International Journal of Hospital Research 2023, 12(1)
http://ijhr.iums.ac.ir
Research Article
Dol: *LBL_COMMENTED_AT/ijhr.2023.383118.1535*

A Promising Method for Correcting Class Noise in the Presence of Attribute Noise

Akram Nakhaei[1], Mohammad Mehdi Sepehri[1*], Toktam Khatibi[1]

[1]Faculty of Industrial and Systems Engineering, Tarbiat Modares University, Tehran, Iran.

Abstract

Background and Objective: Noise is a critical concern for practical machine learning, especially medical applications. There exist two kinds of noise, including attributes and class noises. Class noise is potentially more dangerous, so various filtering techniques, particularly prediction-based, have been proposed to control it. Great attention to class noise has made the researchers ignorant that attribute noise, in turn, is harmful. Hence, it is improper to utilize prediction-based filtering to correct class noise without regarding attribute noise.

Method: To tackle this problem, we developed a method to fix class noise in the presence of attribute noise. This method excludes noisy components of attributes, based on the information bottleneck principle, by compressing attributes locally and gradually in successive iterations. It uses heterogeneous ensemble filtering to correct class noise. In the initial iteration, filtering is conservative and progressively, in succeeding iterations, tends to majority vote.

Results: We compared the proposed method's predictive performance with the RF majority-vote filter on three real binary classification problems from the UCI repository, including Breast, Transfusion, and Ionosphere. Random forest, adaptive boosting, support vector machines, and naïve Bayes were used for assessing methods from different viewpoints. Results show that the proposed method performed better than the RF majority-vote filter and seems to open a promising research scope for noise filtering.

Conclusion: Our study revealed that correcting class noise by controlling attribute noise enhances the predictive performance of classifiers.

Keywords: Inductive inference, Class noise, Attribute noise, Information bottleneck principle.

Background and Objective

Most of the real-world datasets suffer from noise, which negatively influences the formation of a generalizable hypothesis and degrades the predictive performance of induced classifiers[1-5]. So, noise is a critical issue for practical machine learning and must be handled[6,7]. There are two general approaches for this purpose, including noise-tolerant learning and filtering methods[8]. Noise-tolerant learning algorithms have integrated techniques to improve their learning abilities from noisy data. Yet, noise can still introduce severe negative impacts. The filtering approach has the advantage that noise is restrained before modeling and will not influence the learning process, making the induced hypothesis less complex and more reliable[9].

In classification tasks, data consists of two information parts, including attributes and class labels—noise mounts on both legs, which are called attribute noise and class noise, respectively[6]. Literature has shown that class noise is potentially more harmful than attribute noise[6,8,10].

*Corresponding Author: Mohammad Mehdi Sepehri
Email: mehdi.sepehri@gmail.com

So, many filtering techniques, notably prediction-based, have been introduced to identify and handle class noise. Huge attention to class noise has made the researchers ignorant of the fact that attribute noise, in its place, is harmful. Some studies have shown that attribute noise reduces the prediction performance of classifiers, and managing it will improve classification performance[8,10]. To tackle this problem, we extended a method for correcting class noise in the presence of attribute noise using compression and ensemble filtering. The proposed method excludes noisy parts of attributes, based on the information bottleneck principle, by compressing them based on the attribute-class relationship. It uses ensemble filtering to correct class noise.

This paper details the method formed to handle class noise in the presence of attribute noise. In section II, we review the past related works, highlight the research gap, and specify this study's contribution. Section III details the method and techniques used for excluding attribute noise and correcting class noise. In section IV, we present data sets used for evaluation and experimental settings. The results are displayed in section V. In section VI, we discuss the measurements, and in section VII, we highlight the critical findings.

There are two sorts of noise, including attribute noise and class noise. Attribute noise adds a small Gaussian noise to the attributes' values, and class noise changes the observed labels of samples. Former studies have shown that class noise is possibly more harmful than attribute noise. This is true because (1) there are many attributes, but only one class, and (2) the influence of each attribute is different, but the class always has a significant impact on learning[6,8,11,12]. Due to the great importance of class noise, extended research has been carried out, which can generally fall into noise-tolerant algorithms and filtering methods.

Noise-tolerant algorithms are divided into two groups. One group comprises algorithms that are naturally resistant to class noise by avoiding overfitting[6,13-15]. The other group includes algorithms that model class noise as they learn. These algorithms separate the classification model from the class noise model. To this end, they require information about the nature of class noise[6,16,17]. *Filtering methods* use filters to identify noisy examples and correct/remove them before modeling[18,19]. Various methods have been proposed for filtering noise, such as *threshold-based methods*, *cluster-based methods*, and *prediction-based methods*. The noise-tolerant approach depends on the adaptation of each classification technique. Therefore, it is not generalizable to other learning algorithms. Otherwise, the filtering method is independent of the classifier used, which usually makes this approach the most popular choice[11].

Threshold-based methods identify class noise based on anomaly measures. Samples with a value beyond a decided threshold can be identified as noise[6]. The anomaly can be a degree of complexity because class noise can increase induced model complexity[20]. *Cluster-based methods* use clustering techniques for identifying noisy instances. Since clusters are formed independently from the sample label, class noise does not affect the formation of groups[21]. In this approach,

samples are clustered several times. Each cluster's members are weighted based on cases such as the cluster size and distribution of the class labels. Finally, the class of samples can be concluded based on the sum of these weights

Prediction-based methods use classifiers to recognize class noise. In this approach, mislabeled samples are regarded as class noise. For example, Jeatrakul et al. [22] trained a neural network (NN) model and removed misclassified objects. Padala et al. (Padala et al. 2018) also used a NN model to identify noisy events from data captured by image sensors. Blanziri and Melgani[23] induced a support vector machine (SVM) model on the *K* nearest neighbors of each sample to be classified and discarded items for which the prediction is unstable. Single filters may be susceptive and recognize many instances as noisy or be very restrictive and cannot locate all noisy samples. Hence, it is safer to use an ensemble of filters.

