Physica A 553 (2020) 124219

Contents lists available at ScienceDirect

Physica A

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

A mixed solution-based high agreement filtering method for class noise detection in binary classification

Maryam Samami [a], Ebrahim Akbari [a], Moloud Abdar [b,c,*], Pawel Plawiak [d,e], Hossein Nematzadeh [a], Mohammad Ehsan Basiri [f], Vladimir Makarenkov [c]

[a] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
[b] Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Australia
[c] Department of Computer Science, University of Quebec in Montreal, Montreal (QC), Canada
[d] Department of Information and Communications Technology, Faculty of Computer Science and Telecommunications, Cracow University of Technology, Warszawska 24 st., F-3, Krakow 31-155, Poland
[e] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
[f] Department of Computer Engineering, Shahrekord University, Shahrekord, Iran

ARTICLE INFO

Article history:
Received 12 June 2019
Received in revised form 27 September 2019
Available online 31 January 2020

Keywords:
Data mining
High agreement voting filtering
Classification
Removing
Relabeling
Class noise detection

ABSTRACT

Classification of noisy data has been a longstanding topic in data mining and machine learning. Many scientists have proposed effective methods to detect and eliminate such data in diverse real-world datasets. In this paper, we deal with mislabeled instances in supervised learning, including majority voting filtering and consensus voting filtering. The majority voting procedure usually incorrectly identifies many correct instances as noisy, whereas the consensus voting procedure is not able to detect at all many noisy instances. Our new method minimizes the majority and consensus filtering weaknesses by providing a novel class noise detection strategy, namely a high agreement voting filtering with mixed strategy, which proceeds by removing strong and semi-strong noisy records from the dataset as well as by relabeling weak noisy data. The proposed method, designed for binary classification problems, outperforms the high agreement voting filtering procedure. Extensive experiments conducted with 16 real datasets, using four noise filtering methods with two levels of class noise (10% and 15%), prove the superiority of the proposed methodology.

1. Introduction

During multiple data collection processes some errors, having negative effects on information which can be derived from data, may occur. It has been observed that existing errors in datasets would decrease the performance of machine learning models. Thus, to improve the performance of machine learning models, data preprocessing steps should be carried out. It has also been noticed in the related literature that one of the main issues in data mining is dealing with noisy data to increase the accuracy of prediction [1].

Noisy data appear in three forms: instances with wrong labels are known as class noise, instances that have wrong attribute values or abnormal attribute values are known as attribute noise [2], and the combination of both [3–7]. Mislabeling of training data occurs because of insufficient information about instances [4]. Class noise has more effects

* Corresponding author.
E-mail address: m.abdar1987@gmail.com (M. Abdar).

https://doi.org/10.1016/j.physa.2020.124219
0378-4371/

on decreasing performance accuracy of data mining methods rather than attribute noise. This happens because of the two following facts: (1) for each sample in the dataset, there are multiple features, but there is just one label, (2) each attribute may have a bold or faint impact on learning methods, but the class label always has a strong impact on the learning process. Only in one case feature noise is more harmful than label noise. This occurs when there are a lot of polluted attributes in the data [5,8]. The existence of class noise implies that the wrong labeling of training samples has disruptive effects on the dataset [9].

Many papers describe the application of noise detection classification methods in health science and bioinformatics [4, 10]. To do so, two common approaches have been used. First of them concentrates on new noise-resistant classification strategies [11], while the second focuses on data filtering during the preprocessing phase [10]. In this paper, we describe a new methodology based on data preprocessing. Hence, we try to fill the gap between two ensemble voting filtering methods, namely Majority Voting Filtering (MF) method and Consensus Voting Filtering (CF) method. The performance of the proposed method was compared to MF and CF, which were recommended for removing misclassified samples in many studies [9,12–14].

Sluban et al. [3] studied the relation between the performance of various noise detection ensembles and the diversity of heterogeneous ensembles related to classification methods. To do so, they applied the MF and CF methods to identify class noise data. The application of the MF method for detecting class noise in ensemble algorithms yields high recall values, but low precision values, and thus leads to misclassifying many regular instances as noisy. In contrast, the application of the CF method usually yields high precision, but low recall, and thus results in an inability to detect many noisy instances. The use of consensus voting ensemble algorithms is appropriate when the most significant noisy instances should be detected and removed. Experimental observations show that more diversity in CF leads to achieving higher precision of noise labeling detection, but fewer diversity results in higher recall and F-score values [3].

To compensate for the weaknesses of the MF and CF methods in detecting class noise, we propose a new hybrid method, namely High Agreement Voting Filtering (HAVF) using mixed strategy. The HAVF method using mixed strategy removes strong and semi-strong noisy samples and relabels weak noisy instances. The proposed noise detection method includes three main steps. The first step consists in detecting noise instances. At this phase, the detected noisy instances are split into three groups, including weak noise, semi-strong noise, and strong noise. In the second step, the strong and semi-strong noisy instances are removed from the dataset (filtering step). Finally, the weak noisy instances are relabeled (correction step). In our opinion, applying HAVF with mixed strategy (removing and relabeling) could raise the classification accuracy, compared to the MF and CF methods. The main contributions of this work are summarized below:

- The proposed method, i.e., HAVF using mixed strategy, reduces the existing gap between the MF and CF methods [3]. The MF method incorrectly identifies many correct samples as noisy samples [15], what makes it very inaccurate, specifically when it comes to datasets with low levels of noise [16,14,9]. In contrast, the CF method is not able to detect many noisy samples. The CF method is too conservative in detecting mislabeled samples, which makes it unsuitable for detecting noisy data in datasets containing low amounts of mislabeled samples [9,15,16]. Thus, we propose a new hybrid algorithm to minimize the above-mentioned weaknesses.
- The novelty of the proposed method is that it not only helps to detect noisy data that would not be identified as class noise by CF, but also decreases the amount of removed regular instances that would be misidentified as noisy samples by MF, especially when the noise level is low [14]. While the use of the mixed strategy helps the proposed method to be more resistant in removing correct samples, compared to MF [14], it is not as conservative as CF. It is worth noting, the proposed method is designed for binary classification problems.
- This study also opens doors for applying HAVF using mixed strategy to manage class noise, especially for data with low level of noise. Applying MF on low level noisy datasets usually results in discarding many valid samples and decreasing the performance of machine learning algorithms [14]. Thus, we propose to modify the class noise samples by relabeling them instead of removing them incorrectly as noisy data.

The rest of the paper is organized as follows. Section 2 briefly discusses related work in the field. In Section 3, we describe the proposed method in detail. Our experimental validation and the obtained results are presented in Section 4. In Section 5, we discuss the advantages and disadvantages of the new method. Finally, we conclude the study in Section 6.

2. Related work

In this section, we first review some existing related studies. The preliminaries are discussed in Section 2.2.

2.1. Literature review

Lavrac et al. [17] presented a simple compression measure and its ability to detect noisy samples. The proposed technique includes two steps: step one consists of detecting and removing potential noisy samples from the training set (i.e., called the saturation filter step) and step two is the formatting hypothesis. The application of a simple compression measure leads to the detection of noisy samples without constructing a hypothesis from the training set. Potential noisy samples may include outliers which should be added as exceptions to the generated rule after the formatting hypothesis.

Although the class noise handling mechanisms such as Inductive Learning by Logic Minimization (ILLM) integrated to saturation filter, i.e., C4.5, k-nearest neighbors (KNN) and CN2, outperform the proposed saturation filter method in preprocessing, the superiority of the method described in [17] is related to a clear detection of error. One of the advantages of the method by Lavrac et al. is shown on the problem of early diagnosis of rheumatic diseases. On the cleaned datasets, obtained by applying the multiclass saturation, the CN2 learning algorithm without noise handling mechanism yielded a better accuracy, of 0.453, compared to CN2 with noise handling mechanism, 0.429 and 0.45, on the original data [17]. The removal of noisy samples led to better relative information scores. It is worth noting that the application of the multiclass saturation filter in tandem with the C4.5 pruning provided the accuracy of 0.746 which was better than the accuracy values of 0.728 and 0.744, which were obtained respectively by using the proposed filtering method without applying pruning and by using pruning without using saturation filtering. Combining KNN with the proposed filtering method yielded the accuracy of 0.748, outperforming the saturation method used in solo (i.e. without being combined with KNN filtering).

Sáez et al. [18] have analyzed the performance of three classifiers (C4.5, Support vector machine (SVM), and nearest neighbor (NN) rule), with and without noise filtering, considering six different noise filters and 12 medical datasets with different noise levels. SVM classifier is commonly able to provide a good performance when no noise filtering is applied. Applying noise filtering methods is necessary when the noise level is high. Good performance has been usually provided the following filters: Ensemble Filtering (EF), Iterative-Partitioning Filter (IPF), and NCNE (Nearest Centroid Neighborhood Edition). For instance, the application of the IPF, EF, and NCNE filtering methods to the Breast cancer, Parkinson, and Statlog Heart datasets, with 10% of added noise data led to the accuracy values of 0.8052, 0.8615, and 0.8037, respectively.

Guan et al. [19] presented a survey paper which focuses on mislabeled data as well as on the related data detection techniques, namely Local learning-based, Ensemble learning-based, and Single learning-based methods. These authors highlighted the fact that different types of methods have their own advantages and disadvantages, and that there is no best method overall since the obtained results depend on the characteristics of datasets and the related domain subject. Local learning-based methods, edited the nearest neighbor, Nearest centroid neighbor edition, and Relative neighborhood graph edition, have some advantages such as being easy to understand and implement. However, these methods assume that the samples located close to each other always belong to the same class. Ensemble learning-based methods is another type of mislabeled data processing methods, which includes MF and CF techniques. Although these methods rely on the ensemble approach to detect mislabeled data, and thus provide a better accuracy than the competing methods, they also have a high time complexity as they proceed by training multiple classification algorithms. Finally, Single learning-based methods are presented by two main approaches, namely Decision trees (DT) and Neural networks (NN).

The performance of a filtering method on several datasets depends on the selected classification procedure. As a result, we cannot consider a specific filtering technique as the best one for all datasets. Luís P.F. Garcia et al. [5] described a meta-learning system to support the recommendation for choosing the most promising machine learning algorithm(s). The proposed system is able to predict the performance related to each noise filter used to detect noisy data. Meta Learning approach (MTL) recommends the selection of an appropriate algorithm for noise detection regarding the attributes of the dataset. A meta-base should be created before applying MTL. From each dataset associated with a meta-example, a collection of characteristics, namely meta-features are extracted [5,20,21]. By applying DEF (Dynamic Ensemble Filter) and HARF (High Agreement Random Forest Filter) on the Ionosphere dataset, the F-measure value was 0.74 obtained. Applying HARF on the Australian credit approval dataset resulted in the best F-measure value equal to 0.59. Using CVCF (Cross-validated Committees Filter), DEF and PruneSF for Blood transfusion dataset provided the best F-measure value of, 0.44, among other filters. To detect and filter the random noise of the Parkinson dataset, AENN (All-k-NN) acts as the best Filter with the F-measure value of 0.74. For the Statlog-heart dataset, the F-measure value of 0.52 was obtained using DEF [5].

Nicholson et al. [22] proposed two novel algorithms for correcting class noise, namely Self-Training Correction (STC) and Cluster-based Correction (CC). Self-Training Correction uses self-training for relabeling noise labels. Cluster-based Correction is able to group instances to trace ground-truth labels. These authors also presented a method based on consensus called polishing labels (PL) to change the value of attributes and labels. The experimental results showed that only CC allows one to improves the values of all the three metrics: AUC, model quality and label quality. STC was the best method in improving model quality, AUC in binary and multi-class datasets, and in improving label quality in binary class datasets, whereas STC was the worst in improving label quality in multi-class datasets. It is worth noting that PL was the worst in improving all scenarios. Despite the advantages mentioned above, the main weakness of these methods was as follows: when the level of noisy label instances was lower than 10%, all the three-class noise correction methods were unable to improve the quality of labels in more than a half of the considered experimental datasets. The inference algorithms, namely KOS (Karger, Oh, & Shah) and Daiwid Skene (DS), were more effective consensus methods than the MF technique. Overall, CC provided the best classification results, PL performed pretty satisfactorily, and STC provided very inconsistent results. Average results obtained by applying PL, STC, and CC on all datasets in terms of the accuracy were 0.85, 0.90, and 0.90, while the AUC values were 0.60, 0.79, and 0.79, respectively.

Sluban et al. [10] presented methods for noise ranking based on ensemble algorithms. These authors tried to assess the performance of these algorithms as well as their publicly available web implementations. The first method, Noise Rank ensemble-based procedure, detects and ranks the identified noise using a specific toolkit. The second method, applied for visual performance evaluation (VIPER) of noise detection algorithms, compares the performances of noise detection

algorithms via an intuitively understandable visualization of the obtained results. These method allows one create balance among the precision and recall, applying the F-isoline and the ε-proximity assessment methodology. Sluban et al. showed that ensemble filters and their HARF noise detection algorithms outperformed individual filters in terms of precision, while Ens10 and HARF-80 were the most accurate algorithms for the TTT and KRKP datasets. However, HARF-80 was the priciest algorithm in terms of the running time, for CHD and NAKE datasets. Ens1, Ens2, Ens3, and HARF-70 showed the best performance in terms of the noise recall. The HARF-80 algorithm was able to generate the best $F_{0.5}$-score results ($F_{0.5}$-score values were close to 0.94) for all datasets, except the TTT dataset for which Ens7 performed better.

Zhang et al. [23] proposed an Aggregate Ensemble (AE) learning framework in order to create a robust learning system able to tolerate noisy samples in drifting data streams. Although most available methods propose preprocessing techniques to clean noisy samples in data stream environments, these techniques are hard to apply in drifting data streams because of the difficulty in differentiating noise from samples. AE learning framework can create much more accurate prediction models from a noisy concept in drifting data streams. AE provided the average accuracy of 0.94 in the experiments presented in [23].

Sáez et al. [24] proposed a new method, Iterative Noise Filter based on the Fusion of Classifiers (INFFC), for detecting and filtering noise. This method is based on merging the output of several multiple classifiers in order to improve their accuracy after the filtering process. The proposed method uses an iterative noise filtering algorithm that prevents from considering the identified noisy instances at each new iteration of the filtering process. INFFC provided the average accuracy of 0.812 by using the C4.5 classifier on 25 datasets corrupted by 10% of class noise. This was the best result among other applied filtering methods and classifiers. Moreover, Sáez et al. [24] also proposed a noisy score strategy to specify the amount of removed noisy instances at each iteration.

