

Ensemble Based Filtering Approaches to Class Noise Detection

Borut Sluban¹ and Dragan Gamberger² and Nada Lavrač³

Abstract. In real-world datasets noisy instances and outliers require the special attention of domain experts. While noise filtering algorithms are usually used to improve the accuracy of induced classification models, our aim is to detect noisy instances to be inspected by human experts in the phase of data understanding, data cleaning and outlier detection. As a result, new algorithms for explicit noise detection have been developed aiming at highest possible precision of noise detection within a reasonable recall threshold. The best performing noise detection algorithms are therefore selected based on a novel quality measure which models the desired precision-recall trade-off, maximizing the recall in the ε -neighborhood of the best precision-achieving noise detection algorithm. New variants of ensemble noise filtering approaches to noise detection have been developed, including the winning random forest based high agreement filter, and implemented in a tool for noise detection which enables the domain expert to identify potential outliers and/or erroneous data instances.

1 INTRODUCTION

The quality of real-world data is inevitably subject to errors and other irregularities that are usually referred to as noise. While data measurement errors represent noise which should best be repaired or discarded from the data, other irregularities such as outliers⁴ may actually lead to the discovery of intriguing new information. For instance, outlier mining has already proved to have important applications in fraud detection and network intrusion detection.

Both types of irregularities, errors and outliers, appear in the data as class noise (errors in class labels) and/or attribute noise (errors in one or more attributes). Regardless of the type of noise, noise may have an adverse impact on the quality of information retrieved from the data, models created from the data, and decisions made based on the data. A qualitative study of the impacts of both types of noise on reduced classification accuracy, increased time complexity and size of the constructed classification or representation model was performed by Zhu & Wu [13].

The main concern of standard noise handling approaches used in machine learning is improved classification accuracy of models induced from the data. In contrast, the focus of this work is on detecting noisy instances to be inspected by human experts in the phase

of data understanding and data cleaning. Therefore, our approach should result in highest possible precision of noise detection. Moreover, it should detect a relatively small set of data instances for which one can claim—with high certainty—that they are indeed noisy and worth the expert’s time and effort.

Stated more formally, our objective is to design and implement noise filtering approaches⁵ with high *precision* Pr of detected noise in the data. However, we do not insist on maximal precision of noise filtering at the cost of low *recall* Re . Instead, we trade-off high precision by maximizing the recall of noise detection in the ε -neighborhood of the most precise noise detection algorithms, where ε is a (small) tolerated proximity of the maximal possible precision that can be achieved by different filtering algorithms $F_j \in \mathcal{F}$.

$$\begin{aligned} BestFilter(\mathcal{F}) &= \arg \max_{F_i \in \mathcal{F} ; Cond} Re(F_i) \\ Cond &: Pr(F_i) > \max_{F_j \in \mathcal{F}} Pr(F_j) - \varepsilon . \end{aligned} \quad (1)$$

This novel quality criterion for noise filtering algorithms is clearly different from classification accuracy optimization criteria standardly used in predictive machine learning.

This paper presents several variations of elementary algorithms for noise detection as well as ensembles of elementary noise detection algorithms. The main novelty of the paper is the redesign and reimplementations of these algorithms, using new algorithm combinations and settings. Several new instantiations of consensus filtering approaches to noise detection, as well as novel random forest based high agreement filters RF180 and RF580, prove to be the best approaches to noise detection, as shown in the experimental evaluation on two noiseless and two real-world datasets with randomly injected artificial noise. The performance of noise detection algorithms, evaluated using the new quality criterion (Eq. 1), is presented in the precision-recall space which has—to the best of our knowledge—not been proposed before in the noise filtering literature.

The paper is structured as follows. Section 2 presents the related work. Section 3 presents three motivating cases showing the practical utility of expert evaluation of noisy examples. Section 4 describes the experimental setting used in this work. Our experiments started by evaluating several elementary noise filtering approaches. For this purpose, we have intentionally inserted different levels of noise into different datasets and measured the precision and recall values for these known sets of noisy examples. The elementary noise filtering algorithms and their results are presented in Section 5. Results presentation in the precision-recall space turned out to be very useful for the identification of properties of various filtering approaches. After

¹ Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, email: borut.sluban@ijs.si - PhD student

² Rudjer Bošković Institute, Zagreb, Croatia, email: dragan.gamberger@irb.hr

³ Jožef Stefan Institute, Ljubljana, Slovenia, and University of Nova Gorica, Nova Gorica, Slovenia, email: nada.lavrac@ijs.si

⁴ In statistics, an outlier is an observation that is numerically distant from the rest of the data, or more formally, it is an observation that lies outside the overall pattern of a distribution.

