12-1-2019

Enhanced label noise filtering with multiple voting

Donghai Guan
Nanjing University of Aeronautics and Astronautics

Maqbool Hussain
Sejong University

Weiwei Yuan
Nanjing University of Aeronautics and Astronautics

Asad Masood Khattak
Zayed University

Muhammad Fahim
Innopolis University

See next page for additional authors

Recommended Citation

Author First name, Last name, Institution

Donghai Guan, Maqbool Hussain, Weiwei Yuan, Asad Masood Khattak, Muhammad Fahim, and Wajahat
Ali Khan

Enhanced Label Noise Filtering with Multiple Voting

Donghai Guan [1],[†], Maqbool Hussain [2,3],[†] (ID), Weiwei Yuan [1], Asad Masood Khattak [4], Muhammad Fahim [5] and Wajahat Ali Khan [6],* (ID)

[1] College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; dhguan@nuaa.edu.cn (D.G.); yuanweiwei@nuaa.edu.cn (W.Y.)

[2] Department of Software, Sejong University, 209, Neungdong-ro, Gwangjin-gu, Seoul 05006, Korea; maqbool.hussain@sejong.ac.kr or maqboolhussain@oakland.edu

[3] Department of Computer Science and Engineering, Oakland University, Rochester, MI 48309, USA

[4] College of Technological Innovation, Zayed University, Dubai 144534, UAE; asad.khattak@zu.ac.ae

[5] Institute of Information Systems, Innopolis University, Tatarstan 420500, Russia; m.fahim@innopolis.ru

[6] Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Korea

* Correspondence: wajahat.alikhan@oslab.khu.ac.kr

† Joint first authors.

check for updates

Abstract: Label noises exist in many applications, and their presence can degrade learning performance. Researchers usually use filters to identify and eliminate them prior to training. The ensemble learning based filter (EnFilter) is the most widely used filter. According to the voting mechanism, EnFilter is mainly divided into two types: single-voting based (SVFilter) and multiple-voting based (MVFilter). In general, MVFilter is more often preferred because multiple-voting could address the intrinsic limitations of single-voting. However, the most important unsolved issue in MVFilter is how to determine the optimal decision point (ODP). Conceptually, the decision point is a threshold value, which determines the noise detection performance. To maximize the performance of MVFilter, we propose a novel approach to compute the optimal decision point. Our approach is data driven and cost sensitive, which determines the ODP based on the given noisy training dataset and noise misrecognition cost matrix. The core idea of our approach is to estimate the mislabeled data probability distributions, based on which the expected cost of each possible decision point could be inferred. Experimental results on a set of benchmark datasets illustrate the utility of our proposed approach.

Keywords: mislabeled data filter; single-voting; multiple-voting; optimal decision point; cost minimization

1. Introduction

Real-world training data often include noises (or errors), which can be mainly categorized into two types: label error and feature error [1–5]. Label error arises when the class label of data is incorrect, while the feature error arises when the features of data are corrupted. These noises are made for various reasons. For example, sensor involved applications (such as WSN and IoT) may make noises due to the intrinsic instability of sensors [6,7]. In addition, big data further contribute to the emergence of noise [8]. When training data are noisy, the performance of learning based on it will be degraded. These two types of error have been individually studied by many works. We focus on the label error to study in this work.

The label errors are mainly caused by the subjective nature of the labeling task and lack of information for making the true label. Domain experts usually provide labeling that mainly depends on their heuristics and domain knowledge. It is a crucial fact that mislabeling cannot be even avoided

with a thorough inspection of domain experts. It happens commonly when a consensus is not made during the annotation process by multiple domain experts. Mislabeling is very common in domains requiring rapid development, such as bioinformatics. For example, in a study on breast tumor [9], there existed nine subjective mislabelings among forty-nine features in the training data. Furthermore, mislabeling is also caused by insufficient information available to the expert [10,11]. An example of such information includes the unavailability of data of certain observation results of tests. Physicians are not confident to conclude the crisp diagnosis decision in the presence of partial information.