In this regard, Garcia et al.[24] compared single and ensemble filters' performance. Results showed that ensemble filtering has better functionality, making this approach a popular choice. Sáez et al.[11] combined multiple classifiers in an iterative process to promote filtering accuracy. In each iteration, filtering sensitivity was controlled by a noisy score. Zhang et al.[25] proposed an adaptive ensemble filtering for class noise correction. The proposed method divides the training data into the cleaned and noisy sets after ensemble filtering is applied. A model is then built on cleaned data and predicts the labels of noisy instances and relabel them. Finally, clean and modified sets are joined, and the process is repeated till convergence. García-Gil et al.[26] introduced two ensemble approaches, homogenous ensemble and heterogeneous ensemble, to correct class noise in big data classification. A homogeneous ensemble employs a single base classifier over a partitioning of the training set. In comparison, the heterogeneous ensemble uses different classifiers to identify noisy instances.

Ensemble votes are merged by majority vote or consensus vote[27]. If the majority of classifiers misclassify an instance, the majority vote marks the sample as noisy. In contrast, the consensus vote needs all classifiers to misclassify an instance to recognize it as noise. Majority-vote tends to remove many instances, but the consensus vote is conservative, and a small number of samples are discarded[28]. A challenge in ensemble filtering is discovering the optimal decision point. Sabzevari et al.[29] revealed that the optimal threshold depends on the data itself. They found the optimal point by cross-validation and wrapper-based attribute selection. Guan et al.[30] offered a cost-sensitive approach to finding the optimal point. They estimated the mislabeled data probability distribution, which was used to judge each possible decision point's expected cost. Finally, the decision point with the lowest cost was selected as the optimal point.

Huge attention to class noise has made the researchers unaware that attribute noise is harmful in its place. Studies have shown that attribute noise reduces classification accuracy, and handling it improves classification performance[8,10]. Hence, in

prediction-based filtering methods, it is crucial to handle the attribute noise.

To tackle this problem, we developed a method for correcting class noise in the presence of attribute noise. Our proposed method differs from previous prediction-based filtering methods in the:

- It corrects class labels in the presence of attribute noise.
- It employs the information bottleneck (IB) principle for handling attribute noise.

Method

This section first describes the approach to eliminating attribute noise, then explains the working method for correcting class noise, and finally presents the proposed algorithmic view of the method.

Eliminating attribute noise

Data consists of the original signal and the mounting noise. According to study of Priemer[31] and Tuzlukov[32], the original signal is a function that carries information about a special phenom, and noise means a random signal that carries no useful information about that phenom. If we count the class as a phenom understudy, components of attributes containing no information about class labels can be regarded as attribute noise. Hence, by relying on relevance, we can drop noisy parts of attributes by compression.

For the first time, we used the information bottleneck principle, iteratively, to find a compressed form of attributes that is most informative about the class. This principle, which is proposed by Tishby et al.[33], provides an information-theoretic method for extracting relevant parts of an input variable X, regarding an output variable Y. This extraction is done by finding a compressed representation T of X that is most informative about Y[34,35]. For compression, we used the principal component analysis (PCA) technique. This technique simplifies the complexity of data while maintaining trends and patterns. It does this by mapping the data into fewer dimensions, which are summaries of attributes[36].

Correcting class noise

Due to ensemble filtering's excellent performance[24], we used this technique to correct class labels. This method combines a set of base-level classifiers to build a new classifier that is usually more accurate than any of its components[9]. Based on research of Verbaeten and Van Assche[9], the general scheme of ensemble filtering is as follows:

- n classifiers are induced on different subsets of the training data.
- These classifiers predict training samples' labels.
- For each sample, the filter compares the predicted labels with the original tags and decides whether it is noisy or not.

We used the decision tree (DT), SVM, and naïve Bayes (NB) models as base classifiers. To induce the filter, we partitioned the training set into ten equal folds. The ensemble filter was trained ten times, each time leaving out one of the folds, and then the trained filter was used to predict the labels of the removed fold. The algorithm was designed to be iterative. In the first iteration, the combination of votes is conservative

(consensus vote). In subsequent iterations, it gradually tends to the majority vote.

Proposed algorithm

Since both attribute and class carry noise, it is not proper to rely entirely on original attribute values to correct all class labels, or conversely, to rely entirely on original class labels to remove all noisy components of attributes. Hence, we designed the proposed algorithm as iterative and gradual. In each iteration, we first lightly compress the attributes, based on their relationship with the class labels, and then modify class labels by ensemble filtering, based on the renewed attribute set (Fig 1). In the first iteration, the compression degree is low, and the vote combination is conservative (consensus vote). In subsequent iterations, the compression level slowly increases, and the vote combination gradually tends to the majority vote.

Input:
- Attributes (A) and class labels (L)

Output:
- Compressed attributes (A') and modified class labels (L')

Initialization:
- $A' = A$
- $L' = L$

Loop:

$While\ (\frac{sum(L' =!\ L)}{|L|} \geq 0.1)$
- $L = L'$
- $A = A'$
- $\forall\ i, j, k = 1 \ldots |A|, i < j < k,$ calculate:
- $d_{ijk} = I(PC1_{ijk}, L) + I(PC2_{ijk}, L) - I(A_i, L) - I(A_j, L) - I(A_k, L)$
- Find indices i, j, k for which d_{ijk} is minimized
- Merge $A_i, A_j, A_k -> PC1_{ijk}, PC2_{ijk}$
- Update $A' = \{A - \{A_i, A_j, A_k\}\} \cup \{PC1_{ijk}, PC2_{ijk}\}$
- Ensemble filtering on A' and calculate new class labels (L')

Figure 1- Algorithmic view of the proposed *method*.

A fixed amount of attributes are picked and compressed in each repetition. This choice is performed greedily based on the least waste of mutual information between compressed attributes and class labels. Mutual information loss after compression is computed using (1). The number of compressed attributes is considered three in this formula, although it is generalizable to a bigger number. Here, PC means the principal component. Mutual information between two variables is measured using (2).

$$d_{ijk} = I(PC1_{ijk}, L) + I(PC2_{ijk}, L) - I(A_i, L) - I(A_j, L) - I(A_k, L)) \quad (1)$$

$$I(A_i, L) = \sum_{a_i \in A_i} \sum_{l \in L} p(a_i, l) * log(\frac{p(a_i,l)}{p(a_i)p(l)}) \quad (2)$$

After compression, we modify class labels by prediction-based ensemble filtering on the renewed attribute set. This algorithm is repeated till convergence. Convergence happens when the amount of change in class labels drops below a threshold, which is considered 10% in this study.