⁵ Note that the terms noise detection and noise filtering will be used interchangeably in this paper, however the paper actually focuses on noise detection.

that we tested several ensemble filtering approaches using the majority and consensus voting schemes; the results presented in Section 6 are very good, both in the sense of achieved precision and due to the fact that very similar relations among approaches for noise detection repeat for various datasets when presented in the precision-recall space. Finally, Section 7 presents the winning approach: *high agreement random forest filter* with 80% agreement level of decision tree classifiers in the random forest. It outperforms ensemble filters in terms of their effectiveness evaluated in the precision-recall trade-off. A summary and ideas for further work are presented in Section 8.

2 RELATED WORK

A lot of attention in data mining and machine learning has been devoted to the topic of noise handling. One of the first approaches was to make inductive learning algorithms resistant to noise in the data. In order to avoid overfitting noisy training data, constructed models (decision trees and rule sets) were pruned so they can get a more general form, thereby reducing the chance of their performance being influenced by noise [11, 9, 8, 3]. However, pruning alone can not avoid all the damaging effects of noise on the structure of the constructed model.

Another common approach to dealing with noise is to eliminate it by filtering of noisy instances before model construction; this has the advantage that noisy instances will not adversely affect the structure of the induced model. In the filtering approach by Brodley & Friedl [2] multiple learning algorithms are applied to induce classifiers which are then used for noise identification. A data instance is denoted as noise if it is incorrectly classified by one or more classifiers. A cross-validation approach is used: a training set is partitioned into n subsets and repeatedly $n - 1$ subsets are used for classifier induction and the complementary subset for evaluating the classifiers where misclassified instances are identified as noise. This approach is referred to as the *Classification Filter* approach. In the case of multiple classifiers used, a majority or consensus scheme can be used for noise elimination, meaning that an instance is identified as noise only if it is incorrectly classified by the majority or by all of the classifiers, respectively.

A different filtering approach, the so-called *Saturation Filter* by Gamberger & Lavrač [4], is based on the observation that the elimination of noisy examples reduces the Complexity of the Least Complex correct Hypothesis (*CLCH*) value of the training set. The *CLCH* measure is used to find a saturated training set enabling the induction of a hypothesis that correctly captures the generally valid concept of the domain presented by the available data.

An approach to classification filtering using different levels of agreement among multiple classifiers was proposed by Khoshgoftaar et. al [7] and is referred to as the *Ensemble Filter*. For large and distributed datasets a *Partitioning Filter* was presented by Zhu et. al [14] which initially splits the dataset into n partitions and a learning algorithm is induced on each of them. The n predictions are evaluated on the whole dataset. In the end, voting is used to identify noisy examples. Two modifications of the partitioning scheme were introduced by Khoshgoftaar & Rebours [6]. First, the *Multiple-Partitioning Filter* combines the predictions of multiple classifiers learned on each partition. And second, the *Iterative-Partitioning Filter* builds only one model on each subset, but iterates the filtering process until a given stopping criterion is reached.

All of the mentioned filtering algorithms are mainly used for detecting class noise which should be removed from the data to increase its quality. Teng [12] describes a different approach called *Polishing*.

When the noisy instances are identified, instead of removing them, they are repaired by replacing the corrupted values with more appropriate ones. The corrected instances are then reintroduced into the data set.

3 MOTIVATION

Expert evaluation of detected noise can be very useful for domain understanding as well as for the identification of data collection problems. The following real-world cases have been encountered during expert evaluation of patient records, detected as potentially noisy examples during the process of modeling coronary heart disease diagnosis by supervised machine learning methods [5].

- Data measured in a laboratory for two female patients with the same name were accidentally swapped, leading to wrong diagnoses assigned.
- For patients unable to follow a complete testing procedure, test values have been recorded as actual values at the time of test interruption instead of as unknown values.
- There existed a very small set of patients especially strongly reacting to the prescribed drug therapy.

The cases have been recognized due to the fact that available patients' anamnestic, laboratory, and measurement data have suggested a diagnosis different from the diagnosis given by medical practitioners and saved in patient records. This indicates that explicit detection and expert evaluation of noisy examples may help in detecting serious mistakes in data collection, in detection and correction of inappropriate data collection procedures, and distinguishing of a very small number of important exceptions (outliers) from a general concept. The real medical domain in which the described cases have been recognized is included in the experimental part of this work under the name Medical-CHD (where CHD stands for Coronary Heart Disease diagnosis problem, explored in collaboration with medical experts). Obviously, the same methodology may be relevant also for other non-medical domains.