Sabzevari et al. [14] studied bootstrap ensemble algorithms applying them for identifying noisy class noise instances. These authors explained the ability of subsampling to make the ensembles more robust to the label noise. The proposed approach used both filtering and cleaning to tackle noisy data. The average test error rates obtained by using random forest with filtering (FL_rf) and random forest with cleaning (Cl_rf) for 10% of noise were 0.143 and 0.155, respectively [14].

2.2. Preliminaries

To classify binary datasets artificially corrupted by class noise samples, we assume a feature space $x \in R^d$ and a label space $y = \{-1, +1\}$. In this study, we consider $(X, Y, \tilde{Y}) \in x \times y \times y$, where X represents the observations, Y indicates the uncorrupted and unseen labels, and $\tilde{Y}$ denotes the noisy and observed labels. Aiming to make a classifier $f : x \rightarrow y$which is capable to predict Y (the class for each sample x) from X. In this study, the random classification noise (RCN) $\rho_1(X) = \rho_{-1}(X) = \rho$ was applied to create noise data, changing the label of each sample with the probability $\rho \in [0, 1)$. The employed noise rate is denoted as follows [25]:

$$\rho_Y(X) = P(\tilde{Y} = -Y|X, Y). \tag{1}$$

Heterogeneous ensembles considered in our study employ several individual learning classifiers for predicting mislabeled samples [3,26]. In ensemble voting methods, the base classifiers are combined using different combination rules for creating the final ensemble classifier which detects class noisy samples [3]. Predictions of algorithms that give us the class of test samples (like 'mislabeled' and 'correct labeled') can be combined using different voting methods such as MF, CF, the proposed method, and HAVF using removing strategy. The reason for using the ensemble approach instead of single learning-based methods is the ability of multiple classification algorithms to learn from each other in a complementary manner [19]. Another merit of ensemble classifiers over single classifiers is their ability to modify the weaknesses of individual ensemble members in order to increase the overall ensemble performance [3,27]. However, the lack of data witnessed in several fields provides us with improper distributed data. Applying ensemble methods will allow one to minimize wrong decision making when it comes to choosing the base learning algorithms [28].

2.2.1. Noise detection methods

A noise detection ensemble E of size L is created from a set of multiple individual algorithms $\{A_1, \ldots, A_L\}$ applied for noise detection. In this study, four different pre-processing methods for identifying class noise have been used: (1) MF method that removes class noise, (2) HAVF method that removes strong and semi-strong noisy data and also relabels weak noisy data, namely HAVF using mixed strategy, (3) HAVF method that removes strong and semi-strong class noisy data, namely HAVF using removing strategy, and (4) CF method that removes class noise. The MF and CF methods are described below. Our novel method, HAVF, which removes strong and semi-strong class noises and relabels weak class noises is presented in the Section 3.2.

2.2.1.1. Majority filtering (MF) method. If more than a half of the base classifiers A_i from E identify an instance x as noisy, then the ensemble declares it as noisy. MF identifies an instance as mislabeled provided that more than a half of all individual classifiers are unable to classify it with a correct label [29]. The weakness of this method is detecting correct instances as noise, and as a result the elimination of correct samples from the data [3]. According to the MF method presented in Algorithm 1 [30], the instances of E_i for which the majority of the base level algorithms misidentify the correct

label are added to A as potentially noisy samples. We consider Eq. (2) to detect mislabeled samples of X by ensemble E using MF. We assume that the output of the function δ for noisy samples is 1, otherwise it is 0.

$$MF = \sum_{i=1}^{l} \delta\left(A_i\left(\mathbf{X}\right)\right) > L/2 \tag{2}$$

Input: E (training set)

Parameter: n (number of samples), l (number of learning algorithms),

$A_1, A_2, ..., A_l$ (l types of learning algorithms)

Output: A (detected noisy subset of E)

(1) form n disjoint almost equally sized subsets of E_i, where $\cup_i E_i = E$
(2) $A \leftarrow \emptyset$
(3) for $i = 1, \ldots, n$ do
(4) form $E_t \leftarrow E/E_i$
(5) for $j = 1, \ldots, l$ do
(6) induce H_j based on samples in E_t and A_j
(7) end for
(8) for every $e \in E_i$ do
(9) $Errorcounter \leftarrow 0$
(10) for $j = 1, \ldots, l$ do
(11) if H_j incorrectly classifies e
(12) then $ErrorCounter \leftarrow ErrorCounter + 1$
(13) end for
(14) if $ErrorCounter > (l/2)$ then $A \leftarrow A \cup \{e\}$
(15) end for
(16) end for

Algorithm. 1. The Majority Filtering (MF) algorithm.

2.2.1.2. Consensus filtering (CF) method. CF identifies noisy instances only if all the base classifiers classify the instance incorrectly. CF is more conservative than MF due to the stricter rules applied for noise detection. Such a stricter approach keeps more undetected noisy instances in the dataset [29]. The CF method is represented in Algorithm 2 [30]. We consider Eq. (3) to detect mislabeled samples of X by ensemble E using CF.

$$CF = \sum_{i=1}^{l} \delta\left(A_i\left(\mathbf{X}\right)\right) > L \tag{3}$$

3. The proposed method

In this section, we first present the classifiers used in the framework of the proposed method. Then, the description of our new method, i.e., HAVF using removing strategy is provided.

3.1. Classifying ensemble

Five base classifiers, namely KNN, SVM, Decision Tree (DT), Random Forest (RF) and Naive Bayes (NB), are used in all four ensemble-based noise detection methods tested in our study. The advantage of ensemble classifiers over single

classifiers is their ability to correct the errors of individual ensemble members, and thus improve the overall ensemble classifier performance [3]. Each base classifier has its own performance for each dataset considered noisy dataset. This leads to a wide range of different predictions and rather unstable predictions. Here, we briefly discuss the strong and weak points of each individual classifier used in our study. KNN, for instance, is much more sensitive to noisy samples as it demands clean class borders to make decisions. DT is more robust against noisy data. DT is usually capable of tolerating low quantities of noisy data. Moreover, KNN is an expensive method in terms of computation as it requires a lot of storage space [9,31]. NB is usually considered as a more robust algorithm to noisy samples than RF [32]. However, Folleco et al. [33] showed that RF can provide very consistent classification results in some cases. It is worth noting that SVM is not resistant to class noise [11] and only considers a subset of feature space and relevant features. SVM, KNN, RF, NB, and DT have been used as filtering algorithms in many studies [10,12,34–37]. Ensemble models have been applied extensively in credit scoring and other areas as they are considered to be more stable than base classifiers [38]. They are also able to reduce the bias and variance of the model [39,40].

Input: E (training set)

Parameter: n (number of samples), l (number of individual learning algorithms),$A_1, A_2, \ldots, A_l$ (l types of individual learning algorithms)

Output: A (detected noisy subset of E)

(1) form n disjoint almost equally sized subsets of E_i,where $\cup_i E_i = E$

(2) $A \leftarrow \emptyset$

(3) for i =1, …, n do

(4) form $E_t \leftarrow E/E_i$

(5) for j = 1, …, l do

(6) induce H_j based on samples in E_t and A_j

(7) end for

(8) for every $e \in E_i$ do

(9) ErrorCounter ← 0

(10) for j = 1, …, l do

(11) if H_i incorrectly classifies e

(12) then *ErrorCounter*← *ErrorCounter* +1

(13) if *ErrorCounter* = l, then $A \leftarrow A \cup \{e\}$

(14) end for

(15) end for

Algorithm. 2. The Consensus Filtering (CF) algorithm.

The ensemble-based voting methods, MF, HAVF using mixed strategy, HAVF using removing strategy, and CF, have been used in this paper for noise detection. After detecting, removing/ relabeling noisy data, the bagging classification strategy is applied to calculate the accuracy (Acc), specificity, sensitivity, STD, AUC and ROC metrics related to each dataset. These six measures are used to compare the performances of the four ensemble-based methods compared in terms of dealing with noisy data.

3.2. The proposed method: HAVF

The proposed method, HAVF using removing strong and semi-strong noisy instances and relabeling weak noisy instances, improves the results of the MF method in terms of detecting noisy data. In other words, our new method can cover the gap between MF and CF in identifying noisy instances. First, the strong and semi-strong noisy instances

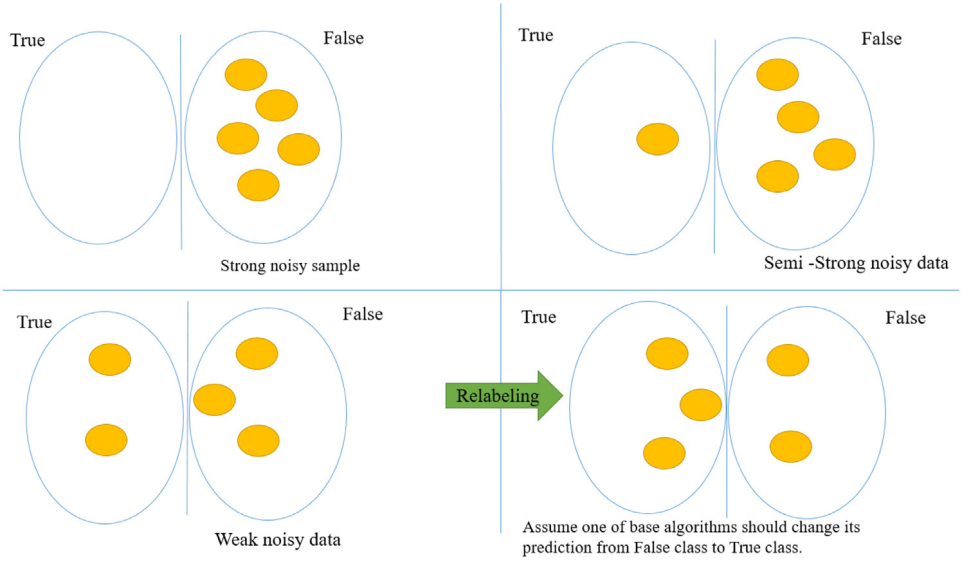

Fig. 1. Application of a classification filtering method (e.g., HAVF using mixed strategy) for different types of class noise.

are determined by the proposed method. Then, the weak noisy instances are relabeled. The definition of strong noisy instances, semi-strong noisy instances, and weak noisy instances are given below:

- **Strong noisy instance**: an instance was considered as a "strong noisy instance" if all five base classifiers classified it incorrectly. This instance should be removed from the dataset as an ensured class noisy sample [3].
- **Semi-strong noisy instance**: an instance was considered as a "semi-strong noisy instance" if only one algorithm classified it correctly. This kind of instances should be removed from the dataset as they are very likely to be potential noisy instances.
- **Weak noisy instance**: If only two classifiers predicted the label of instance correctly, the instance was considered as "weak noisy instance". Since the difference between the numbers of classifiers which can classify the instance correctly in comparison with the number of classifiers that predict it incorrectly was only one classifier, this kind of instances cannot be considered as noisy instances to be removed with confidence. Thus, instead of removing a weak noisy instance, it should rather be relabeled.

More information on these three types of class noise is given in Fig. 1.

If there are L individual algorithms, the lower bound of $[L/2] + 1$ represents the amounts of individual classifiers that predict the label of a given sample as a False label and the lower bound of $[L/2]$ represents the number of base classifiers that classify a given sample with a True label. If the ground truth label of the predicted sample is True, then we can assume it is very likely that one of the individual algorithms which assigns the False label to the sample, classifies it incorrectly. Because the difference between the numbers of classifiers that misclassify the sample, compared to the number of classifiers that are able to predict the correct class is only one algorithm, we cannot be sure that the detected noisy sample is an inherent class noise sample. The main idea of our method arises from this slight difference between the two groups of classifiers allowing us to relabel a given sample as a weak noisy sample.

To prove whether our assumption is plausible or not, the STD, AUC, and ROC evaluation metrics were considered. Our assumption would be true if the output results in terms of these criteria would be better for our new method, i.e., HAVF using mixed strategy. Algorithm 3 and Fig. 2 depict the algorithm and the block diagram of the proposed method. Moreover, the following equations were considered:

$$\sum_{i=1}^{l} \delta\left(A_i\left(\mathbf{X}\right)\right) > L \tag{4}$$

$$\sum_{i=1}^{l} \delta\left(A_i\left(\mathbf{X}\right)\right) \leq (L - l + 1) \tag{5}$$

$$\sum_{i=1}^{l} \delta\left(A_i\left(\mathbf{X}\right)\right) = (L - l + 2), \tag{6}$$

where Eq. (4) is the condition used to remove strong noisy samples, Eq. (5) is the condition used to remove semi-strong noisy samples, and Eq. (6) is the condition used to remove weak noisy samples.

Input: E (training dataset)

Parameter: n (number of samples), l (number of learning algorithms),

$A_1, A_2, \ldots, A_l$ (l kinds of learning algorithms)

Output: A (detected strong and semi-strong noisy subset of E)

Output: B (detected weak noisy subset of E)

(1) form n disjoints almost equally sized subsets of E_i, where $\cup_i E_i = E$

(2) $A \leftarrow \emptyset$

(3) $B \leftarrow \emptyset$

(4) for $i = 1, \ldots, n$ do

(5) form $E_t \leftarrow E/E_i$

(6) for $j = 1, \ldots, l$ do

(7) induce H_j based on examples in E_t and A_j

(8) end for

(9) for every $e \in E_i$ do

(10) *Errorcounter* $\leftarrow 0$

(11) for $j = 1, \ldots, l$ do

(12) if H_j incorrectly classifies e

(13) then *ErrorCounter*$\leftarrow$*ErrorCounter*+1

(14) end for

(15) if ErrorCounter>3 then $A \leftarrow A \cup \{e\}$

(16) else if *ErrorCounter* =2 then $B \leftarrow B \cup \{e\}$

(17) re-label every $e \in B$

(18) end for

(19) end for.

Algorithm. 3. The HAVF algorithm using mixed strategy.

To show the positive impact of using mixed strategy on noise detection, we have conducted a simulation study using four ensemble-based voting methods: 1- MF using removing strategy, 2- HAVF using mixed strategy, 3- HAVF using removing strategy, and 4- CF using removing strategy. The obtained results are presented in Section 4.2.

