, pp. 251–258, 2013.

[38] S.-H. Chen and R. Venkatachalam, "Agent-based modelling as a foundation for big data," *Journal of Economic Methodology*, vol. 24, no. 4, pp. 362–383, 2017.

[39] D. Barbucha, I. Czarnowski, P. Jędrzejowicz, E. Ratajczak-Ropel, and I. Wierzbowska, "e-JABAT—an implementation of the web-based A-Team," in *Intelligence Agents in the Evolution of Web and Applications. Studies in Computational Intelligence 167*, N. T. Nguyen and L. C. Jain, Eds., pp. 57–86, Springer, Berlin, Heidelberg, 2009.

[40] I. Czarnowski, "Cluster-based instance selection for machine classification," *Knowledge and Information Systems*, vol. 30, no. 1, pp. 113–133, 2012.

[41] J. R. Cano, F. Herrera, and M. Lozano, "On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining," *Applied Soft Computing*, vol. 6, no. 3, pp. 323–332, 2006.

[42] R. Sikora and O. H. Al-Laymoun, "A modified stacking ensemble machine learning algorithm using genetic algorithms," *Journal of International Technology and Information Management*, vol. 23, no. 1, pp. 1–11, 2014.

[43] S. H. Lee and J. S. Lim, "Evolutionary instance selection algorithm based on Takagi-Sugeno fuzzy model," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, pp. 1307–1312, 2014.

[44] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[45] S. Talukdar, L. Baerentzen, A. Gove, and P. de Souza, "Asynchronous teams: cooperation schemes for autonomous," *Computer-Based Agents, Technical Report EDRC 18–59-96*, Carnegie Mellon University, Pittsburgh, 1996.

[46] S. Zhou and J. Q. Gan, "Mercer kernel fuzzy c-means algorithm and prototypes of clusters," *Proceedings of the International Conference on Data Engineering and Automated Learning*, , pp. 613–618, Lecture Notes in Computer Science, 2004.