Experimental settings

This section describes datasets, the mechanism for adding noise, the number of attributes to be compressed, parameters of ensemble filtering, models, and evaluation metric.

Datasets

We assess the proposed method on three real binary classification problems from the UCI repository (Dua and Graff, 2019), including two medical and one nonmedical dataset. Table I presents the properties of these datasets. We split each dataset into train and test sets, with a ratio of 70 to 30. Next, we add 0, 5,10, 15, and 20% noise to both its class labels and attribute values for each train set.

Table I- Characteristics of the classification problems used for evaluation

Dataset	Train	Test	Number of attributes
Breast	488	210	8
Transfusion	523	225	4
Ionosphere	244	106	34

Adding noise

To add noise, we randomly selected data elements, including attribute values and class labels. For each chosen piece, if it was a class label, we inverted its value. If it was an attribute, we added to its value a random number in the range (-SD, SD) of that attribute column.

1. Number of attributes to be compressed

We just considered datasets with equal or more than three attributes. The number of attributes to be compressed for each data set was computed based on (3). This number is the square root of the dimension of each data set. If the data set holds more than three attributes, and its square root is less than three, we considered it three.

$$\begin{cases} if\ \sqrt[2]{|A|} \leq 3 & number \leftarrow 3 \\ else & number \leftarrow ceil(\sqrt[2]{|A|}) \end{cases} \quad (3)$$

2. Ensemble filtering

Ninety-six models, including 32 DT, 32 SVM, and 32 NB models, were induced. The train set was randomly divided into ten equal folds. We took each fold out and trained the models on the remaining parts. Then, the trained models were used to label samples of the departed fold.

Ensemble classifiers are induced from the different bags of the remaining parts. Thresholds of 0.8, 0.7, 0.6, and 0.5 are considered for aggregating ensemble votes. In the algorithm's first iteration, the threshold is considered 0.8, the second iteration, 0.7, the third 0.6, and the fourth and after 0.5. These values are considered by default for the initial evaluation of the proposed algorithm and are not optimal.

Models

We considered several techniques for evaluation, including Bagging, Boosting, SVM, and NB. For Bagging, we used the random forest (RF) model consisting of 500 trees. In this model, the splitting criterion was the Gini index, and the Grid search determined the number of variables to be tried at each split. For Boosting, we used the adaptive boosting model (Adaboost). The Grid search determined the number of AdaBoost iterations. We selected the polynomial kernel as the SVM's kernel function, and the Grid search determined the degree and scale parameters of the SVM.

Evaluation metric

We selected F-measure[4,5,6] to evaluate the proposed method's predictive performance. TP means true positive, FP false positive, TN true negative, and FN true negative.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F-measure = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

Results

In this section, we present and discuss the predictive performance of the proposed method. Since RF has high stability and performance in different noise levels[29,37,38], we compared the proposed method with an RF majority-vote filter (RF-MV-F). The reported results are average, followed by the standard deviation (SD), over five different executions. Different random partitioning of the data into training and test sets was used in these executions.

Predictive performance

Table II shows the predictive performance of methods on three data sets and different noise levels. Table III shows the average performance on all data sets.

Table II- Predictive performance of models on different methods, different datasets, and different noise levels

			Noisy data	Proposed method	RF-MV-F
Models	dataset	Noise percent	F-measure	F-measure	F-measure
RF	Breast	0	95.8 ± 2.8	96.2 ± 2.8	95.5 ± 2.8
		5	95.5 ± 2.9	95.9 ± 2.6	94.9 ± 2.6
		10	95.4 ± 2.5	95.9 ± 2.1	95 ± 2.8
		15	94.2 ± 3.4	95.8 ± 2.2	95 ± 2.9
		20	93.6 ± 3.3	95.2 ± 2.5	94.9 ± 2.7
	Transfusion	0	46 ± 4.7	48 ± 4.7	49.9 ± 6.4
		5	46.1 ± 4.7	48.1 ± 3.3	47.6 ± 4.9
		10	44.5 ± 6.5	47.9 ± 3.6	47.6 ± 4.2
		15	44.4 ± 7.1	48.2 ± 3.4	46.6 ± 4.4
		20	44.7 ± 3	48.4 ± 3.4	44.2 ± 5.2
	Ionosphere	0	92 ± 3.6	90.7 ± 3	87.3 ± 1.2
		5	90 ± 1.8	89.3 ± 2.7	88.2 ± 2.3
		10	89.8 ± 3.1	89.2 ± 1.9	87.6 ± 2.9
		15	88.8 ± 4.6	82.1 ± 6.5	85.1 ± 7.8
		20	87.4 ± 6.5	85.4 ± 3.3	82.5 ± 5