4. Experimental validation

Our simulation experiments included two main steps: pre-processing and classification. Pre-processing included noise detection as well as sample removal and relabeling by MF, HAVF using mixed strategy, HAVF using removing strategy, and CF using removing strategy. In the classification step, 90% of the pre-processed samples were randomly selected for training and 10% for testing. The applied ensemble classifiers used k-fold cross validation, in which k was set to 10. Next, the classification was carried out using bagging. Five base classifiers mentioned in Section 3.1 were used and their performance assessed for each test set. In this study, the MATLAB program was used for all implementations.

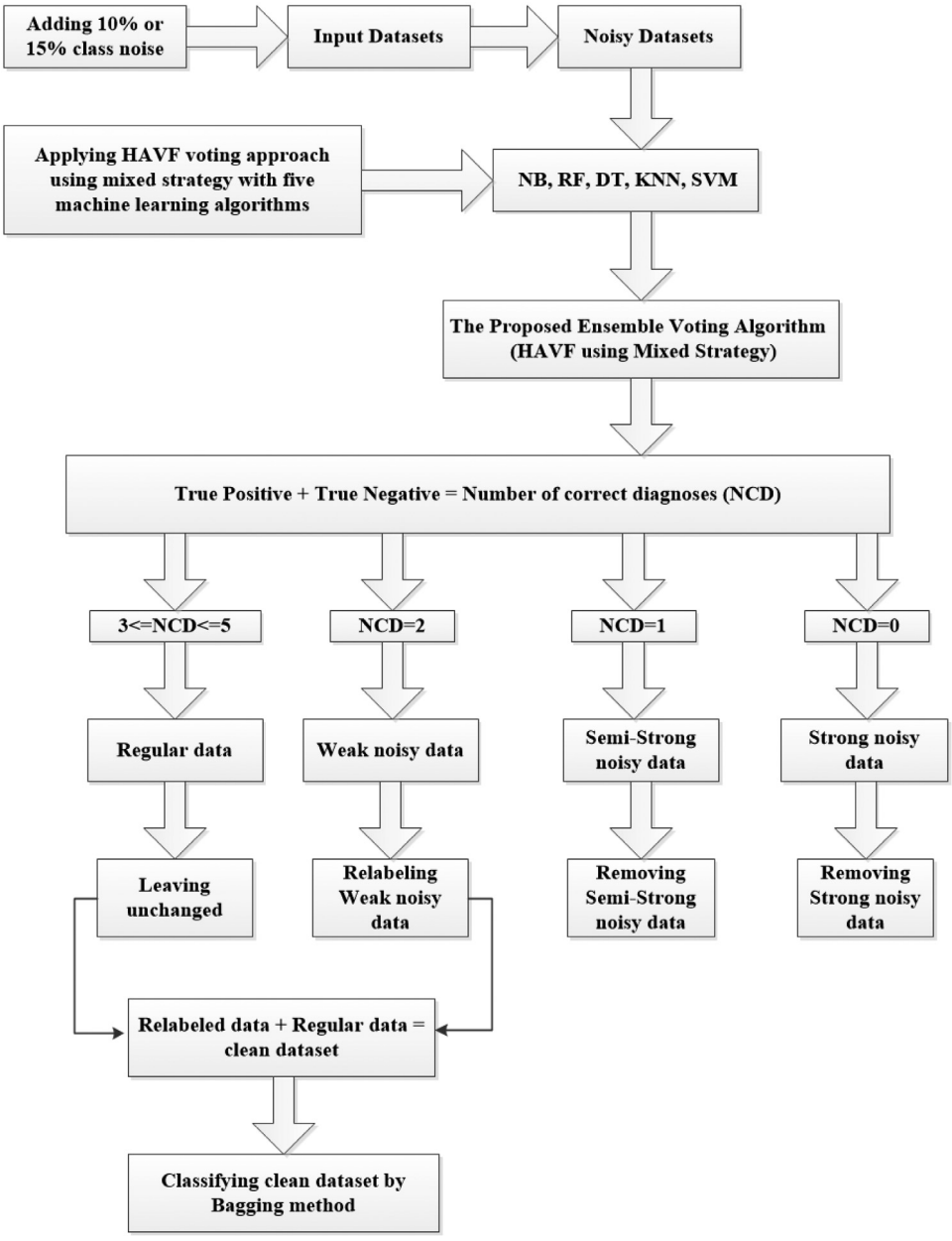

Fig. 2. Block diagram of the proposed methodology.

4.1. Experimental setup

In this paper, 16 benchmark datasets from the UCI machine learning repository have been used. All of them include instances that belong to two classes. The main characteristics of these 16 datasets are presented in Table 1. Moreover, in Table 2, we present the acronyms used for these datasets as well as for the main methods considered in our work. The methods under study were applied to these 16 datasets containing numerical types of data.

4.2. Performance analysis

After detecting, removing/relabeling the noisy instances using the four mentioned methods, the performance of each of them has been assessed. To evaluate the performance of each noise detection method, the accuracy, specificity, sensitivity, STD, AUC, and ROC metrics have been calculated for validation sets of all datasets by Bagging classifier.

Table 1
Main characteristics of the 16 real datasets from the UCI repository considered in this work.

The real names of datasets	# of instances	# of features	# of training instances	# of testing instances	# of classes
Parkinson	195	22	176	19	2
Brest cancer Wisconsin diagnostic	569	32	513	56	2
Ion sphere	351	34	316	35	2
Hepatitis	138	18	125	13	2
Vertebral columns	310	7	31	279	2
Connectionist Bench (Sonar, Mines vs. Rocks)	208	60	188	20	2
Statlog (Australian Credit Approval)	690	14	621	69	2
Mammographic	830	6	747	83	2
Statlog (Heart)	270	13	243	27	2
Congressional Voting	435	16	392	43	2
Haberman Survival	306	3	276	30	2
Blood Transfusion	748	4	674	74	2
Pima Indian Diabetes	768	8	692	76	2
Diabetic Retinopathy	1151	19	1036	115	2
Blogger	100	5	90	10	2
Tic-Tac-Toe	958	9	862	96	2

Table 2
The acronyms of datasets and methods considered in this work.

Full name	Acronym
Parkinson	Prk
Brest cancer Wisconsin diagnostic	BCWD
Ion sphere	Ion
Hepatitis	Hpt
Vertebral column	Vrtbc
Connectionist Bench (Sonar, Mines vs. Rocks)	Sonar
Statlog (Australian Credit Approval)	SAC
Mammographic Mass	Mamo
Statlog (Heart)	SHrt
Congressional Voting	Vot
Haberman Survival	Hbrs
Blood Transfusion	Bldt
Pima Indian Diabetes	Pdib
Diabetic Retinopathy	Dibt
Tic-Tac-Toe	TTT
Blogger	Blgr
Support Vector Machine	SVM
Naïve Bayes	NB
Random Forest	RF
Decision Tree	DT
K Nearest Neighbor	KNN
High Agreement Voting Filtering using mixed strategy	Proposed method
High Agreement Voting Filtering using removing strategy	HAVF
Majority Filtering	MF
Consensus Filtering	CF
University of California, Irvine	UCI
Standard Deviation	STD
Accuracy	Acc
Area Under the Curve	AUC
Receiver Operating Characteristics	ROC
True Positive	TP
True Negative	TN
False Negative	FN
False Positive	FP

4.2.1. Performance measures

Accuracy, sensitivity, specificity, AUC, STD, and Receiver Operating Characteristic (ROC) metrics are frequently applied in machine learning as measurement criteria based on the consideration that a test sample could be either a false positive (FP), or a false negative (FN), or a true positive (TP), or a true negative (TN). If the system classifies the test sample into a positive class, while it is negative, it is called FP. If the system labels it as a negative, but it is positive, it is known as FN. Moreover, if the classifier predicts the label of the positive and negative test samples correctly, they are named TP and TN, respectively [41]. The confusion matrix [42] is used to evaluate the performance of the applied voting methods more accurately. To assess the obtained results of the four filtering methods, the accuracy, sensitivity and specificity were

computed using the confusion matrix [43] and Eqs. (7), (8), and (9), respectively.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FN} + \text{FP}), \tag{7}$$

$$\text{Sensitivity} = (\text{TP})/(\text{TP} + \text{FN}), \tag{8}$$

$$\text{Specificity} = (\text{TN})/(\text{TN} + \text{FP}). \tag{9}$$

4.2.2. The effects of the methods on each dataset

As mentioned before, our new method can remove less data using relabeling strategy compared to the other methods, as clearly shown in Table 3. The results presented in Table 3 confirm that the proposed method tends to maintain more correct instances. According to Table 3, the number of removed noisy instances by our method is lower than that removed by MF and higher than that removed by CF for all considered datasets and both noise levels. Keeping more correct instances in the dataset can be considered as the main advantage of our method, while a higher number of remaining instances ensured by CF confirms its inability to recognize mislabeled instances. Although the correct instances identified by our new method and by HAVF are the same, the evaluation results present a significant advantage of the new method consisting of relabeling mislabeled data.

4.3. Results provided by the filtering methods

In this section, the results in terms of the accuracy, sensitivity, specificity, STD, and AUC criteria, obtained for the 16 considered datasets are assessed after carrying out the four specified filtering methods used to detect, remove or relabel mislabeled data for two-class noisy datasets. We first present the results obtained for data with 10% of noisy class labels, followed by those obtained for 15% of noisy class labels. The results are reported in Tables 4–8.

4.3.1. Comparison of the accuracy results

The accuracies achieved by each of the four filtering methods have been compared for both considered noise values (i.e., 10% and 15% of noisy samples). These accuracy results are reported in Table 4. The best obtained accuracies are highlighted in bold.

- **Accuracy results for 10% of class noise**

On one hand, Table 4 shows that the proposed method provides better accuracy than MF for 11 datasets. Moreover, it is worth noting that after applying the proposed method, HAVF using relabeling strategy, the obtained accuracy per dataset is greater than the accuracy achieved in the case of using both HAVF and CF. On the other hand, Table 4 indicates that MF was able to provide better accuracy than the proposed method only for 5 of 16 datasets. Furthermore, MF provided a greater accuracy value compared to HAVF and CF for 9 and 16 datasets, respectively. Table 4 also shows that the performance of HAVF is superior compared to CF for all datasets. CF was the worst method overall when the data with 10% of class noise were investigated.

- **Accuracy results for 15% of class noise**

Table 4 shows that the proposed method was able to provide achieve a higher accuracy value than MF for 9 of 16 datasets. Furthermore, we can observe that by applying the proposed method we were able to get better results than HAVF for 15 datasets. Noteworthy, the proposed method had a better performance compared to CF for all 16 datasets.

We were able to obtain better accuracy values for 9 datasets when MF was used compared to HAVF. Finally, the application of MF allowed us to gain better accuracy values than CF for all 16 datasets. It can be also seen in Table 4 that HAVF has a higher performance than CF for all 16 datasets. The order of methods concerning noise detection remained the same as for 10% of class noise. Thus, overall, the proposed method was still the best one and CF was still the worst one.

4.3.2. Comparison of the sensitivity results

The sensitivities achieved by each of the four filtering methods have been compared for both considered noise values (i.e., 10% and 15% of noisy samples). These sensitivity results are reported in Table 5. The best obtained sensitivities are highlighted in bold.

- **Sensitivity results for 10% of class noise**

Table 5 shows the results that for 10% of class noise the proposed method generates better sensitivities than MF and CF for 11 and 13 datasets (out of 16), respectively. Moreover, the proposed method yields greater sensitivities for 11 datasets compared to HAVF. Based on the sensitivity criterion, MF has a better performance for only 6 datasets when compared to HAVF, which has superiority over MF on 10 datasets. HAVF was able to generate better sensitivities than CF for all datasets except for TTT. CF was able to provide a higher sensitivity value only for one dataset.

- **Sensitivity results for 15% of class noise**

For this type of noise condition, the proposed method was able to achieve greater sensitivities, compared to MF, for 9 datasets, and for 13 datasets compared to HAVF. Table 5 also shows that CF has no superiority over the other methods except for the TTT dataset. MF provided greater sensitivities compared to HAVF and CF for 10 and 15 datasets, respectively. Furthermore, HAVF achieved better sensitivities in comparison with the proposed method only for 3 datasets. In addition, HAVF yielded better sensitivities for 6 and 15 datasets, compared to MF and CF, respectively.

Table 3
This table presents the data used in our simulation study. Noise level — indicates two levels of the added class noise, Weak — indicates the number of relabeled weak noisy samples, SNE — indicates the number of strong and semi-strong noisy samples detected and removed by HAVF and by our new method, Remain — indicates the number of correct samples remained after identifying and removing noisy samples, NE — indicates the number of noisy samples detected by MF and CF, Proposed method — is the HAVF method using relabeling strategy, and HAVF — is the HAVF method using removing strategy.

Dataset	Noise level	Proposed method			MF		HAVF		CF	
		Weak	# SNE	Remain	# NE	Remain	# SNE	Remain	# NE	Remain
Prk	10%	10	29	166	33	162	29	166	8	187
	15%	16	33	162	36	159	33	162	11	184
BCWD	10%	24	50	519	76	493	24	519	17	552
	15%	23	78	491	101	468	78	491	29	540
Ion	10%	16	40	311	58	293	40	311	18	333
	15%	13	47	304	58	293	47	304	18	333
Hpt	10%	16	28	110	43	95	38	100	3	135
	15%	20	38	100	64	74	100	38	10	128
Vrtbc	10%	28	34	276	40	270	34	276	15	295
	15%	25	59	251	63	247	59	251	39	271
Sonar	10%	23	29	179	39	169	29	179	7	201
	15%	21	31	177	51	157	31	177	9	199
Sac	10%	71	77	613	103	587	71	613	17	673
	15%	64	118	572	143	547	118	572	39	651
Mamo	10%	68	152	678	173	657	152	678	58	772
	15%	75	169	661	261	569	169	661	79	751
SHrt	10%	26	48	222	72	198	48	222	6	264
	15%	36	49	221	93	177	49	221	14	256
Vot	10%	18	50	385	67	368	50	385	29	406
	15%	17	67	368	85	350	67	368	48	387
Hbrs	10%	30	76	230	88	218	76	230	31	275
	15%	32	76	230	98	208	76	230	36	270
Bldt	10%	73	159	589	253	531	159	589	47	701
	15%	40	172	576	231	517	172	576	80	668
Pima	10%	97	142	626	230	538	142	626	58	710
	15%	118	128	620	243	505	128	620	47	701
Diabetes	10%	190	251	900	438	713	251	900	92	1059
	15%	204	251	900	469	682	251	900	92	1059
Blogger	10%	4	19	81	23	77	19	81	13	87
	15%	11	25	75	34	66	25	75	16	84
TTT	10%	212	104	854	309	649	104	854	62	896
	15%	150	176	782	324	634	176	782	93	865

Table 4
Results obtained for the accuracy criterion.