Motivated by the above described cases, we have developed a tool which can support the expert in inspecting the data by finding and explicitly presenting noisy instances. The tool offers the expert a choice of noise filtering algorithms, and returns a set of noisy data instances detected. The scheme of the system is shown in Figure 1.

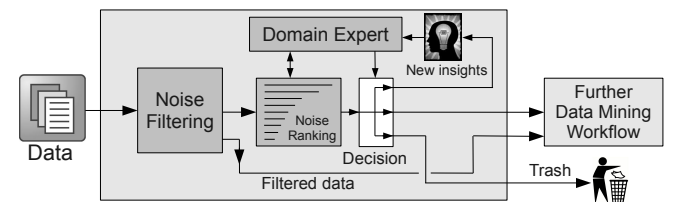


Figure 1. An expert guided filtering tool used in a data mining workflow.

As consensus filters proved to be among the best noise filtering approaches, it was decided that in order to better support the user, visual output should be provided, by ranking instances according to the agreement of being noisy achieved by the majority/consensus of filtering approaches. A visual representation of the list of detected noisy examples, ranked according to the agreement of filtering approaches, is shown in Figure 2.

Filtering was done on dataset: medical-CHD.tab
Noisy examples (class, id, detected by):

```

non-CHD 38  _Bayes_ RF100_ RF500_ SVM_ SVMEasy_ SatFilt_ PruneSF_
----- "Consensus Filter" END -----
non-CHD 14  _RF100_ RF500_ SVMEasy_ SatFilt_ PruneSF_
CHD 29  _RF500_ SVM_ SVMEasy_ PruneSF_
non-CHD 207 _Bayes_ SVM_ SatFilt_ PruneSF_
CHD 210  _RF100_ RF500_ SVM_ SVMEasy_
----- "Majority Filter" END -----
non-CHD 26  _Bayes_ SVM_ SVMEasy_
non-CHD 37  _SVM_ SVMEasy_ PruneSF_
non-CHD 45  _Bayes_ SatFilt_ PruneSF_
CHD 82  _RF100_ RF500_ SVMEasy_
non-CHD 148 _Bayes_ RF100_ SVMEasy_
CHD 157 _Bayes_ SVM_ SVMEasy_
CHD 177 _Bayes_ SVM_ SVMEasy_
non-CHD 24  _SVM_ SVMEasy_
non-CHD 48  _Bayes_ SVM_
non-CHD 104 _Bayes_ SVM_
non-CHD 119 _SVM_ SVMEasy_
non-CHD 154 _SatFilt_ PruneSF_
CHD 162 _Bayes_ PruneSF_
CHD 194  _RF500_ SVM_
CHD 212  _Bayes_ RF100_
non-CHD 3  _Bayes_
non-CHD 34  _SatFilt_
CHD 74  _SVMEasy_
CHD 93  _PruneSF_
non-CHD 165 _SVMEasy_
CHD 173 _SVMEasy_
CHD 175 _SVMEasy_
CHD 186  _SVM_

Noise detected by filtering algorithm:
_Bayes_ detected 12 noisy examples.
_RF100_ detected 6 noisy examples.
_RF500_ detected 6 noisy examples.
_SVM_ detected 14 noisy examples.
_SVMEasy_ detected 15 noisy examples.
_SatFilt_ detected 6 noisy examples.
_PruneSF_ detected 10 noisy examples.
Consensus detected 1 noisy examples.
Majority detected 5 noisy examples.

```

Figure 2. Visual representation of noise filtering output. Instances are ranked by the number of filters identifying them as noise.

4 EXPERIMENTAL SETTING

In our work we examined different filtering approaches, starting with few basic settings of *Classification Filters* and *Saturation Filters*. For the experiments we have chosen two noiseless and two real-world datasets that are specified in Table 1. All the datasets have been injected with three different levels of class noise: in 2%, 5% and 10% of the dataset size, class values of instances were replaced.⁶ We have intentionally chosen two artificial noiseless UCI datasets containing instances of all feasible attribute-value configurations of these domains. This seemed as a very good starting point to evaluate the filtering algorithms’ ability to detect injected noise. The other two datasets are real-world datasets, containing their own noisy instances. All the datasets have two different class values.

For each of the four datasets and for each noise level we made ten datasets with randomly injected noise. This was done to get more reliable noise filtering results, meaning each result presented in this work for a specific domain and a specific noise level is the average value of the results obtained from ten separate experiments each on a dataset with different randomly injected noise. In this way each filtering algorithm was tested not only on four different datasets but actually on 120 datasets, allowing for massive testing of noise detection algorithms despite the relatively small number of selected domains.

5 ELEMENTARY NOISE FILTERING ALGORITHMS

This section presents our settings of elementary filtering approaches which have already been previously mentioned in the literature [2, 4].