			Noisy data	Proposed method	RF-MV-F
Models	dataset	Noise percent	F-measure	F-measure	F-measure
Adaboost	Breast	0	95.3 ± 2.3	95.2 ± 3	94.2 ± 2.6
		5	92.1 ± 2.7	95.4 ± 2.7	94.9 ± 3.4
		10	89.7 ± 4.2	96 ± 2.9	94.9 ± 2.7
		15	87 ± 3	95.6 ± 2.2	94.9 ± 3.1
		20	83.8 ± 4.5	95.4 ± 2	95.1 ± 2.9
	Transfusion	0	46.5 ± 4	50.4 ± 4	49.6 ± 7.2
		5	45.4 ± 4.3	48.3 ± 3.2	46 ± 7
		10	44.1 ± 4.9	47.4 ± 2.5	44.7 ± 5.6
		15	45.6 ± 7.3	48.8 ± 3	43.8 ± 5.8
		20	42.5 ± 2.3	48.6 ± 5	42.4 ± 6
	Ionosphere	0	89.6 ± 4.4	91 ± 3.2	87.4 ± 2.8
		5	85.6 ± 1.3	89.3 ± 2.9	88.6 ± 3.4
		10	87.2 ± 2.8	89.6 ± 1.7	86.1 ± 3.9
		15	84.2 ± 5.6	84.6 ± 4.6	85.6 ± 5.3
		20	83.4 ± 3.2	85.6 ± 2.7	82.5 ± 4.7
SVMs	Breast	0	95.4 ± 2.2	95.4 ± 2.6	95.2 ± 2.7
		5	94.9 ± 2.5	95.6 ± 2.7	95.7 ± 2.6
		10	95.5 ± 2.6	95.5 ± 2.4	95.5 ± 2.2
		15	95.2 ± 2.5	95.3 ± 2.3	95.1 ± 2.5
		20	95.6 ± 2.5	95.5 ± 2.3	95.5 ± 2.2
	Transfusion	0	49.7 ± 3.7	48.8 ± 3.9	48.7 ± 5.1
		5	49.1 ± 3.9	48.3 ± 3.9	49.8 ± 4.4
		10	48.6 ± 3.8	49 ± 3.7	47.8 ± 3.3
		15	47.9 ± 3.8	49.3 ± 4.9	50 ± 6.5
		20	49.7 ± 4.9	47 ± 5	48.3 ± 5.5
	Ionosphere	0	87.6 ± 2.4	88.5 ± 5.4	83.4 ± 5.6
		5	85.4 ± 3.1	85.4 ± 4.9	82.9 ± 3.6
		10	80.7 ± 5.5	83.5 ± 3.3	79.1 ± 4.3
		15	77 ± 3.1	80 ± 3.5	78.4 ± 5.1
		20	75.5 ± 6	78.4 ± 3.1	77.6 ± 4.5
NB	Breast	0	95.8 ± 2.4	96 ± 2.3	95.5 ± 2.2
		5	94.3 ± 2.9	95.5 ± 2.8	96 ± 2.5
		10	92.6 ± 3	95 ± 2.3	95.6 ± 2.4
		15	93.6 ± 3.4	95.4 ± 3.3	95 ± 3.3
		20	93.2 ± 2.6	95.2 ± 2.2	95 ± 2
	Transfusion	0	46.5 ± 5.2	47.3 ± 2.7	46 ± 4.8
		5	46.7 ± 3.6	47.1 ± 2.6	47.9 ± 5
		10	47.5 ± 3.2	46 ± 2.8	46 ± 2.8
		15	47.3 ± 2.8	47.1 ± 4.1	47.9 ± 4.6
		20	41.8 ±8.8	47.7 ±2.9	45.9 ±4.3
	Ionosphere	0	88.5 ±3.5	88.6 ±4.8	86.2 ±4.2
		5	85.4 ±5.4	87.5 ±3.7	86.1 ±3.4
		10	84.6 ±5.8	86 ±6.3	85.6 ±5.8
		15	82.5 ±7.6	89.8 ±3.6	83.6 ±3.7
		20	80 ±8.7	80.9 ±5.2	82.4 ±3.6

Table III- Average performance of models on different datasets

		Noisy data	Proposed method	RF-MV-F
Models	Noise Percent	F-measure	F-measure	F-measure
RF	0	77.9 ±24	78.9 ±22	77.6 ±21
	5	77.2 ±23	77.8 ±22	76.9 ±22
	10	76.6 ±24	77.6 ±22	76.7 ±22
	15	75.8 ±24	75.4 ±21	75.6 ±22
	20	74 ±23	75.4 ±22	73.1 ±23
Adaboost	0	77.1 ±23	78.9 ±21	77.1 ±21
	5	74.4 ±22	77.7 ±22	76.5 ±23
	10	73.7 ±22	77.7 ±22	75.2 ±23
	15	72.3 ±20	76.3 ±21	74.8 ±23
	20	68.7 ±21	76 ±22	72.7 ±24
SVMs	0	77.6 ±21	77.6 ±22	75.8 ±21
	5	76.5 ±21	76.4 ±21	76.1 ±21
	10	74.9 ±21	76 ±21	74.1 ±21
	15	73.4 ±20	74.8 ±20	74.5 ±20
	20	73.6 ±21	75 ±22	73.9 ±21
NB	0	76.9 ±23	77.3 ±22	75.9 ±23
	5	75.4 ±22	76.7 ±22	76.7 ±22
	10	74.9 ±21	75.7 ±22	75.7 ±22
	15	74.5 ±21	74.1 ±21	75.5 ±21
	20	71.3 ±24	73.7 ±21	73.6 ±22

Breast dataset

The performance of methods, measured through RF, AdaBoost, SVM, and NB, are represented in the following Figures. The horizontal axis represents the noise level, and the vertical axis represents the F-measure. The proposed method results are shown in blue, RF-MV-F in orange, and noisy data in grey. The distance between each point in the methods' curve and the noisy data shows improvement.

Figs 1-4 shows the performance of methods measured by different models on the Breast dataset. RF predictive measures show that the proposed method was a better filtering technique than RF-MV-F (Fig 2, Table II). For all noise levels, the proposed technique leads to a higher mean of F-measure. Furthermore, F-measure distribution had a lower SD than RF-MV-F and noisy data. A lower standard deviation shows that data points are clustered more closely to the mean. Therefore the generating source is more reliable. We can conclude that the proposed method enhanced both the RF model's predictive performance and stability.

Adaboost is very sensitive to noise. It is prone to overfitting since excessively increases the weight for noisy instances[39]. Figures show that the model performance is significantly reduced due to noise. The functional distance between the two methods is small. However, the proposed method shows a higher F-measure mean and a lower SD for almost all noise levels (Fig 3, Table II). We conclude that both methods, mostly the proposed algorithm, could enhance the Adaboost performance and stability.

The SVM model constructs a set of separating hyperplanes and selects the optimal one by maximizing the margin. SVM optimization is a nonlinear problem subject to constraints. In noisy environments, the optimization function may present many local minima that result in performance

degradation. SVM results confirm this and show that the optimization is stuck in the local optima caused by noise (Fig 4). F-measure mean and SD fluctuate so much that one method's superiority over the other is not distinguishable (Fig 4, Table II). However, the intensity of the fluctuations for the proposed method is lower than the RF-MV-F. We conclude that the proposed method refines some of the local optima created by noise.