Datasets	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
Prk	**0.9313**	0.9063	0.9231	0.8394	**0.8969**	0.8647	0.8825	0.7911
BCWD	0.9567	**0.9761**	0.9488	0.8942	0.9502	**0.9570**	0.9402	0.8604
Ion	**0.9535**	0.9353	0.9452	0.8730	**0.9420**	0.9207	0.9253	0.8324
Hpt	**0.8627**	0.8422	0.8427	0.6823	**0.8350**	0.8200	0.8220	0.6192
Vrtbc	**0.9230**	0.8819	0.9130	0.8690	**0.9284**	0.8825	0.8967	0.8463
Sonar	**0.8812**	0.8419	0.8256	0.7705	**0.8624**	0.8567	0.8253	0.7642
SAC	**0.9107**	0.8991	0.8985	0.8261	**0.9242**	0.8907	0.8982	0.7986
Mamo	**0.9215**	0.8032	0.9200	0.7971	0.9120	**0.9686**	0.9241	0.8179
SHrt	0.8809	**0.9395**	0.8655	0.7476	0.8500	**0.9224**	0.8450	0.7240
Vot	0.9771	**0.9828**	0.9700	0.9100	0.9706	**0.9774**	0.9642	0.9195
Hbrs	**0.8735**	0.8357	0.8639	0.7333	**0.8939**	0.8540	0.8878	0.7404
Bldt	**0.9769**	0.9592	0.9281	0.7611	0.9279	**0.9514**	0.9144	0.7805
Pima	0.8819	**0.9242**	0.8573	0.7603	0.8569	**0.9198**	0.8435	0.7511
Diabetes	0.8121	**0.8780**	0.8010	0.6944	0.7978	**0.8874**	0.7833	0.6849
Blogger	**0.9688**	0.7290	0.9550	0.7087	**0.9700**	0.9367	0.9429	0.9038
TTT	**0.9489**	0.9392	0.9261	0.9175	**0.9319**	0.9249	0.8909	0.8734

Table 5
Results obtained for the sensitivity criterion.

Datasets	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
Prk	**0.8469**	0.7119	0.7048	0.6384	**0.6773**	0.5945	0.6116	0.5303
BCWD	0.9784	**0.9857**	0.9738	0.9313	0.9502	**0.9570**	0.9402	0.8604
Ion	0.9631	0.9436	**0.9679**	0.9269	**0.9586**	0.9433	0.9425	0.8861
Hpt	**0.9071**	0.8636	0.8856	0.74730	**0.84**	0.7539	0.8059	0.6208
Vrtbc	**0.9631**	0.9440	0.9505	0.9317	**0.9489**	0.9281	0.9339	0.8764
Sonar	**0.7634**	0.7054	0.7084	0.6599	**0.8671**	0.8239	0.8075	0.7270
SAC	**0.8902**	0.8749	0.8793	0.7980	**0.8836**	0.8525	0.8775	0.7469
Mamo	0.9168	0.8067	**0.9290**	0.8209	0.8897	**0.9575**	0.9252	0.8017
SHrt	0.9041	0.8894	**0.9544**	0.7712	0.8989	**0.9289**	0.8968	0.7881
Vot	0.9766	**0.9841**	0.9816	0.9101	0.9595	**0.9789**	0.9601	0.9204
Hbrs	**0.9472**	0.9358	0.9401	0.8547	**0.9691**	0.9578	0.9600	0.8698
Bldt	**0.8960**	0.8395	0.7209	0.3645	0.7530	**0.7945**	0.7117	0.4732
Pima	0.7848	**0.8629**	0.7473	0.6179	0.6830	**0.8061**	0.6796	0.5757
Diabetes	0.8510	**0.9025**	0.8126	0.7048	0.8673	**0.9346**	0.8419	0.7450
Blogger	**1**	0.8432	0.9757	0.8350	**0.9847**	0.9760	0.9691	0.9306
TTT	0.1832	0.2766	0.8618	**0.8735**	0.1320	0.0966	0.6894	**0.7602**

Table 6
Results obtained for the specificity criterion.

Datasets	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
Prk	**0.9826**	0.9778	0.9791	0.9282	**0.9669**	0.9583	0.9605	0.8961
BCWD	0.9211	**0.96**	0.9108	0.8337	**0.9601**	0.9223	0.8937	0.7721
Ion	**0.9473**	0.9206	0.9123	0.7882	**0.9184**	0.8867	0.8930	0.7561
Hpt	**0.8057**	0.7811	0.7800	0.6041	**0.8422**	0.5644	0.8410	0.0605
Vrtbc	**0.8359**	0.7315	0.8236	0.7315	**0.8818**	0.7926	0.8219	0.7893
Sonar	**0.9574**	0.9446	0.9117	0.8599	0.8562	**0.8933**	0.8488	0.8035
SAC	**0.9293**	0.9144	0.9152	0.8530	**0.9611**	0.9207	0.9145	0.8368
Mamo	**0.9278**	0.8017	0.9213	0.7742	0.9294	**0.9756**	0.9236	0.8316
SHrt	0.8574	**0.9206**	0.8390	0.7281	0.7761	**0.8971**	0.7790	0.6598
Vot	0.9774	**0.9822**	0.9753	0.9143	**0.9902**	0.9768	0.9690	0.9224
Hbrs	**0.5607**	0.4551	0.5292	0.3901	0.5250	0.2272	**0.5298**	0.3558
Bldt	**0.9914**	0.9735	0.9749	0.9056	0.9697	**0.9789**	0.9600	0.8926
Pima	0.9318	**0.9542**	0.9113	0.8426	0.9325	**0.9661**	0.9227	0.8527
Diabetes	0.7666	**0.8483**	0.7890	0.6855	0.6683	**0.8014**	0.6877	0.6039
Blogger	**0.8148**	0.3591	0.7424	0.4237	**0.7649**	0.4570	0.6216	0.7049
TTT	0.8574	**0.9206**	0.8390	0.7281	**0.9983**	0.9969	0.9540	0.9244

4.3.3. Comparison of the specificity results

The specificities achieved by each of the four filtering methods have been compared for both considered noise values (i.e., 10% and 15% of noisy samples). These specificity results are reported in Table 6. The best obtained specificities are highlighted in bold.

- **Specificity results for 10% of class noise**

As reported in Table 6 the proposed method enabled us to achieve better specificities for 10 datasets (out of 16) in comparison with MF. Based on the specificity criterion, our new method has a better performance than HAVF and CF for 14 and 15 datasets, respectively. Clearly, CF was the worst method in terms of the specificity criterion for 10% of class noise data. As can be seen in Table 6, we can achieve better specificities for 9 datasets when MF is used, compared to HAVF.HAVF using removing strategy generated a better specificity value compared to our new method only for one dataset.

- **Specificity results for 15% of class noise**

Table 6 reports that greater specificities were obtained by our method for 10 datasets compared to MF when the data with 15% of class noise were used. The proposed method enabled us to get better specificities for 13 datasets compared to HAVF. It is worth noting that CF was never able to achieve a better specificity result compared to the other methods. As presented in Table 6, higher specificities were obtained for 6 datasets using MF, compared to the proposed method. Moreover, MF showed a better performance than HAVF for 10 datasets.

4.3.4. Comparison of the STD results

The STD values provided by each of the four filtering methods have been compared for both considered noise values (i.e., 10% and 15% of noisy samples). These STD results are reported in Table 7. The best obtained STD values are highlighted in bold.

Table 7
Results obtained for the STD criterion.

Datasets	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
Prk	**0.0072**	0.0092	0.0106	0.0180	**0.0079**	0.0110	0.0184	0.0158
BCWD	0.0048	**0.0019**	0.0035	0.0049	0.0040	**0.0017**	0.0046	0.0069
Ion	**0.0038**	0.0057	0.0061	0.0085	**0.0052**	0.0061	0.0063	0.0067
Hpt	**0.0151**	0.0202	0.0210	0.0159	**0.0118**	0.0228	0.0181	0.0239
Vrtbc	**0.0080**	0.0081	0.0093	0.0083	0.0076	**0.0058**	0.0129	0.0141
Sonar	**0.0110**	0.0202	0.0139	0.0123	**0.0084**	0.0173	0.0104	0.0193
SAC	0.0055	**0.0021**	0.0022	0.0050	**0.0036**	0.0052	0.0048	0.0037
Mamo	**0.0060**	0.0069	0.0061	0.0062	0.0044	**0.0024**	0.0048	0.0045
SHrt	0.0151	**0.0100**	0.0094	0.0118	0.0127	0.0104	**0.0081**	0.0209
Vot	**0.0033**	0.0039	0.0037	0.0065	0.0033	0.0044	**0.0031**	0.0053
Hbrs	0.0063	0.0098	**0.0041**	0.0153	**0.0075**	0.0099	0.0142	0.0118
Bldt	0.0034	0.0031	0.0060	**0.0030**	**0.0039**	0.0040	0.0055	0.0095
Pima	0.0072	0.0064	0.0058	**0.0050**	0.0042	**0.0039**	0.0095	0.0060
Diabetes	0.0061	**0.0044**	0.0048	0.0088	0.0060	**0.0056**	0.0059	0.0094
Blogger	**0.0066**	0.0268	0.0230	0.0306	**0.0171**	0.0205	0.0223	0.119
TTT	**0.0019**	0.0056	0.0020	0.0037	**0.0020**	0.0021	0.0065	0.0057

Table 8
Results obtained for the AUC criterion.

Datasets	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
Prk	**0.8958**	0.8167	0.8451	0.7359	**0.8126**	0.7690	0.7935	0.6876
BCWD	0.9433	**0.9739**	0.9341	0.8841	0.9449	**0.9567**	0.9349	0.8506
Ion	**0.9573**	0.9337	0.9447	0.8496	**0.9399**	0.9191	0.9122	0.8068
Hpt	**0.8580**	0.8435	0.8282	0.6447	**0.8769**	0.8542	0.8502	0.6573
Vrtbc	0.8925	0.8650	**0.8934**	0.8375	**0.9104**	0.8489	0.8989	0.6643
Sonar	**0.8590**	0.8073	0.8254	0.7900	**0.9016**	0.8551	0.8235	0.7897
SAC	**0.9059**	0.8920	0.8990	0.8244	**0.9273**	0.8850	0.8942	0.7884
Mamo	**0.9207**	0.8109	0.9194	0.7914	0.9169	**0.9613**	0.9274	0.8177
SHrt	0.8938	**0.9300**	0.8720	0.7506	0.8730	**0.9031**	0.8219	0.7173
Vot	**0.9832**	0.9634	0.9757	0.9110	**0.9721**	0.9144	0.9693	0.9040
Hbrs	**0.7717**	0.7300	0.7293	0.6007	0.6643	0.6426	**0.7184**	0.6114
Bldt	0.8026	0.8649	**0.9263**	0.6202	0.8509	**0.8909**	0.8482	0.6947
Pima	0.8646	**0.9029**	0.8386	0.7255	0.8219	**0.8826**	0.8063	0.7060
Diabetes	0.8168	**0.8830**	0.8032	0.6874	0.7629	**0.8775**	0.7577	0.6738
Blogger	**0.9773**	0.6156	0.9417	0.6470	**0.9912**	0.8235	0.8994	0.8538
TTT	0.5543	0.9112	0.6471	**0.9305**	0.5100	0.5678	**0.8241**	0.8212

- **STD results for 10% of class noise**

Based on the STD criterion, our new method had a better performance for 10 datasets (out of 16) when it was compared to HAVF on the data with 10% of class noise. The proposed method outperformed MF on 10 datasets. Moreover, on 12 datasets the proposed method also outperformed CF. Based on the STD criterion, MF showed a better performance for 9 datasets when compared to HAVF. The use of MF allowed us to obtain greater STD values than CF for 10 datasets. The application of HAVF led to better STD results for 11 datasets in comparison with CF. Finally, CF provided better STD results than MF for 7 datasets.

- **STD results for 15% of class noise**

As reported in Table 7, the proposed method provided better performance for 10 datasets in comparison with MF when the data with 15% of class noise were used. The proposed method also generated better STD results for 13 datasets in comparison with HAVF. Moreover, the application of the proposed method resulted in better STD values for all 16 datasets in comparison with CF. As can be seen in Table 7, the application of MF allowed us to get better STD values for 11 datasets in comparison with HAVF. Finally, MF showed a better performance for 15 datasets in comparison with CF, and HAVF allowed us to generate better results for 10 datasets when compared to CF.

4.3.5. Comparison of the AUC results

The AUC values provided by each of the four filtering methods have been compared for both considered noise values (i.e., 10% and 15% of noisy samples). These AUC results are reported in Table 8. The best obtained AUC values are highlighted in bold.

- **AUC results for 10% of class noise**

As reported in Table 8, the proposed method was able to generate better AUC results for 13 datasets (out of 16) when compared to HAVF for the data with 10% of class noise. The proposed method also provided better AUC results for 10

datasets in comparison with MF. Finally, CF was superior to the proposed method for only one dataset. Moreover, MF provided better AUC results for 7 datasets in comparison with HAVF. Finally, MF generated better AUC values for 14 datasets in comparison with CF, and HAVF outperformed CF for 14 datasets.

• **AUC results for 15% of class noise**

For this type of noise condition, the proposed method outperformed MF on 9 datasets in terms of AUC. Our new method provided better performance than HAVF for 13 datasets, and than CF for 15 datasets. MF had a better performance than HAVF for 9 datasets, and MF provided greater AUC values compared to CF for 14 datasets. Finally, the HAVF method outperformed CF on all 16 datasets considered in this study.

4.3.6. Comparison of the ROC results

The area under the curve (AUC) summarizes the location of the ROC. The AUC is a mixed measure of sensitivity and specificity explaining the inherent validity of diagnostic tests.

The maximum value of AUC = 1 means that the performed test is perfectly suited for the differentiation between two different class labels. This happens when the distributions of the test results for the two classes do not overlap. Moreover, the value of AUC = 0.5 refers to the curve located on the diagonal line of the ROC space, showing a certain chance for discrimination. However, the case when AUC = 0 reveals the fact that the test results are incorrectly categorized [44].