⁶ Alternatively, attribute noise could have been inserted, but we were motivated by medical domains where false diagnostics is the main concern.

Table 1. Datasets used for experimental evaluation of noise filtering algorithms. Datasets TTT and KRKP are taken from the UCI machine learning repository [1], MCHD is described in [5], and NA is described in [10].

Dataset		Instances	Attributes	Type
TTT:	Tic Tac Toe	956	9	noiseless
KRKP:	King-Rook vs. King-Pawn	3196	36	noiseless
MCHD:	Medical-CHD	219	37	real-world
NA:	News Articles	464	500	real-world

5.1 Classification Filters

The idea behind this filtering approach presented by Brodley & Friedl [2] is to use a classifier as a tool for detecting noisy instances in data. In this work the simple classifiers used in [2] were replaced by contemporary classifiers, because the assumption is that we should trust the classifiers that they are able to correctly predict the class of a data instance. In this way the incorrectly classified instances are considered to be noise. We performed classification filtering in a ten-fold cross-validation manner with five different classifiers:

- Bayes:** Naïve Bayes classifier (as the baseline filtering algorithm)
- RF100:** Random forest classifier with 100 decision trees
- RF500:** Random forest classifier with 500 decision trees
- SVM:** Support vector machine classifier (with radial basis kernel function)
- SVMEasy:** Support vector machine classifier which performs data scaling and parameter optimization

Recall (Re) and Precision (Pr) values of noise detection by different classification filters on the datasets with injected noise are presented in Table 2. All the values in the table are expressed in percentages (%) of noise level, recall and precision.

The results show that the classification filters are fairly good at finding all the injected noise, since most of the recall values in Table 2 are higher than 90%. On the other hand, in terms of precision, the filters perform rather poorly, as the results only rarely exceed 80%, meaning that these noise filtering algorithms identified also a lot of non-noisy/regular instances as noisy.

5.2 Saturation Filters

Gamberger & Lavrač [4] introduced a concept of a *saturated* training set enabling the induction of a simple hypothesis correctly capturing the concept of the domain presented by the data. By elimination of noisy examples a non-saturated training set may be transformed into a saturated one. In a saturation filtering algorithm noisy examples are recognized as those enabling the reduction of the complexity of the resulting hypothesis measured by the Complexity of the Least Complex correct Hypothesis (*CLCH* value) of the training set. A practical problem of the saturation filter implementation is to find a complexity measure for a classification model constructed from a training set, which corresponds to the *CLCH* value and is sensitive enough to enable the filtering algorithm to distinguish between noisy and non-noisy instances.

The filter is constructed from two basic methods. The first one is a *saturation test*. At first it computes the complexity of the classification model for the given training set, then it iteratively excludes only one training example and computes the complexity of a classification model induced from the rest of the training examples. The examples which had the most effect in reducing the complexity of the classification model with their exclusion are labeled as the *most*

Table 2. Results of different elementary noise filtering algorithms on four datasets with various levels of injected class noise, where *Re* stand for recall and *Pr* for precision of detecting artificially injected class noise.

Dataset	Noise level (%)	Classification filters										Saturation filters			
		Bayes		RF100		RF500		SVM		SVMEasy		SatFilt		PruneSF	
		Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr	Re	Pr
TTT	2	65.79	4.23	89.47	14.84	88.42	15.65	98.42	54.36	98.95	54.91	70.00	22.58	83.16	13.52
	5	69.79	10.96	91.25	30.98	91.46	32.19	98.54	75.58	98.54	75.36	63.13	38.98	78.96	25.93
	10	69.16	20.61	89.69	44.17	89.27	45.89	98.44	86.71	98.44	86.71	51.77	46.63	67.71	34.18
KRP	2	87.81	12.44	98.59	60.10	99.06	60.20	91.72	18.92	99.69	79.25	84.69	81.06	94.68	45.99
	5	87.50	26.02	98.56	75.98	98.44	77.80	91.88	37.80	98.69	83.41	62.29	73.74	91.67	48.04
	10	85.78	40.93	97.47	84.20	97.56	84.92	91.62	56.97	97.78	85.00	41.88	60.74	78.88	49.11
MCHD	2	92.50	19.73	100.00	36.32	100.00	41.61	95.00	17.36	100.00	19.64	75.00	35.16	82.50	25.90
	5	94.55	40.16	96.36	60.15	96.36	61.05	92.73	36.51	93.64	37.56	54.55	51.73	78.18	48.53
	10	92.27	54.66	95.91	76.24	97.73	79.10	91.82	52.29	90.91	50.73	40.00	61.78	66.36	61.17
NA	2	93.33	18.78	93.33	24.90	95.56	24.65	96.67	27.25	97.78	33.40	46.67	23.03	55.56	14.89
	5	91.30	36.35	94.35	44.91	94.35	46.21	96.09	49.64	97.35	55.93	30.43	33.05	52.61	29.05
	10	91.30	53.13	92.17	64.37	92.83	64.37	93.91	67.42	93.91	70.33	22.39	40.89	43.70	38.87

noisy and are passed on to the second method. The second method, the *filter*, randomly chooses one among the most noisy examples and finally excludes it from the training dataset, while the others are returned back. This is repeated as long as the saturation test finds noisy examples, meaning that a saturated subset has not yet been obtained.