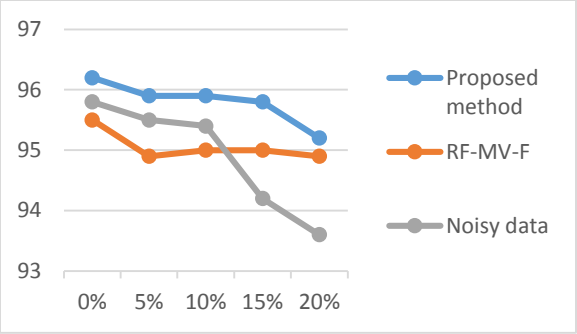

Figure 2- Performance of RF for different techniques and different noise levels of the Breast dataset.

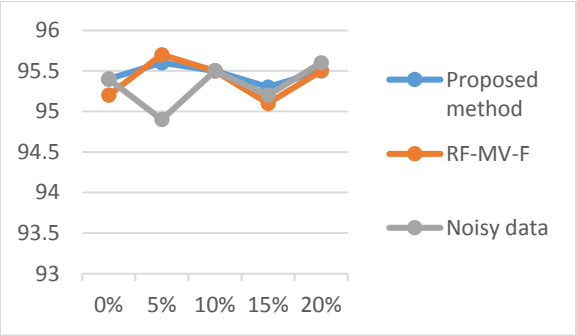

Figure 4- Performance of SVM for different techniques and different noise levels of the Breast dataset.

Transfusion dataset

Fig 6 shows that both the proposed method and RF-MV-F have improved the RF performance. As the noise level increases, the RF-MV-F performance decreases so that in 20% noise, it is placed under noisy data. In comparison to RF-MV-F, the proposed method shows consistent performance in

NB is a probability-based model. Since noise distorts the data shape, it adversely affects the NB performance. Results show that both methods reconstruct the underlying probability distributions and improve NB performance (Fig 5). RF-MV-F has a higher F-measure mean for noise levels 5 and 10%. However, as the noise level increases, its performance decreases, and the proposed method excels. We conclude that the proposed method can reconstruct the underlying distribution in high noise levels.

.

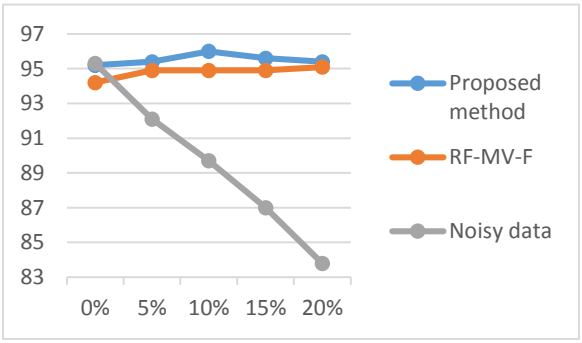

Figure 3- Performance of Adaboost for different techniques and different noise levels of the Breast dataset.

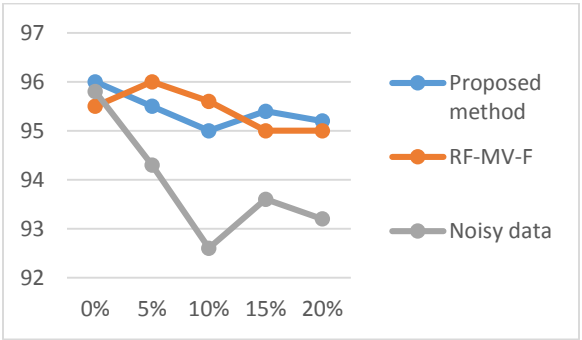

Figure 3-Performance of NB for different techniques and different noise levels of the Breast dataset.

different noise levels. Furthermore, it has a lower SD for all noise levels. We conclude that the proposed method performed better than RF-MV-F in all noise levels.

Fig 7 (AdaBoost) shows that the proposed method performed better in handling noise. It has a higher F-measure mean and lower SD

for all noise levels (Table II). SVM results show many fluctuations so the superior method is not distinguishable (Fig 8). However, generally, the proposed method has a lower SD (Table II).). Fig 9 shows fluctuations for both methods. However, as the noise level reaches 20%, the proposed method exceeds RF-MV-F. Also, in this model, the proposed method has a smaller SD for all noise levels.

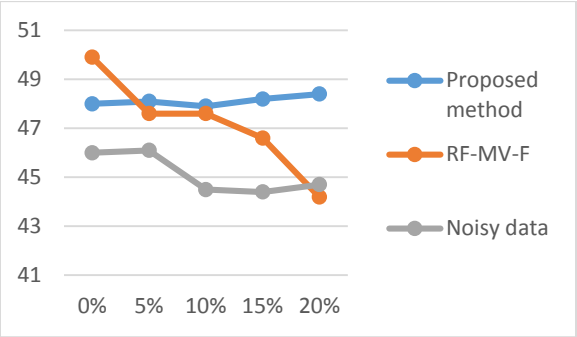

Figure 4- Performance of RF for different techniques and different noise levels of the Transfusion dataset.

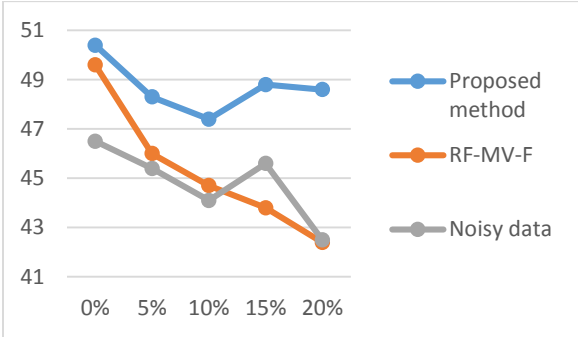

Figure 7- Performance of Adaboost for different techniques and different noise levels of the Transfusion dataset.

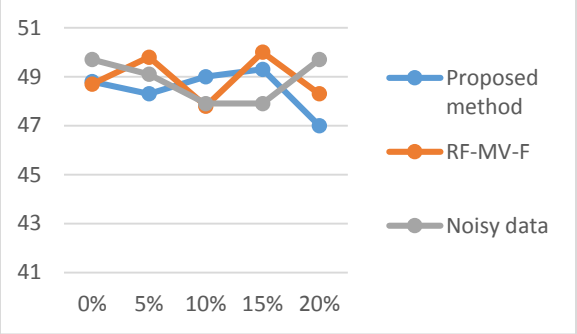

Figure 8- Performance of SVM for different techniques and different noise levels of the Transfusion dataset.