The ROC curves for all the 16 benchmark datasets considered in our study were determined for the four mentioned noise filtering methods. The ROC curves obtained after detecting, removing and relabeling noisy data using the four noise filtering methods for 10% class noise are illustrated in Figs. 3a, 3b, 3c, and 3d. Likewise, to compare the performance of each noise filtering method for 15% class of noise, the corresponding ROC curves were depicted in Figs. 4a, 4b, 4c, 4d. For a better understanding of the relationship between the ROC curves and the performance of each method, we refer the reader to the results reported for the AUC criterion. As known, there is a direct relationship between the AUC value and the ROC performance. The higher the AUC value, the better the ROC performance. In case of adding 10% of noise labels (see Figs. 3a, 3b, 3c, and 3d) the proposed method provided the best results for 9 of 16 benchmark datasets. MF generated the most accurate ROC results for 5 datasets and HAVF for 2 datasets. As shown in Figs. 4a, 4b, 4c, and 4d regarding 15% of class noise, the proposed method allowed us to achieve the most accurate ROC results for 8 datasets, MF for 6 datasets, and HAVF for 2 datasets. CF fails to achieve the most accurate result on any dataset.

5. Discussion

In Section 5.1, we first provide an example of the recognition of mislabeled data. Thereafter, in Section 5.2, we review and compare the performance of our new method with those of existing approaches. Finally, the advantages and weaknesses of the proposed method, as well as those of MF, CF, and HAVF, are discussed in Section 5.3.

5.1. Visualizing noisy data

For more clarity, we present a visualization of correct original samples of the Vrtbc dataset in Fig. 5. Then, we illustrate the noisy samples added to it (shown by green arrows in Fig. 6; here 10% of class noise was added). The weak noisy samples which are identified and relabeled by our new method are shown by green arrows in Fig. 7.

5.2. Comparison with existing methods

Several researchers who proposed new noise detection methods have assessed their performance using benchmark data. To show the effectiveness of the proposed method, we compared our accuracy results with those provided by the existing approaches (see Table 9). Table 9 reports the best accuracies obtained for 10 real datasets by the proposed method and the existing approaches. The results presented in this table reveal that the proposed method provided the highest accuracy in the majority of cases in the presence of 10% of noise. The only case where the existing approach outperformed our new method was in the case of the BCWD dataset for which CF_1 [4] and MF_MF [30] were the best performers.

5.3. Comparison of the best results provided by the four compared noise filtering methods

The number of datasets in which each noise filter method provides the best result was reported in Table 10 for both noise levels, 10% and 15%. The experimental criteria used for evaluation were: accuracy, sensitivity, specificity, STD, and AUC. Observing the results indicated in Table 10, we can conclude that, based on the five selected evaluation criteria, the proposed method is the most accurate one with 10% of noisy samples. MF, HAVF, and CF got the second, third, and fourth place, respectively. The results obtained for 15% of class noise reveal that the proposed method is still the best one, while CF remains the worst one. It is worth noting that MF usually provided better results for data with 15% of class noise than for data with 10% of class noise.

• **Advantages and weaknesses of each filtering method**

As discussed in Section 4, the proposed method is the most accurate one compared to MF, CF, and HAVF, based on the selected evaluation criteria, namely, accuracy, sensitivity, specificity, STD, AUC, and ROC. One of the main advantages of

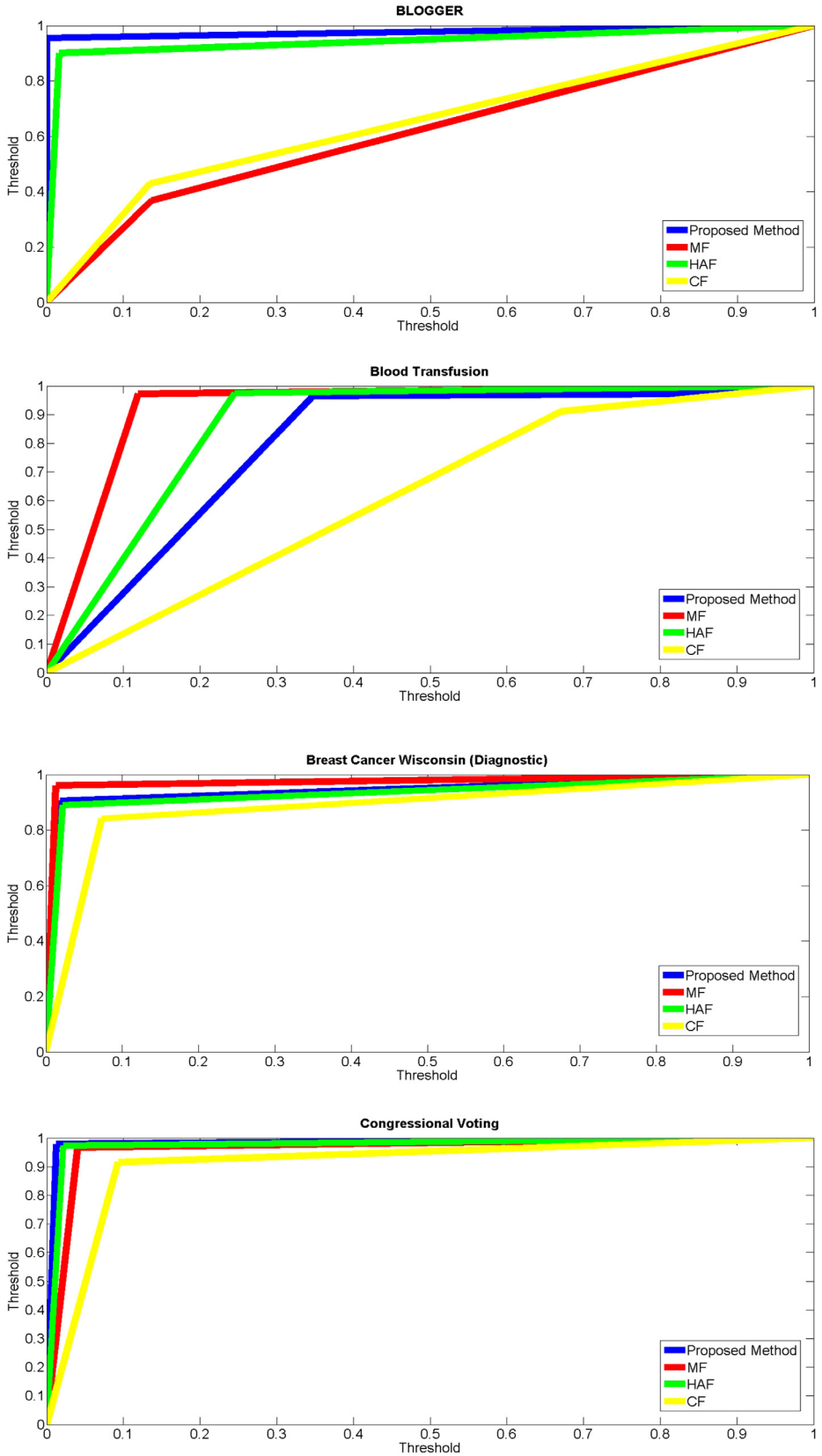

Fig. 3a. ROC curves for 10% of added noise.

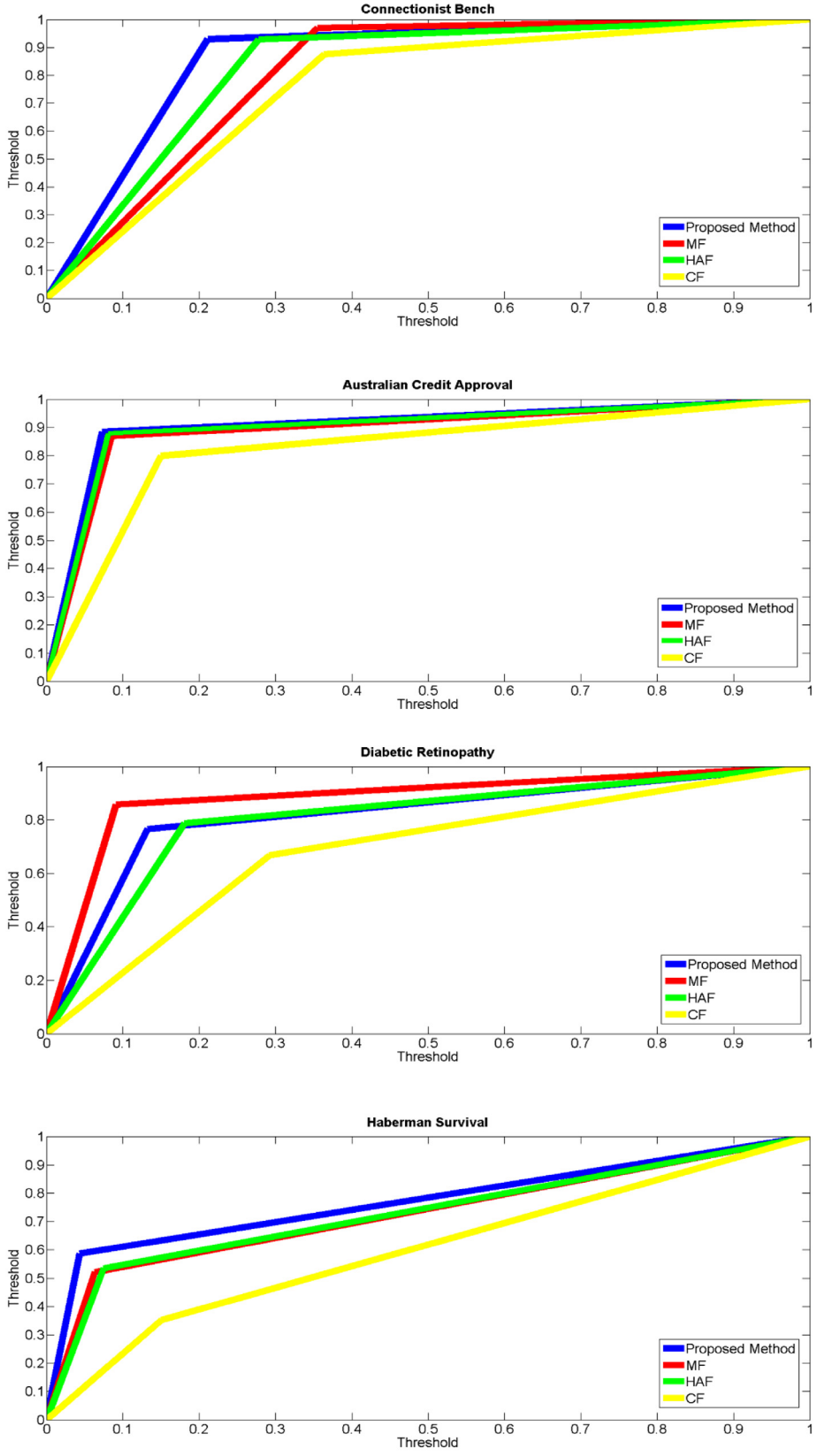

Fig. 3b. ROC curves for 10% of added noise.

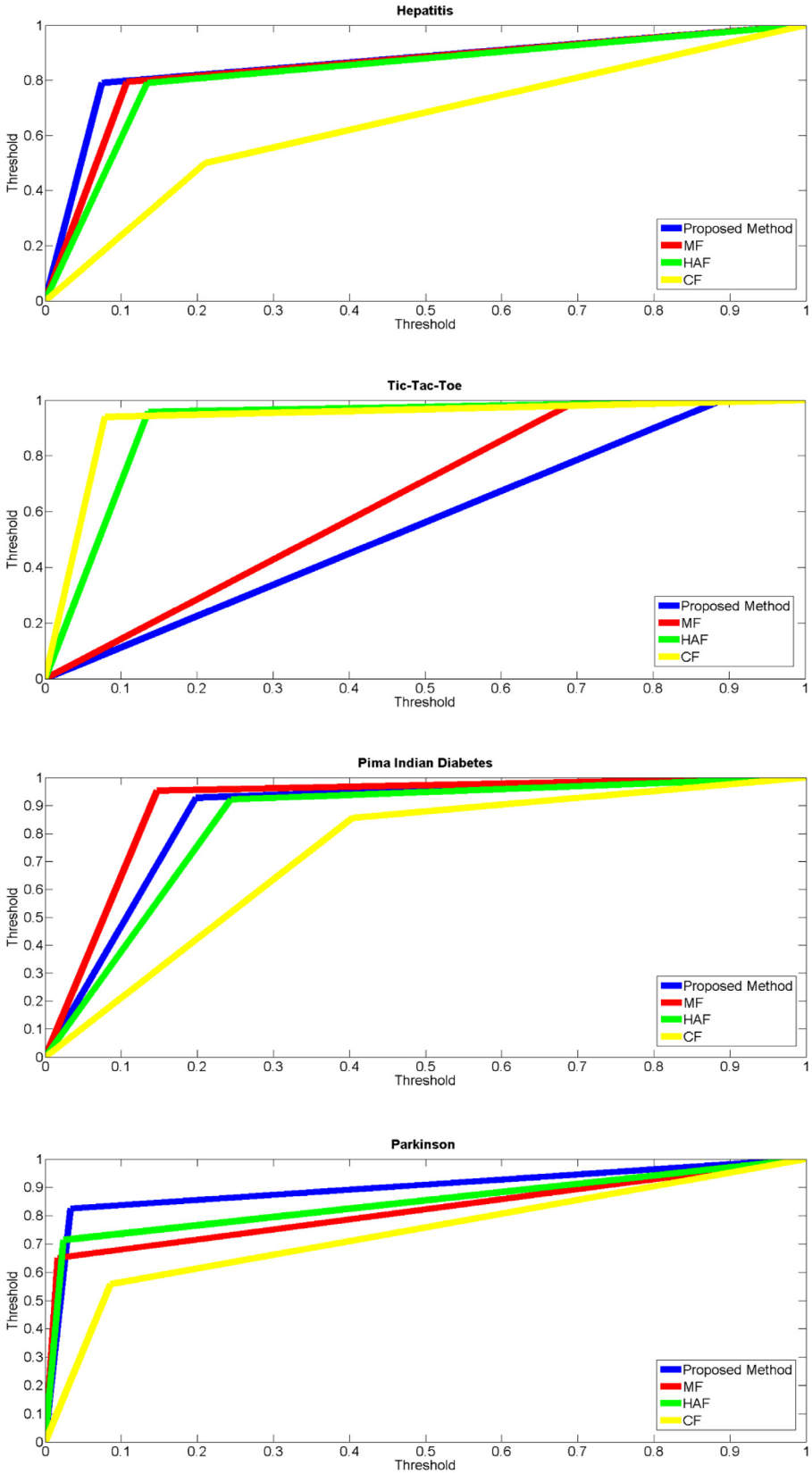

Fig. 3c. ROC curves for 10% of added noise.