In our implementation a decision tree learner is used to construct models and the number of all nodes in the decision tree is used as the *CLCH* measure of the complexity of the model. The saturation filtering algorithm has to construct as many decision trees as there are instances in the dataset and this is repeated as many times as there are potentially noisy instances. Consequently, this implementation is very time consuming. Therefore we also tested an implementation, named *PruneSF*, that prunes all the leaves of the decision tree supporting only one instance of the first constructed model, removes those instances, and only then starts the filtering phase. This works as a speed up to the existing saturation filter implementation. Performance results of noise detection of both methods can be found in Table 2 as well.

These two filters have much lower recall of injected noise detection than the classification filters (CFs) and the precision of injected noise detection is usually between the best and the worst CFs, except for the News Articles domain where the saturation based filters perform worst in every aspect. However, among the saturation filters a permanent pattern can be observed, the *PruneSF* has always a significantly better recall than the *SatFilt*, but of course at the expense of lower precision of injected noise detection.

5.3 Evaluation in the Precision-Recall Space

Previous research in noise filtering [2, 4, 7, 13, 14] has mainly concentrated on noise filtering with the aim of cleaning the dataset and thereby improving the classification accuracy of models induced from data with eliminated noisy instances. Note that model accuracy maximization is not our goal. Instead, we aim at precise detection of a small set of noisy instances to be shown to the domain expert for inspection. Hence, in the setting of artificially injected noise our goal is to maximize the precision of detected noise. More precisely, according to Equation 1, our goal is to detect noisy instances with the maximal recall that can be achieved within an ε proximity of the maximal precision of noise detected by the filters. When deciding on the best filtering algorithm, the ε value is defined based on the maximal precision achieved by one of the filtering algorithms, meaning how much decrease/drop in average precision we still allow in terms

of gaining higher recall.

In order to simplify the interpretation of results presented in Table 2, take the exact (noiseless) domain King-Rook vs. King-Pawn (KRP) to present the results in the Precision-Recall space. The Precision-Recall space is a two dimensional space where recall is plotted on the X-axis and precision on the Y-axis. This space allows us to visually inspect the quality of the so far presented filters on the selected domain, which is shown in Figure 3.

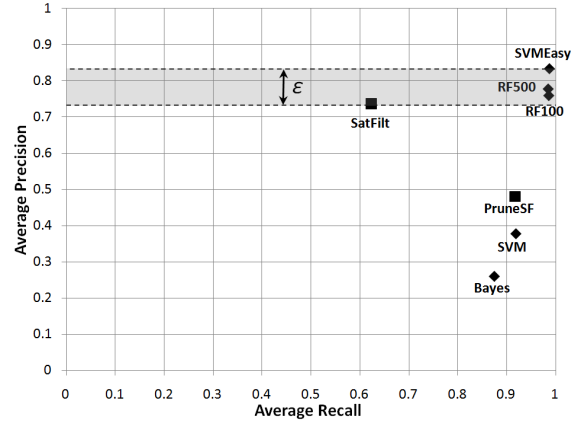


Figure 3. Results in the Precision-Recall space of the performance of different filtering algorithms on the KRP dataset with 5% injected noise.

By inspecting Figure 3 we can see that within the ε proximity of the maximal precision achieving filtering algorithm (SVMEasy) there are three other filtering algorithms (RF500, RF100 and SatFilt). But since no one of them is achieving a higher recall, the classification filter with SVMEasy is selected as the best filtering algorithm according to Equation 1 with $\varepsilon = 0.1$. This ε value was at this point only chosen to depict the principle. A more reasonable ε value selection in this case would be up to 0.03. Moreover, we see that the elementary filters have a relatively high recall (especially SVMEasy, RF500, RF100, PruneSF and SVM), whereas their precision is not satisfactory. This observation has motivated us to inspect also ensemble filtering algorithms in the Precision-Recall space. While ensemble filtering has been previously proposed in the noise filtering literature, its inspection and evaluation in the Precision-Recall space proposed in this paper sheds a new light on ensemble filtering, as shown in the next section.