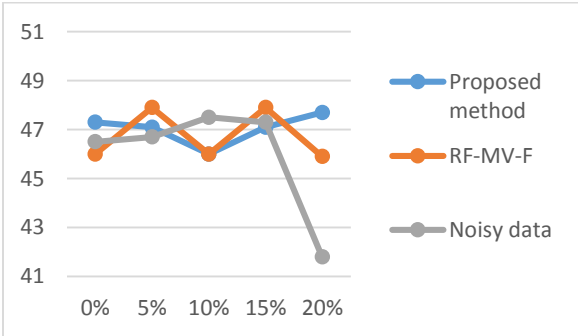

Figure 5- Performance of NB for different techniques and different noise levels of the Transfusion dataset.

Ionosphere dataset

Surprisingly, Fig 10 shows that noise handling methods could not enhance the RF performance at any noise level. Compared to each other, the proposed method has a higher F-measure mean than RF-MV-F for all noise levels except 15% noise (Fig 10). It also has a lower SD for all noise levels except 5% noise (Table II). Overall, we conclude that the proposed method is superior. For Adaboost, like RF, the proposed method has a higher mean for all noise levels, except for the noise level of 15% (Fig 11). Also, it has a smaller SD for all noise levels except a noise level of 0% (Table II). Unlike the Breast and Transfusion datasets, for the Ionosphere dataset, the SVM model showed fewer fluctuations. Maybe this happened due to the increase in data dimension. SVM results show that the proposed method has a higher mean and a lower SD for all noise levels (Fig 12, Table II). For NB, the proposed method has a higher mean for all noise levels, except for the noise level of 20% but has a higher SD for all noise levels (Fig 13, Table II).

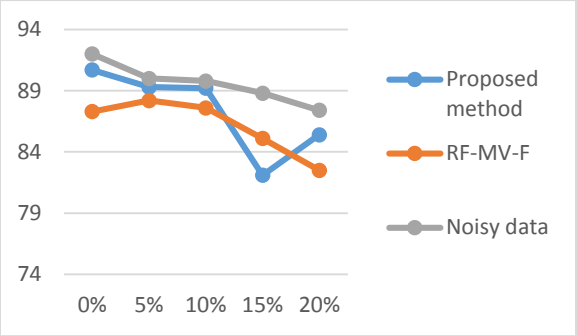

Figure 6- Performance of RF for different techniques and different noise levels of the Ionosphere dataset.

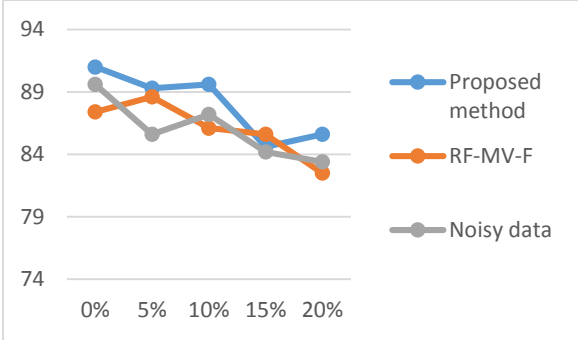

Figure 81- Performance of Adaboost for different techniques and different noise levels of the Ionosphere dataset.

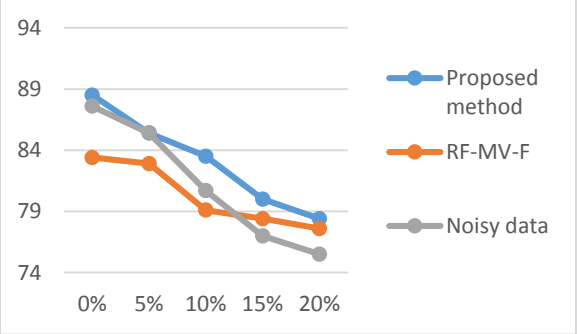

Figure 72- Performance of SVM for different techniques and different noise levels of the Ionosphere dataset.

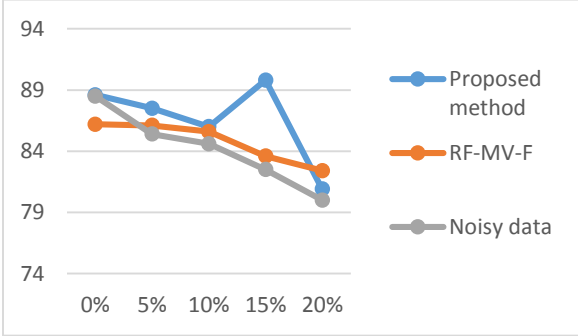

Figure 9- Performance of NB for different techniques and different noise levels of the Ionosphere dataset.

Mean results on all data sets

This section illustrates the average performance of models on all three datasets (Figs 14-17). RF results show that the proposed method has a higher F-measure mean for all noise levels except for 15% noise, which shows a slight decrease (Fig 14). The reason for this is a sharp drop in the performance for 15% noise in the Ionosphere dataset.

Fig 15 shows that for AdaBoost, the proposed method has a higher F-measure mean for all noise levels. For SVM, too (Fig 16), the proposed method has a higher F-measure average for above 5% noise. Fig 17 shows that for NB, the proposed method has a higher mean for all noise levels except for a noise level of 15%. As Table III shows, the SDs of the averaged results are the same for all methods.

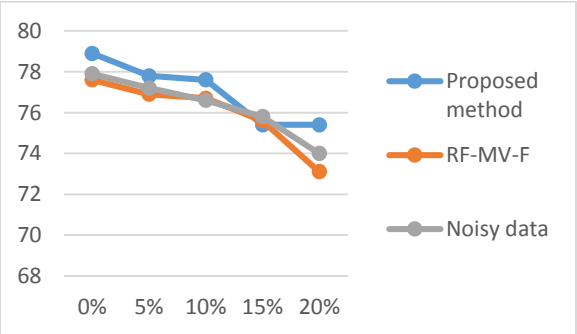

Figure 10- Mean results of RF on all datasets.