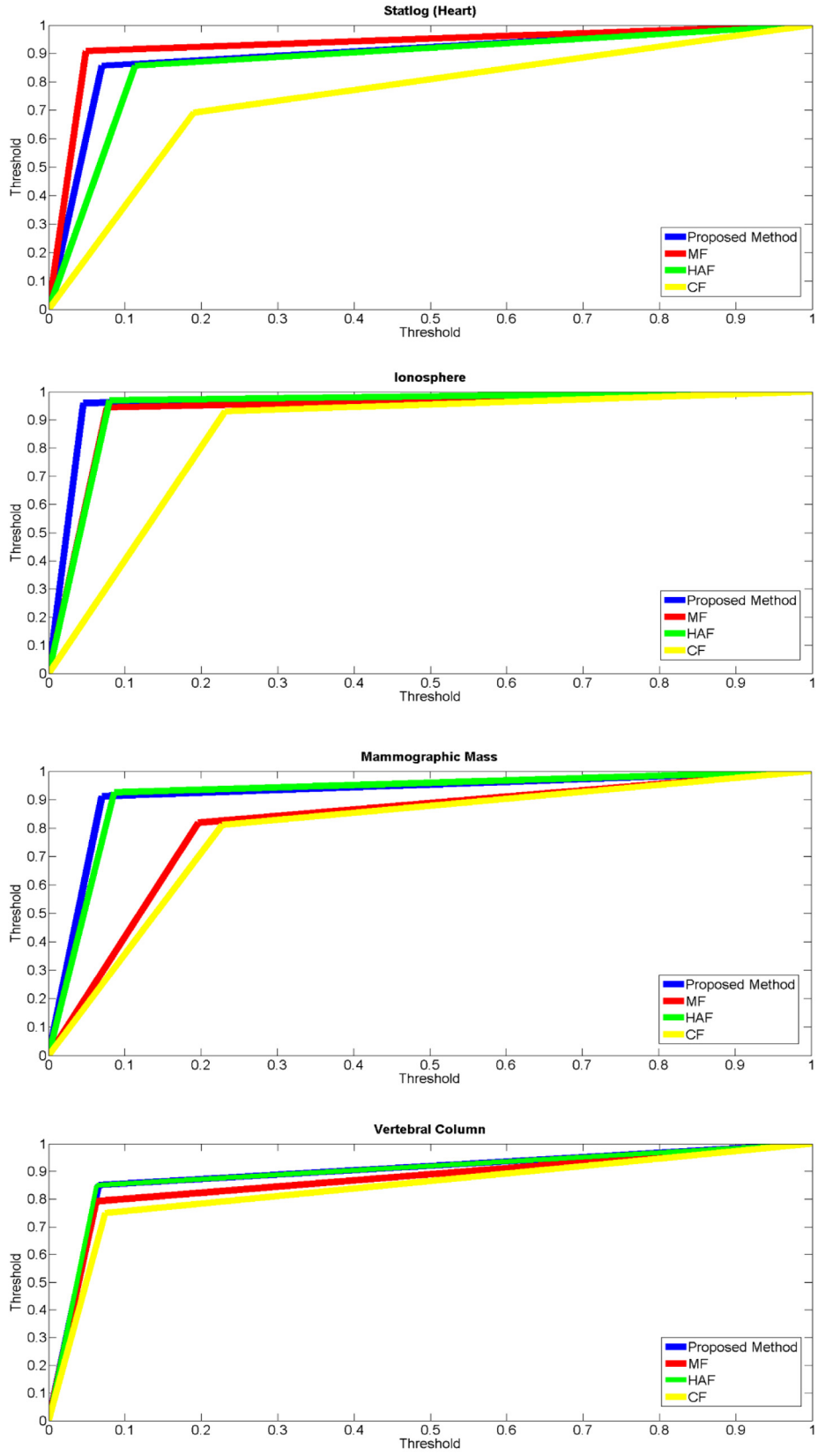

Fig. 3d. ROC curves for 10% of added noise.

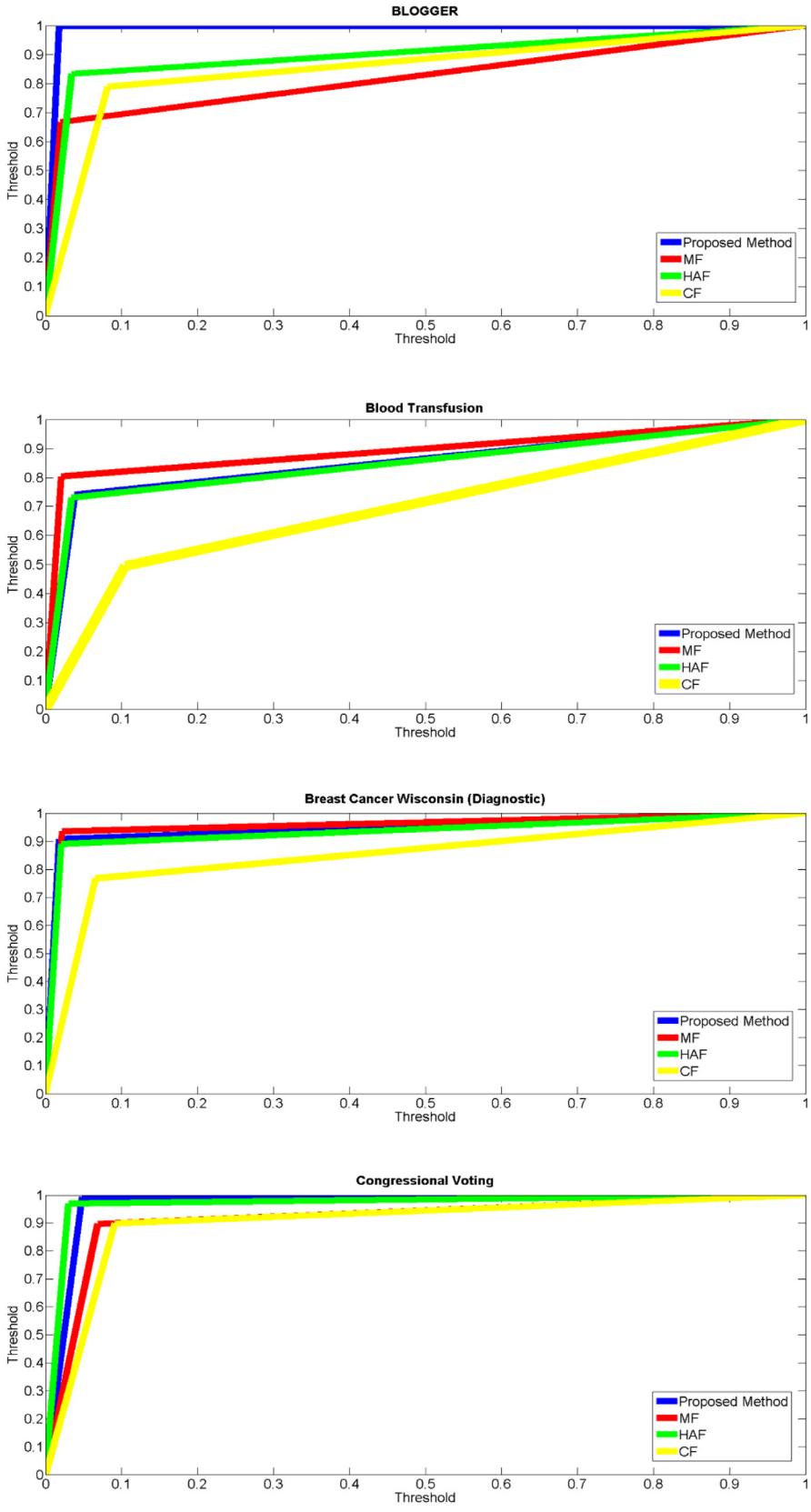

Fig. 4a. ROC curves for 15% of added noise.

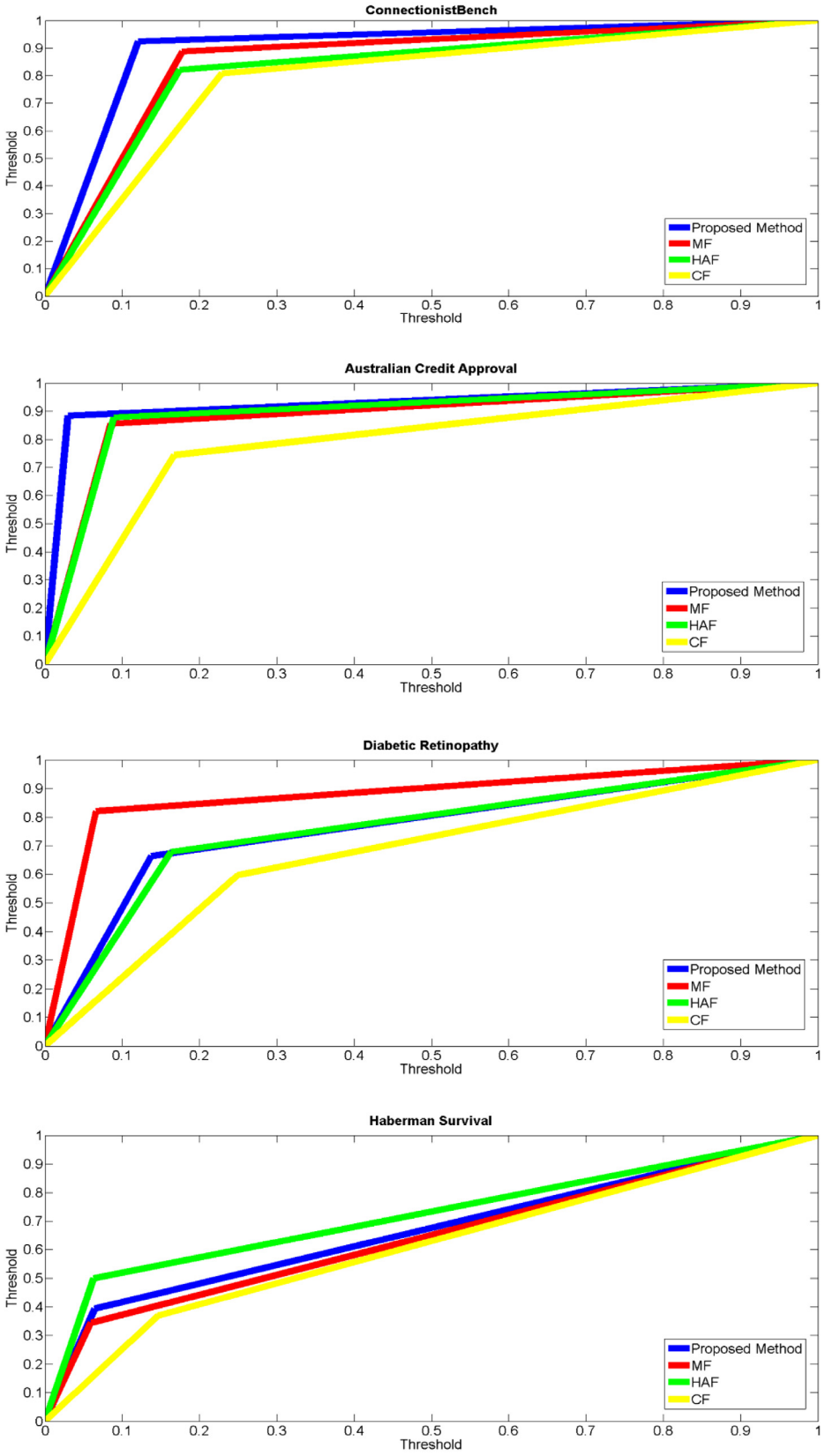

Fig. 4b. ROC curves for 15% of added noise.

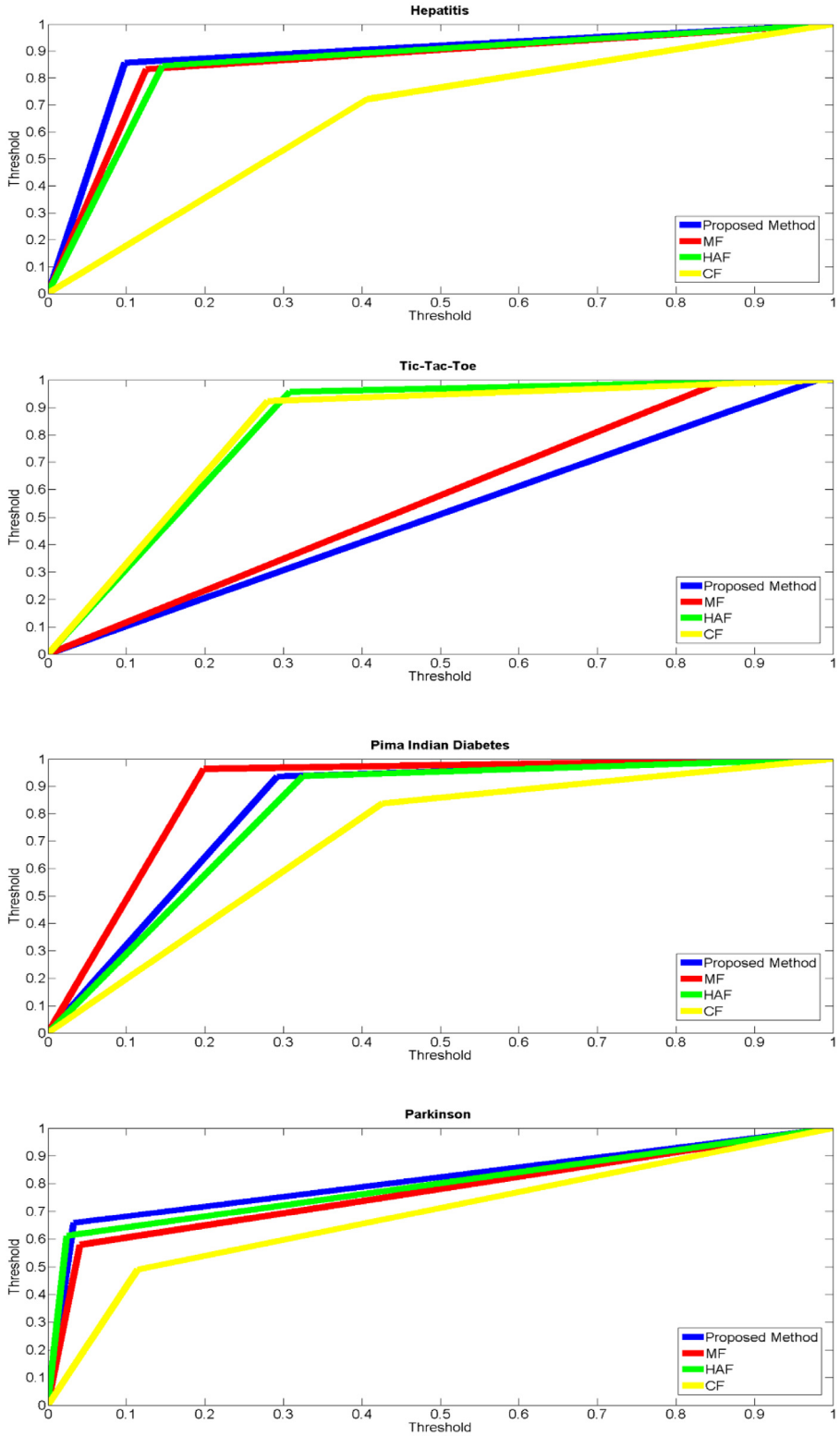

Fig. 4c. ROC curves for 15% of added noise.