6 ENSEMBLE FILTERS

The filtering algorithms presented so far were quite good at detecting the injected noise, but they identified as noisy also a lot of regular (non-noisy) instances and therefore having a low precision of injected noise detection. To be more convinced that an instance is noisy, predictions of several different filtering algorithms should be taken into account when deciding if an instance should be identified as noise.

We experimented with ensemble filtering in a majority and in a consensus voting scheme. The novelty of this approach is that ensemble filters were constructed from classification and saturation filters, forming different groups of filtering algorithms:

Ens1: Bayes, RF100, RF500, SVM, SVMEasy

Ens2: Bayes, RF100, RF500, SVM, SVMEasy, SatFilt, PruneSF

Ens3: Bayes, RF500, SVMEasy, SatFilt

Ens4: Bayes, RF500, SVMEasy, PruneSF

Ensemble filtering using a majority voting scheme (a simple majority of votes of filters in the given ensemble) did not outperform the precision of injected noise detection achieved by the best elementary filtering algorithms alone.⁷ On the other hand, with consensus filtering we significantly improved the precision of noise detection, however, at the expense of a worse recall.

An overview of the results is given in Table 3. The results show that the consensus filter made of classification filters (Ens1) has the best recall, which was already a notable feature of the classification filters examined in Section 5.1. The other three consensus filters (Ens2-4) used also one or two saturation-based filters, which use a different approach to noise filtering and thereby contribute to the diversity among filters that get to vote on noise identification. If more filters from different filtering approaches agree on an instance being noisy, the precision of noise detection is increased (see the precision results of filters Ens2 to Ens4). On the other hand a lower rate of recall is hence also an obvious consequence.

Rather than concluding all this by inspecting a table of numbers, a much easier and more intuitive way is to interpret the results in the Precision-Recall space. See Figures 4, 5, 6 and 7 for all four experimental domains, illustrated in the case of 5% artificially injected noise.

Table 3. Results of four consensus filters (made from four different groups of elementary filters) on four datasets with various levels of injected class noise.

Data	NL	Group of filters							
		Ens1		Ens2		Ens3		Ens4	
		Re	Pr	Re	Pr	Re	Pr	Re	Pr
TTT	2	65.26	60.30	50.53	96.89	51.58	91.72	58.42	95.71
	5	68.96	79.36	42.92	98.70	47.92	96.15	57.92	95.32
	10	66.98	89.09	34.27	100.0	38.75	98.63	47.71	97.87
KRKP	2	82.19	90.08	67.97	93.59	74.38	93.88	82.66	92.50
	5	81.94	91.88	49.38	96.82	52.92	97.09	80.63	96.70
	10	80.16	94.34	32.25	97.87	35.94	97.76	66.94	97.27
MCHD	2	90.00	64.38	65.00	89.50	70.00	84.50	77.50	90.00
	5	87.27	84.85	47.27	90.08	50.00	85.08	72.73	88.90
	10	84.09	92.12	31.82	96.32	33.64	93.07	58.64	95.89
NA	2	90.00	48.39	42.22	79.71	45.56	76.68	55.56	60.77
	5	89.13	69.83	25.65	77.15	28.70	73.19	48.70	76.03
	10	86.30	83.02	18.70	89.75	21.30	89.10	42.61	85.15

⁷ Tabular results are available by request from the authors. Due to space limitations they are not presented in this paper.

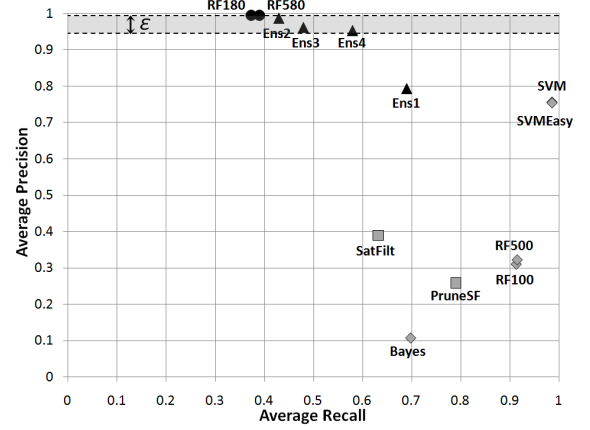


Figure 4. Performance results in the Precision-Recall space of different filtering algorithms on the Tic Tac Toe dataset with 5% injected noise.