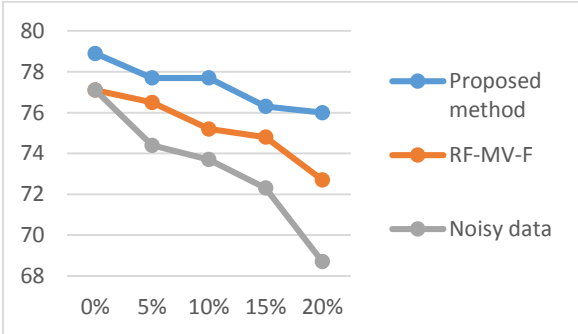

Figure 11- Mean results of AdaBoost on all datasets.

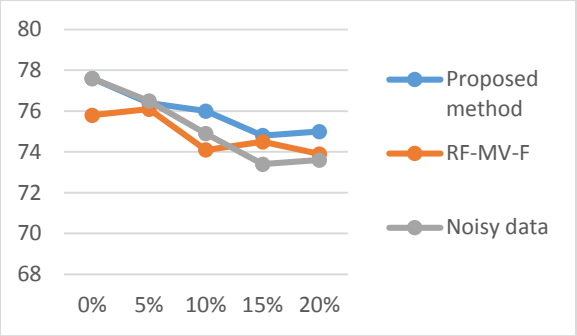

Figure 12- Mean results of SVM on all datasets.

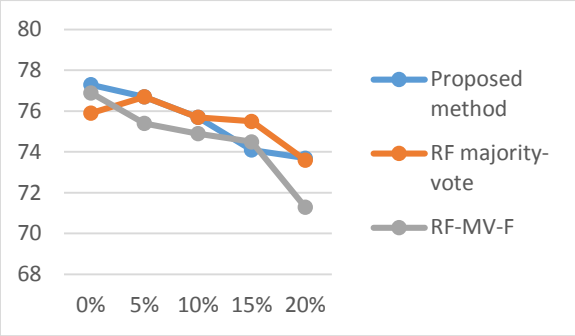

Figure 13- Mean results of NB on all datasets.

Conclusion

In this study, we have proposed an iterative method for correcting class noise in the presence of attribute noise, a challenge that had not been addressed before. The proposed method excludes noisy parts of attributes based on their mutual information with class labels. To this end, attributes are compressed using PCA locally and gradually in succeeding iterations. This method also employed heterogeneous ensemble filtering, including DT, SVM, and NB, to correct class noise. In the beginning, filtering is conservative and progressively, in subsequent iterations, tends to majority vote. We compared the proposed method's predictive performance with RF-MV-F, which has high stability and performance in different noise levels[29,37,38]. We measured the performance (F-measure) of methods through RF, AdaBoost, SVM, and NB models.

Using different models allows us to examine the performance of methods from different perspectives. RF is one of the most noise-resistant models[37]. However, in noisy environments, its performance decreases. The proposed method shows that it can enhance the RF performance in low-dimensional datasets like Breast and Transfusion. However, in both low-dimensional and high-dimensional datasets, it shows a better ability to enhance RF performance than RF-MV-F. The average performance over three datasets also shows that the proposed method almost has a higher F-measure mean than the RF-MV-F.

AdaBoost is very sensitive to noise and successively increases noisy samples' wights that lead to overfitting. Hence, AdaBoost performance significantly decreases in noisy situations. Results show that in low-dimensional and high-dimensional datasets, medical and nonmedical, the proposed method works better in raising AdaBoost performance. The average performance over three datasets shows that the proposed method has a higher F-measure mean than the RF-MV-F. By counting the AdaBoost performance, we conclude that the proposed method has an acceptable ability to eliminate/correct noise.

The SVM model is susceptible to noisy data. It tries to find an optimal hyperplane by maximizing the distance between margins[40]. This cost function is highly sensitive to noise, generating many local optima[41], especially in low-dimensional data. SVM results in Breast and Transfusion datasets show fluctuations representing cost function sensitivity to

different noise levels. The proposed method's fewer fluctuations may show that it could control the effect of noise on the cost function. As the number of attributes increases, fluctuations calm down. Ionosphere results show that the proposed method performs better than RF-MV-F for all noise levels. The average of results on three datasets also shows that the proposed method performed better than RF-MV-F.

NB is a parametric model and relies on the probability distribution of data. Since noise distorts the data shape, NB performance decreases in noisy environments. Results of low-dimensional datasets, including Breast and Transfusion, show that the proposed method performs better than RF-MV-F in high noise levels. In the Ionosphere, the proposed method has a higher performance than RF-MV-F, but its performance falls in 20% noise. The average of results on three datasets shows that the two methods have almost the same performance.

Overall, the experimental results on real-world data sets, including two medical and one nonmedical dataset, show that the proposed method has a better performance than RF-MV-F in terms of predictive performance. This method can control the effect of noise on SVM's cost function, enhance RF performance, avoid AdaBoost overfitting, and reconstruct the underlying probability distributions to improve the NB's performance. Furthermore, our study shows that correcting class noise by handling attribute noise leads to better classifiers' predictive performance.

This study has some limitations. Several issues must be investigated more carefully, like the convergence threshold, the number of compressed attributes, and the slope of the move from conservative to majority vote. We selected the default value for these cases. Optimal values could be investigated in future research.

Consent for publication

The authors have approved the final manuscript for publication.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The authors have made an equal contribution to this research.

References

1. Bellazzi R, Zupan B. Predictive data mining in clinical medicine: current issues and guidelines. Int J Med Inf. 2008;77(2):81–97.
2. Feder SL. Data quality in electronic health records research: quality domains and assessment methods. West J Nurs Res. 2018;40(5):753–66.
3. Feldman K, Faust L, Wu X, Huang C, Chawla NV. Beyond volume: the impact of complex healthcare data on the machine learning pipeline. In: Towards Integrative Machine Learning and Knowledge Extraction. Springer; 2017. p. 150–69.
4. Kim S, Zhang H, Wu R, Gong L. Dealing with noise in defect prediction. In: 2011 33rd International Conference on Software Engineering (ICSE). IEEE; 2011. p. 481–90.
5. Nazari Z, Nazari M, Danish MSS, Kang D. Evaluation of class noise impact on performance of machine learning algorithms. IJCSNS. 2018;18(8):149.