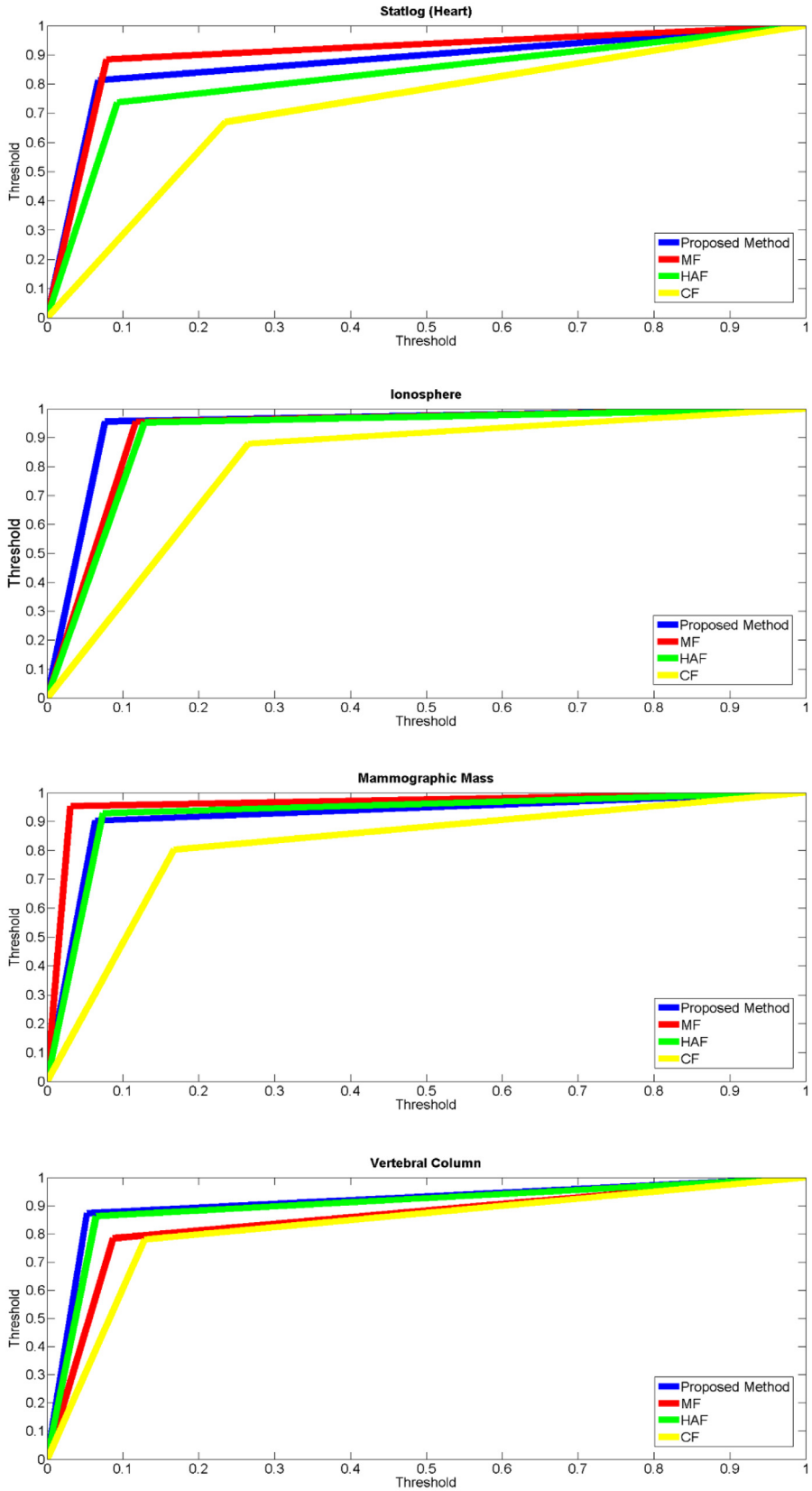

Fig. 4d. ROC curves for 15% of added noise.

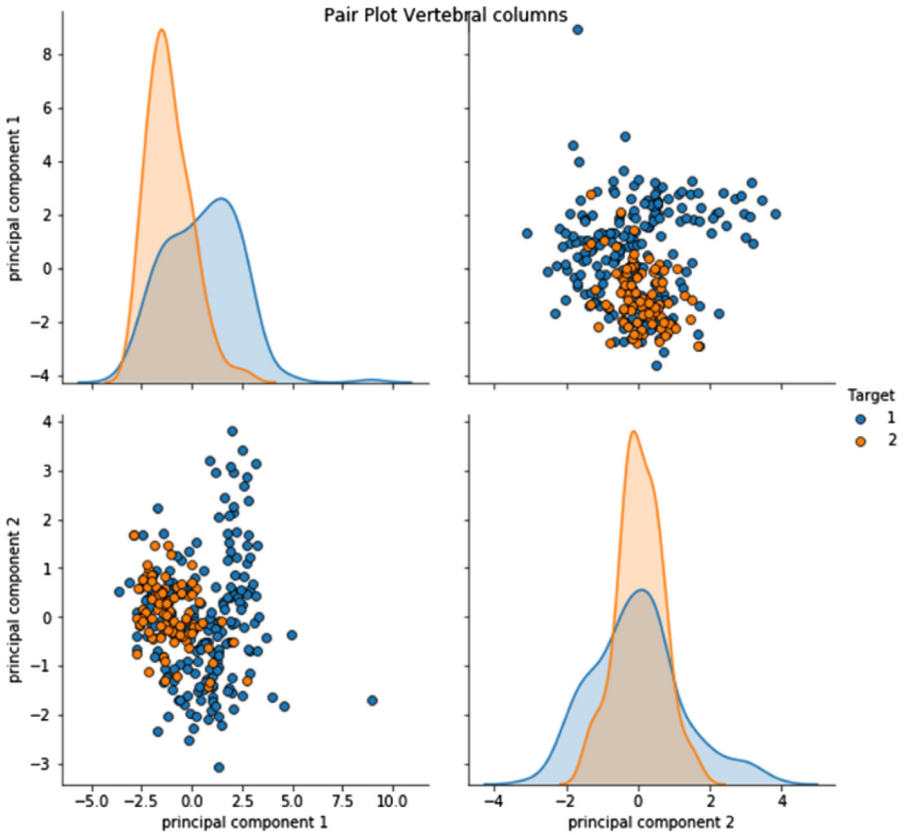

Fig. 5. Visualizing the distribution of the original samples of the Vrtbc dataset.

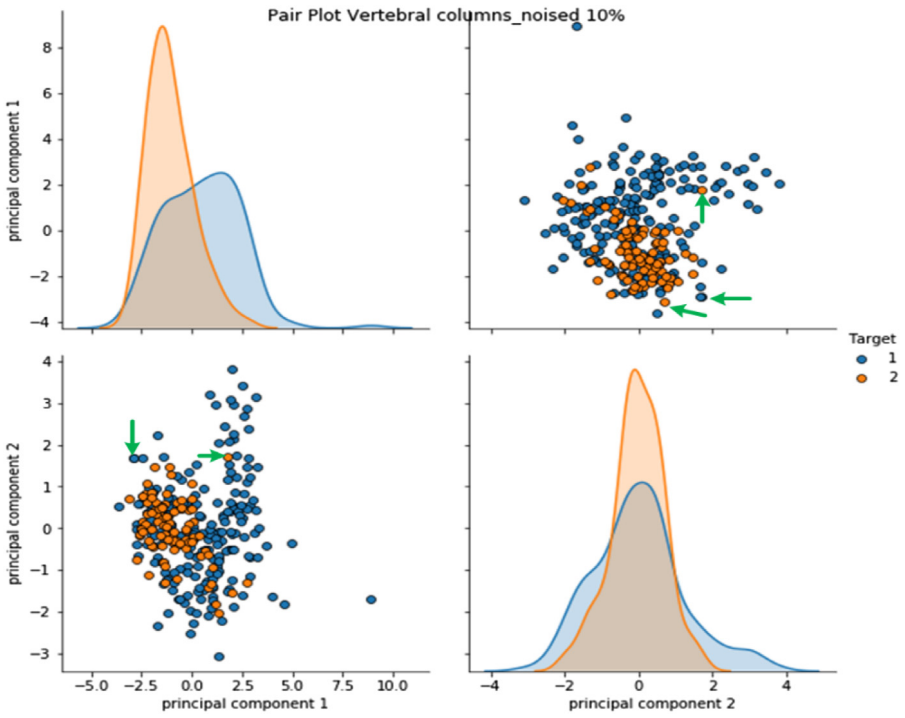

Fig. 6. Visualizing the distribution of the original and noisy samples (see the green arrows) of the Vrtbc dataset.

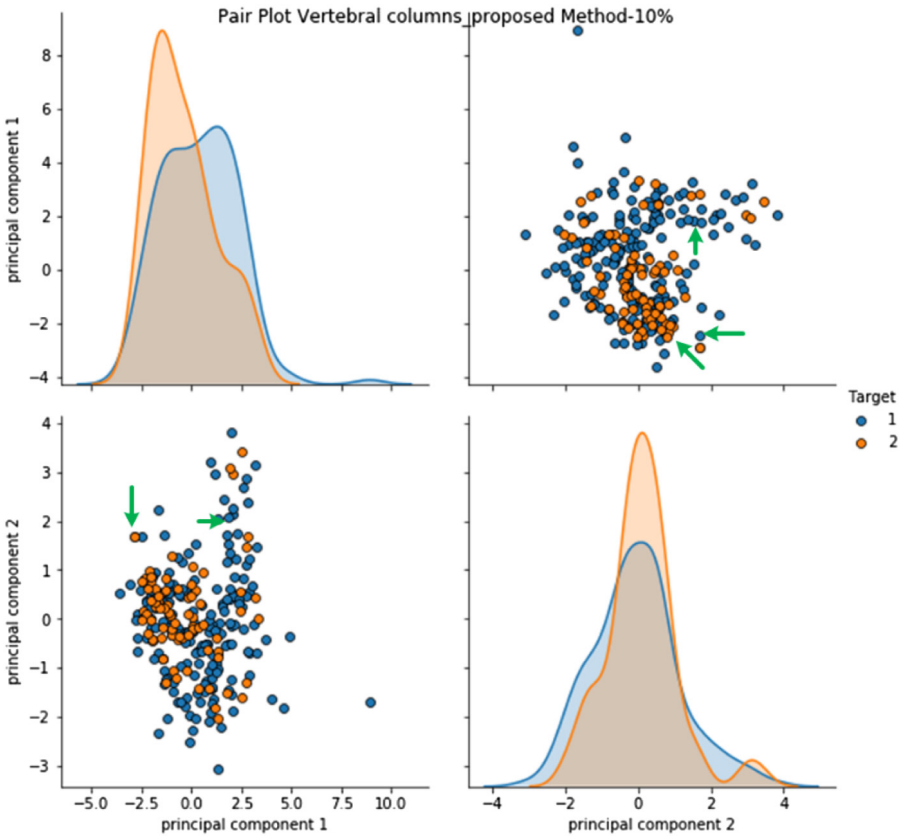

Fig. 7. Visualizing the original and recognized weak noisy samples (see the green arrows) of the Vrtbc dataset.

the proposed method is its ability to keep some regular data in the dataset using relabeling. For example, these data are frequently incorrectly detected and removed as noisy by MF. As reported in Table 3, the number of samples identified and removed as noisy using the proposed method is close to the actual number of noisy samples (for datasets with both 10% and 15% of noise). The MF method usually overestimates the level of class noise and incorrectly detects some non-noise samples as noise. Removing many correct instances can be very harmful especially when the dataset contains a low amount of samples. Although CF removes fewer samples from the data, it has difficulty to detect real noisy instances.

5.4. Future work directions

There are some future work directions that could be pursued to deal with the disadvantages of the proposed method. As mentioned earlier, we applied our model to binary classification data. However, in the future, we plan to modify it and apply it to the data containing more than two classes. We will explore the data coming from the following popular areas: health care [48–53], business planning [54], intelligent information technology (IT) [55], laser induced breakdown spectroscopy (LIBS) [56], etc.

6. Conclusions

This paper describes a new filtering method, High Agreement Voting Filtering (HAVF) using mixed strategy, to deal with the data misclassification problem. Our method applies a mixed strategy that consists of removing and relabeling noisy instances in order to improve the performance of the MF and CF methods through preventing regular data removal. A comprehensive simulation study has been carried out to compare the performances of four noise detection methods discussed in this paper. Our new method is able to identify the mislabeled instances (strong noisy and semi-strong noisy instances) that are likely to be noisy and correct the instances that are less likely to be noisy instances (weak noisy data). Thus, weak noisy data could be re-categorized instead of being removed. The mixed strategy has been applied to prevent the wrong removal of many regular instances incorrectly. Our experiments, conducted on 16 well-known datasets, showed the superiority of the proposed method over the existing approaches. To conduct our comparison, we used the following evaluation metrics: accuracy, specificity, sensitivity, STD, and AUC, which have been assessed using bagging

Table 9
The comparison of the proposed method with the existing approaches in terms of the accuracy.