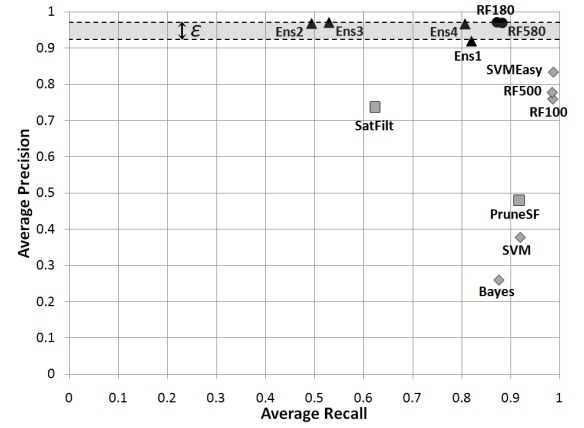


Figure 5. Performance results in the Precision-Recall space of different filtering algorithms on the King-Rook vs. King-Pawn dataset with 5% injected noise.

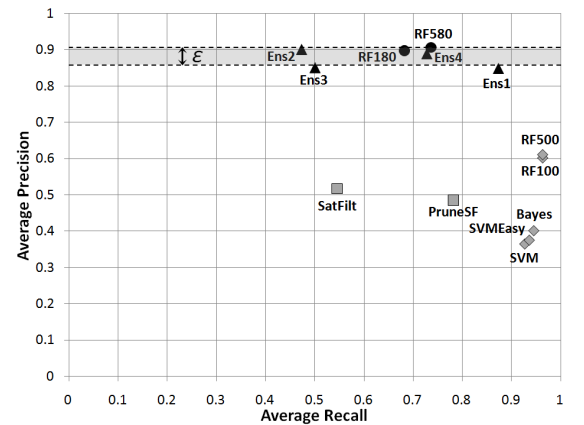


Figure 6. Performance results in the Precision-Recall space of different filtering algorithms on the Medical-CHD dataset with 5% injected noise.

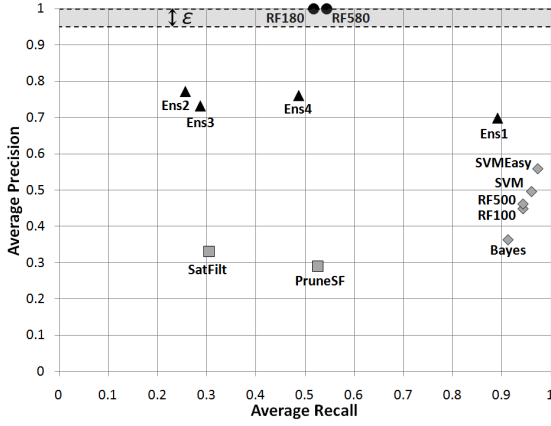


Figure 7. Performance results in the Precision-Recall space of different filtering algorithms on the News Articles dataset with 5% injected noise.

7 THE WINNING APPROACH: HIGH AGREEMENT RANDOM FOREST FILTER

Motivated by very good precision results of ensemble (consensus) filtering in Section 6 and very good recall results of classification filtering with random forest (RF) classifiers in Section 5.1, we further focused on the RF classifier. When used for filtering, RF classification filters act like an ensemble filter, since randomly generated decision trees vote for the class to be assigned to an instance. However, if the class that got the maximal number of votes has won only by a single vote, this result is not really convincing, since the decision trees are generated randomly. Therefore, we decided to demand high agreement of classifiers, i.e., a significantly increased majority of votes, namely, for two class domains used in our experiments this means that more than 60, 70, 80 or 90% of the randomly generated decision trees should classify an instance into the opposite class in order to accept it as noisy.

We have tested this agreement based approach with RF classifiers with 100 and 500 decision trees, each with four different agreement levels: 60, 70, 80 and 90%. Since the results obtained did not vary a lot depending on the number of trees, we present only the results obtained with RF500 in Table 4. Figure 8 clearly indicates the increase in precision based on the increased agreement level. However, in the majority of the experimental settings the best results in terms of precision-recall trade-off are achieved at the 80% agreement level. Therefore, Figures 4, 5, 6 and 7 depict just these results: in the case of 100 decision trees in the RF used (RF100) the result at 80% agreement level is referred to as RF180, and in the case of RF500 it is referred to as RF580.

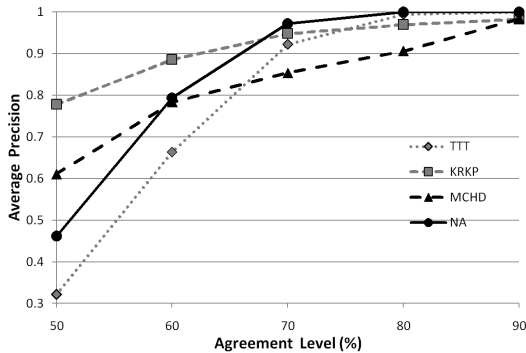


Figure 8. Precision of noise detection based on the agreement level for RF500 on all domains with 5% injected noise.