6. Frénay B, Verleysen M. Classification in the presence of label noise: a survey. IEEE Trans Neural Netw Learn Syst. 2013;25(5):845–69.
7. Naseri H, Homaeinezhad MR, Pourkhajeh H. Noise/spike detection in phonocardiogram signal as a cyclic random process with non-stationary period interval. Comput Biol Med. 2013;43(9):1205–13.
8. Zhu X, Wu X. Class noise vs. attribute noise: A quantitative study. Artif Intell Rev. 2004;22(3):177–210.
9. Verbaeten S, Van Assche A. Ensemble methods for noise elimination in classification problems. In: International Workshop on Multiple classifier systems. Springer; 2003. p. 317–25.
10. Wickramasinghe RIP. Attribute Noise, Classification Technique, and Classification Accuracy. In: Data Analytics and Decision Support for Cybersecurity. Springer; 2017. p. 201–20.
11. Sáez JA, Krawczyk B, Woźniak M. On the influence of class noise in medical data classification: Treatment using noise filtering methods. Appl Artif Intell. 2016;30(6):590–609.
12. Samami M, Akbari E, Abdar M, Plawiak P, Nematzadeh H, Basiri ME, et al. A mixed solution-based high agreement filtering method for class noise detection in binary classification. Phys Stat Mech Its Appl. 2020;124219.
13. Hopkins M, Kane D, Lovett S, Mahajan G. Noise-tolerant, reliable active classification with comparison queries. ArXiv Prepr ArXiv200105497. 2020;
14. Nguyen DT, Ngo TPN, Lou Z, Klar M, Beggel L, Brox T. Robust Learning Under Label Noise With Iterative Noise-Filtering. ArXiv Prepr ArXiv190600216. 2019;
15. Ukil A, Roy UK. Smart cardiac health management in IoT through heart sound signal analytics and robust noise filtering. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE; 2017. p. 1–5.
16. Kaneko T, Ushiku Y, Harada T. Label-noise robust generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019. p. 2467–76.
17. Yang Y, Xia Y, Chi Y, Muntz RR. Learning naive Bayes classifier from noisy data. Univ Calif Los Angel Dep Comput Sci Tech Rep CSD-TR. 2003;(030056).
18. Ichikawa K, Kawashima H, Shimada M, Adachi T, Takata T. A three-dimensional cross-directional bilateral filter for edge-preserving noise reduction of low-dose computed tomography images. Comput Biol Med. 2019;111:103353.
19. Padala V, Basu A, Orchard G. A Noise Filtering Algorithm for Event-Based Asynchronous Change Detection Image Sensors on Truenorth and Its Implementation on Truenorth. Front Neurosci. 2018;12:118.
20. Gamberger D, Lavrač N, Džeroski S. Noise elimination in inductive concept learning: A case study in medical diagnosis. In: International Workshop on Algorithmic Learning Theory. Springer; 1996. p. 199–212.
21. Nicholson B, Sheng VS, Zhang J. Label noise correction and application in crowdsourcing. Expert Syst Appl. 2016;66:149–62.
22. Jeatrakul P, Wong KW, Fung CC. Data cleaning for classification using misclassification analysis. J Adv Comput Intell Intell Inform. 2010;14(3):297–302.
23. Blanzieri E, Melgani F. An adaptive SVM nearest neighbor classifier for remotely sensed imagery. In: 2006 IEEE International Symposium on Geoscience and Remote Sensing. IEEE; 2006. p. 3931–4.
24. Garcia LP, Lorena AC, Matwin S, de Carvalho AC. Ensembles of label noise filters: a ranking approach. Data Min Knowl Discov. 2016;30(5):1192–216.
25. Zhang J, Sheng VS, Li T, Wu X. Improving crowdsourced label quality using noise correction. IEEE Trans Neural Netw Learn Syst. 2017;29(5):1675–88.
26. García-Gil D, Luengo J, García S, Herrera F. Enabling smart data: noise filtering in big data classification. Inf Sci. 2019;479:135–52.

27. Brodley CE, Friedl MA. Identifying mislabeled training data. J Artif Intell Res. 1999;11:131–67.
28. Berthelsen H, Megyesi B. Ensemble of classifiers for noise detection in PoS tagged corpora. In: International Workshop on Text, Speech and Dialogue. Springer; 2000. p. 27–32.
29. Sabzevari M, Martínez-Muñoz G, Suárez A. A two-stage ensemble method for the detection of class-label noise. Neurocomputing. 2018;275:2374–83.
30. Guan D, Hussain M, Yuan W, Khattak AM, Fahim M, Khan WA. Enhanced Label Noise Filtering with Multiple Voting. Appl Sci. 2019;9(23):5031.
31. Priemer R. Introductory signal processing. Vol. 6. World Scientific; 1991.
32. Tuzlukov V. Signal processing noise. CRC Press; 2018.
33. Tishby N, Pereira FC, Bialek W. The information bottleneck method. ArXiv Prepr Physics0004057. 2000;
34. Shamir O, Sabato S, Tishby N. Learning and generalization with the information bottleneck. Theor Comput Sci. 2010;411(29–30):2696–711.
35. Slonim N. The information bottleneck: Theory and applications [PhD Thesis]. Citeseer; 2002.
36. Lever J, Krzywinski M, Altman N. Points of significance: Principal component analysis. Nature Publishing Group; 2017.
37. Folleco A, Khoshgoftaar TM, Van Hulse J, Bullard L. 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. p. 3853–9.
38. Saki F, Sehgal A, Panahi I, Kehtarnavaz N. Smartphone-based real-time classification of noise signals using subband features and random forest classifier. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2016. p. 2204–8.
39. Gómez-Ríos A, Luengo J, Herrera F. A study on the noise label influence in boosting algorithms: AdaBoost, GBM and XGBoost. In: International Conference on Hybrid Artificial Intelligence Systems. Springer; 2017. p. 268–80.
40. Gangsar P, Tiwari R. Effect of noise on support vector machine based fault diagnosis of im using vibration and current signatures. In: MATEC Web of Conferences. EDP Sciences; 2018. p. 03009.
41. Cosme R de C, Krohling RA. Support Vector Machines applied to noisy data classification using differential evolution with local search. 2011;

Please cite this article as:

Akram Nakhaei, Mohammad Mehdi Sepehri. A Promising Method for Correcting Class Noise in the Presence of Attribute Noise . Int J Hosp Res. 2023; 12(1).