Datasets	Study	Method	Accuracy
Dataset 1: Prk	Donghai Guan et al. (2014) [4]	MF_{MF}	0.7900
	Saba Bashir et al. (2016) [31]	k-means clustering	0.8923
	This paper	**Proposed method**	**0.9313**
Dataset 2: Ion	Donghai Guan et al. (2013) [30]	CF_MF	0.811
	Joaquín Abellán et al. (2012) [45]	B-CDT	0.9135
	Piyasak Jeatrakul (2012) [46]	CMTNN cleaningtechnique II	0.92
	XI-ZHAO WANG et al. (2008) [47]	NR-MCS	0.9142
	Chaoqun Li et al. (2016) [12]	IPF	0.8687
	This paper	**Proposed method**	**0.9535**
Dataset 3: BCWD	**Donghai guan et al. (2014)** [4]	**CF_1**	**0.971**
	Donghai guan et al. (2013) [30]	**MF_MF**	**0.971**
	Saba Bashir et al. (2016) [31]	k-means clustering	0.9671
	This paper	Proposed method	0.9567
Dataset 4: Sonar	Donghai guan et al. (2013) [30]	CF_MF	0.752
	Chaoqun Li et al. (2016) [12]	IPF	0.6257
	Joaquín Abellán et al. (2012) [45]	B-CDT	0.7699
	This paper	**Proposed method**	**0.8812**
Dataset 5: SHrt	Donghai Guan et al. (2014) [4]	MF_{MF}	0.823
	Saba Bashir et al. (2016) [31]	k-means clustering	0.8449
	Chaoqun Li et al. (2016) [12]	IPF	0.763
	This paper	**Proposed method**	**0.8809**
Dataset 6: Pdib	Saba Bashir et al. (2016) [31]	k-means clustering	0.7708
	Joaquín Abellán et al. (2012) [45]	B-CDT	0.7584
	XI-ZHAO WANG et al. [47]	NR-MCS	0.7114
	Piyasak Jeatrakul (2012) [46]	CMTNN cleaningtechnique II	0.7662
	This paper	**Proposed method**	**0.8819**
Dataset 7: Hpt	Saba Bashir et al. (2016) [31]	k-means clustering	Classifier -DT-IG0.8129
	Joaquín Abellán et al. (2012) [45]	B-CDT	0.8264
	Chaoqun Li et al. (2016) [12]	IPF	0.6257
	This paper	**Proposed method**	**0.8627**
Dataset 8: Vot	Joaquín Abellán et al. (2012) [45]	B-CDT	0.9556
	This paper	**Proposed method**	**0.9771**
Dataset 9: Hbrs	XI-ZHAO WANG et al. [47]	NR-MCS	0.7464
	This paper	**Proposed method**	**0.8738**
Dataset 10: Diabetes	Donghai Guan et al. (2014) [4]	MF_{MF}	**0.7850**
	Donghai Guan et al. (2014) [4]	CF_1	**0.7780**
	This paper	Proposed **method**	**0.8121**

Table 10
Number of times (out of 16) when each of the methods compared (Proposed method, MF, HAVF, and CF) provided the best result according to a specific evaluation criterion (accuracy, sensitivity, specificity, STD, and AUC).

Ranking in terms of criteria	10% of noise				15% of noise			
	Proposed method	MF	HAVF	CF	Proposed method	MF	HAVF	CF
The best Accuracy	**11**	5	0	0	**9**	7	0	0
The best Sensitivity	**8**	4	3	1	**8**	7	0	1
The best Specificity	**10**	6	0	0	**9**	6	1	0
The best STD	**9**	4	1	2	**9**	5	2	0
The best AUC	**9**	4	2	1	**8**	6	2	0

classification. Our experimental results reveal that the proposed method, HAVF using mixed strategy, outperformed HAVF using removing strategy in the presence of 10% of noise for 16, 15, 11, 10, and 13 (out of 16 in all cases) datasets according to the accuracy, specificity, sensitivity, STD and AUC criteria, respectively. Likewise, the proposed method outperformed MF according to the accuracy, specificity, sensitivity, STD and AUC criteria, for 11, 10, 11, 10, and 10 datasets, respectively in the presence of 10% of noise. Experimental results also show that CF is the weakest of all the methods compared. Clearly, CF remains a better method for datasets in which significant mislabeling occurs. Likewise, in the case of 15% noise data, the proposed method outperformed MF for 9, 10, 9, 10, and 9 datasets according to the accuracy, specificity, sensitivity, STD, and AUC criteria, respectively. Moreover, the proposed method provided the best results for 13 datasets for all metrics, including the accuracy, specificity, sensitivity, STD, and AUC, in comparison with HAVF using data removal strategy. The proposed method was also superior to MF even in the presence of 15% of noise data. However, the number of datasets for which our new method was able to generate the best results decreased compared to the case of 15% of noise data. Although the performance of the proposed method slightly decreases as the class noise level increases, the

HAVF approach with mixed strategy provides very promising results by preventing many regular instances from being removed from original data.

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.

References

[1] K. Sim, Jinyan Li, Vivekanand Gopalkrishnan, Guimei Liu, Mining maximal quasi-bicliques: Novel algorithm and applications in the stock market and protein networks, Stat. Anal. Data Min. 2 (4) (2009) 255–273.
[2] A. Zimek, E. Schubert, H.P. Kriegel, A survey on unsupervised outlier detection in high-dimensional numerical data, Stat. Anal. Data Min. 5 (5) (2012) 363–387.
[3] B. Sluban, N. Lavrač, Relating ensemble diversity and performance: A study in class noise detection, Neurocomputing 160 (2015) 120–131.
[4] D. Guan, et al., Detecting potential labeling errors for bioinformatics by multiple voting, Knowl.-Based Syst. 66 (2014) 28–35.
[5] L.P. Garcia, A.C. de Carvalho, A.C. Lorena, Noise detection in the meta-learning level, Neurocomputing 176 (2016) 14–25.
[6] S. Lee, Hyejin Shin, Sang Han Lee, Label-noise resistant logistic regression for functional data classification with an application to Alzheimer's disease study, Biometrics 72 (4) (2016) 1325–1335.
[7] K. Das, Kanishka Bhaduri, Petr Votava, Distributed anomaly detection using 1-class SVM for vertically partitioned data, Stat. Anal. Data Min. 4 (4) (2011) 393–406.
[8] C. Catal, O. Alan, K. Balkan, Class noise detection based on software metrics and ROC curves, Inform. Sci. 181 (21) (2011) 4867–4877.
[9] D. García-Gil, et al., Enabling smart data: noise filtering in big data classification, Inform. Sci. 479 (2019) 135–152.
[10] B. Sluban, D. Gamberger, N. Lavrač, Ensemble-based noise detection: noise ranking and visual performance evaluation, Data Min. Knowl. Discov. 28 (2) (2014) 265–303.
[11] B. Frénay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[12] S. Verbaeten, A. Van Assche, Ensemble methods for noise elimination in classification problems, in: International Workshop on Multiple Classifier Systems, Springer, 2003.
[13] T.W. Liao, Classification of weld flaws with imbalanced class data, Expert Syst. Appl. 35 (3) (2008) 1041–1052.
[14] M. Sabzevari, G. Martínez-Muñoz, A. Suárez, A two-stage ensemble method for the detection of class-label noise, Neurocomputing 275 (2018) 2374–2383.
[15] P. Jeatrakul, K.W. Wong, C.C. Fung, Data cleaning for classification using misclassification analysis, J. Adv. Comput. Intell. Intell. Inform. 14 (3) (2010) 297–302.
[16] C. Li, et al., Noise filtering to improve data and model quality for crowdsourcing, Knowl.-Based Syst. 107 (2016) 96–103.
[17] D. Gamberger, N. Lavrac, S. Dzeroski, Noise detection and elimination in data preprocessing: experiments in medical domains, Appl. Artif. Intell. 14 (2) (2000) 205–223.
[18] J.A. Sáez, B. Krawczyk, M. Woźniak, On the influence of class noise in medical data classification: Treatment using noise filtering methods, Appl. Artif. Intell. 30 (6) (2016) 590–609.
[19] D. Guan, W. Yuan, A survey of mislabeled training data detection techniques for pattern classification, IETE Tech. Rev. 30 (6) (2013) 524–530.
[20] J. Kanda, Andre Carvalho, Eduardo Hruschka, Carlos Soares, Selection of algorithms to solve traveling salesman problems using meta-learning, Int. J. Hybrid Intell. Syst. 8 (2011) 117–128, (Feature and algorithm selection with Hybrid Intelligent Techniques).
[21] A.L.D. Rossi, A.C. Carvalho, C. Soares, Meta-learning for periodic algorithm selection in time-changing data, in: Neural Networks, SBRN, 2012 Brazilian Symposium on, IEEE, 2012.
[22] B. Nicholson, V.S. Sheng, J. Zhang, Label noise correction and application in crowdsourcing, Expert Syst. Appl. 66 (2016) 149–162.
[23] P. Zhang, et al., Robust ensemble learning for mining noisy data streams, Decis. Support Syst. 50 (2) (2011) 469–479.
[24] J.A. Sáez, et al., Inffc: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control, Inf. Fusion 27 (2016) 19–32.
[25] J. Cheng, et al., Learning with bounded instance-and label-dependent label noise, 2017, arXiv preprint arXiv:1709.03768.
[26] J. Xiao, SVM and KNN ensemble learning for traffic incident detection, Physica A 517 (2019) 29–35.
[27] F. Shen, et al., A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation, Physica A 526 (2019) 121073.
[28] M. Samami, Binary classification of lupus scientific articles applying deep ensemble model on text data, in: 2019 Seventh International Conference on Digital Information Processing and Communications, ICDIPC, IEEE, 2019.
[29] D. Guan, et al., Identifying mislabeled training data with the aid of unlabeled data, Appl. Intell. 35 (3) (2011) 345–358.
[30] D. Guan, W. Yuan, L. Shen, Class noise detection by multiple voting, in: Natural Computation, ICNC, 2013 Ninth International Conference on, IEEE, 2013.
[31] S. Bashir, et al., HMV: a medical decision support framework using multi-layer classifiers for disease prediction, J. Comput. Sci. 13 (2016) 10–25.
[32] J. Van Hulse, T.M. Khoshgoftaar, A comprehensive empirical evaluation of missing value imputation in noisy software measurement data, J. Syst. Softw. 81 (5) (2008) 691–708.
[33] A. Folleco, et al., Software quality modeling: The impact of class noise on the random forest classifier, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008.
[34] J. Thongkam, et al., Support vector machine for outlier detection in breast cancer survivability prediction, in: Asia-Pacific Web Conference, Springer, 2008.
[35] S. Fefilatyev, et al., Label-noise reduction with support vector machines, in: Proceedings of the 21st International Conference on Pattern Recognition, ICPR2012, IEEE, 2012.
[36] A.L. Miranda, et al., Use of classification algorithms in noise detection and elimination, in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2009.
[37] R.C. de Amorim, V. Makarenkov, B. Mirkin, Core clustering as a tool for tackling noise in cluster labels, J. Classification (2019) http://dx.doi.org/10.1007/s00357-019-9303-4.
[38] S. Lessmann, et al., Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research, European J. Oper. Res. 247 (1) (2015) 124–136.
[39] M.-J. Kim, S.-H. Min, I. Han, An evolutionary approach to the combination of multiple classifiers to predict a stock price index, Expert Syst. Appl. 31 (2) (2006) 241–247.

[40] C.-F. Tsai, Y.-C. Hsiao, Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (1) (2010) 258–269.
[41] K. Veropoulos, C. Campbell, N. Cristianini, Controlling the sensitivity of support vector machines, in: Proceedings of the International Joint Conference on AI, 1999.
[42] M. Hassoon, et al., Rule optimization of boosted c5. 0 classification using genetic algorithm for liver disease prediction, in: 2017 International Conference on Computer and Applications, ICCA, IEEE, 2017.
[43] A. Luque, et al., The impact of class imbalance in classification performance metrics based on the binary confusion matrix, Pattern Recognit. 91 (2019) 216–231.
[44] K. Hajian-Tilaki, Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation, Casp. J. Intern. Med. 4 (2) (2013) 627.
[45] J. Abellán, A.R. Masegosa, Bagging schemes on the presence of class noise in classification, Expert Syst. Appl. 39 (8) (2012) 6827–6837.
[46] P. Jeatrakul, Enhancing Classification Performance over Noise and Imbalanced Data Problems, Murdoch University, 2012.
[47] X.-Z. Wang, et al., NRMCS: Noise removing based on the MCS, in: 2008 International Conference on Machine Learning and Cybernetics, IEEE, 2008.
[48] M.R. Ogiela, R. Tadeusiewicz, Nonlinear processing and semantic content analysis in medical imaging-a cognitive approach, IEEE Trans. Instrum. Meas. 54 (6) (2005) 2149–2155.
[49] M.R. Ogiela, R. Tadeusiewicz, Nonlinear processing and semantic content analysis in medical imaging, in: IEEE International Symposium on Intelligent Signal Processing, 2003, IEEE, 2003.
[50] P. Pławiak, U.R. Acharya, Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals, Neural Comput. Appl. (2019) 1–25.
[51] M. Abdar, V. Makarenkov, CWV-BANN-SVM ensemble learning classifier for an accurate diagnosis of breast cancer, Measurement 146 (2019) 557–570.
[52] M. Abdar, et al., IAPSO-AIRS: A novel improved machine learning-based system for wart disease treatment, J. Med. Syst. 43 (7) (2019) 220.
[53] W. Książek, et al., A novel machine learning approach for early detection of hepatocellular carcinoma patients, Cogn. Syst. Res. 54 (2019) 116–127.
[54] R. Tadeusiewicz, L. Ogiela, M.R. Ogiela, Cognitive analysis techniques in business planning and decision support systems, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2006.
[55] L. Ogiela, R. Tadeusiewicz, M.R. Ogiela, Cognitive analysis in diagnostic DSS-type IT systems, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2006.
[56] K. Rzecki, et al., Application of computational intelligence methods for the automated identification of paper-ink samples based on LIBS, Sensors 18 (11) (2018) 3670.