Table 4. Recall (Re) and Precision (Pr) of RF500 with different agreement levels on four datasets with various levels of injected class noise.

Data	NL	Agreement level							
		60%		70%		80%		90%	
		Re	Pr	Re	Pr	Re	Pr	Re	Pr
TTT	2	83.16	46.89	63.68	87.31	40.53	100.00	15.26	100.00
	5	80.00	66.35	63.96	92.24	38.96	99.44	13.33	100.00
	10	76.46	73.00	56.98	94.01	31.04	99.00	9.48	100.00
KRKP	2	98.13	78.74	95.31	87.79	90.16	92.62	73.28	95.33
	5	96.94	88.56	94.38	94.81	88.25	96.98	64.50	98.28
	10	95.38	92.62	89.94	96.38	78.28	98.39	48.13	99.04
MCHD	2	97.50	54.44	97.50	64.10	87.50	78.17	45.00	90.00
	5	94.55	78.37	85.45	85.40	73.64	90.62	35.45	98.33
	10	92.73	89.75	83.18	94.42	55.45	95.54	15.45	100.00
NA	2	87.78	55.22	83.33	92.14	57.78	100.00	23.33	100.00
	5	88.70	79.40	78.26	97.22	54.35	100.00	12.17	100.00
	10	83.91	86.94	72.39	97.25	40.43	100.00	2.83	70.00

8 SUMMARY AND FURTHER WORK

We proposed several new elementary and ensemble noise filtering algorithms and evaluated their performance using a new performance criterion which trades off precision and recall of noise detection. The evaluation in the Precision-Recall space shows that ensemble filters composed of a mixture of classification and saturation filters outperform the elementary filters. Moreover, it was shown that consensus filtering outperforms filtering by simple majority voting of filters in the given ensemble. The winning approach is however the new high agreement based random forest filter (with 80% agreement level of decision tree classifiers in the random forest): it outperforms ensemble filters in terms of the achieved precision-recall trade-off. In further work we will focus on this filtering approach and evaluate its capacity for filtering massive amounts of data.

References

- [1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
- [2] C. E. Brodley and M. A. Friedl, ‘Identifying mislabeled training data’, *Journal of Artificial Intelligence Research*, **11**, 131–167, (1999).
- [3] J. Fürnkranz, ‘Pruning algorithms for rule learning’, *Machine Learning*, **27**, 139–171, (May 1997).
- [4] D. Gamberger and N. Lavrač, ‘Conditions for Occam’s razor applicability and noise elimination’, in *Lecture Notes in Artificial Intelligence: Machine Learning: ECML-97*, volume 1224, pp. 108–123, (1997).
- [5] D. Gamberger, N. Lavrač, and G. Krstačić, ‘Active subgroup mining: A case study in a coronary heart disease risk group detection’, *Artificial Intelligence in Medicine*, **28**, 27–57, (2003).
- [6] T. M. Khoshgoftaar and P. Rebour, ‘Generating multiple noise elimination filters with the ensemble-partitioning filter’, in *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, pp. 369–375, (2004).
- [7] T. M. Khoshgoftaar, S. Zhong, and V. Joshi, ‘Noise elimination with ensemble-classification filtering for software quality estimation’, *Intelligent Data Analysis*, **9**(1), 3–27, (2005).
- [8] J. Mingers, ‘An empirical comparison of pruning methods for decision tree induction’, *Machine Learning*, **4**, 227–243, (1989).
- [9] T. Niblett and I. Bratko, ‘Learning decision rules in noisy domains’, in *Research and Development in Expert Systems*, ed., M. Bramer. Cambridge University Press, (1987).
- [10] S. Pollak, ‘Text classification of articles on kenyan elections’, in *Proceedings of the 4th Language & Technology Conference: Human language technologies as a challenge for computer science and linguistics*, pp. 229–233, (2009).
- [11] J. R. Quinlan, ‘Simplifying decision trees’, *International Journal of Man-Machine Studies*, **27**, 221–234, (1987).
- [12] C. M. Teng, ‘Correcting noisy data’, in *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 239–248, (1999).
- [13] X. Zhu and X. Wu, ‘Class noise vs. attribute noise: A quantitative study of their impacts’, *Artificial Intelligence Review*, **22**, 177–210, (2004).
- [14] X. Zhu, X. Wu, and Q. Chen, ‘Eliminating class noise in large datasets’, in *Proc. of the Int. Conf. on Machine Learning*, pp. 920–927, (2003).