The existence of mislabeled data usually degrades the performance of learning [12–17]. In general, the goal of a learning algorithm is to search for the best hypothesis from its hypothesis space. In supervised learning, the best hypothesis is usually decided by the correlations between the features and the labels of training data. Therefore, searching for the best hypothesis will be influenced by the mislabeled data, which results in selecting a non-optimal hypothesis. The non-optimal hypothesis can bring a set of negative effects, including classification accuracy reduction, classifier construction time and complexity increase, and others.

The approaches dealing with mislabeled data are categorized into two main groups: robust algorithm design [18–22] and noise filter [23–41]. The first one is mainly good at developing a novel algorithm that deals with noisy data during model training. The second approach is good at the identification and filtering of mislabeled data before the training. Evidence exists that it is usually difficult to develop a robust algorithm that is insensitive to noisy data. Furthermore, it is revealed that mislabeled noisy data have a severe impact on the approach; even if the design is claimed to be robust. In comparison, filter based approaches have significant performance leverage over the robust algorithm. The core contribution of this work is in the area of filter based approaches.

Several filter based approaches are used to deal with mislabeled data, where the ensemble learning based filter (EnFilter) is a widely used approach based on its promising performance [23,24,30,33]. EnFilter leverages others with a unique approach by employing multiple classifiers to identify noises based on their voting.

According to the adopted voting mechanism, EnFilter consists of two types: single-voting based (SVFilter) and multiple-voting based (MVFilter).The SVFilter detects noises only based on one-time voting of multiple classifiers, and therefore, it has a potential instability problem.

To solve this instability problem, an MVFilter was proposed in our previous work [40]. In essence, an MVFilter consists of a set of SVFilters (assume this number is t). For the training data, if at least m ($m \leq t$) SVFilters treat it as noisy, then the MVFilter regards it as noisy. The internal mechanism of the SVFilter makes comparisons between each SVFilter, therefore through their fusion, the MVFilter can improve the noise detection stability and accuracy compared to the SVFilter. In the design of the MVFilter, one of the key issues is how to define the value of m (called the decision point), which actually defines the noise detection rule. An optimal decision point (ODP) could maximize the performance of an MVFilter. In [40], the decision points were empirically explored with different representative values. However, a systematic approach to determining the ODP is lacking.

To this end, a novel approach is proposed in this work to compute the ODP for an MVFilter. Instead of only considering the number of errors, our approach takes cost information into account because many applications have unequal costs for various errors. When a cost matrix containing various cost values is given, the ODP selected by our approach is able to identify the noises that minimize the expected cost.

The core idea of our approach is as follows: firstly, estimating the mislabeled data distribution in the noisy training dataset; secondly, estimating the expected costs of each possible decision point; and finally, the optimal decision point determined by minimizing the expected cost.

We tested our approach based on a set of MVFilters. The experimental results show that our approach can significantly improve the performance of existing MVFilters. Our approach consistently works well for different datasets and different cost matrices. In addition, our approach is effective and straightforward. Only a few predefined parameters and prior knowledge are required.

In the next section, we will briefly review ensemble learning based noise filters. Section 3 analyzes the performance of the MVFilter when costs are considered. Our novel approach is presented in Section 4. The experimental evaluations are presented in Section 5. Section 6 concludes this work and presents future work.

2. Related Work

This work presents an approach to improve multiple-voting based ensemble filters for mislabeled data recognition. As necessary background knowledge, conventional ensemble learning based filters (EnFilter) will be introduced firstly. Then, multiple-voting based filter (MVFilter) will be presented.

EnFilter employs an ensemble classifier to detect mislabeled instances by constructing a set of base level classifiers and then using their classifications to identify mislabeled instances. The general approach is to tag an instance as mislabeled if x of the m base level classifiers cannot classify it correctly. The majority filter (MF) and consensus filter (CF) are the representative EnFilter algorithms [27,28]. MF tags an instance as mislabeled if more than half of the m base level classifiers classify it incorrectly. CF requires that all base level classifiers must fail to classify an instance as the class given by its training label for it to be eliminated from the training data.

The reason for employing ensemble classifiers in EnFilter is that the ensemble classifier has better performance than each base level classifier on a dataset if two conditions hold: (1) the probability of a correct classification by each individual classifier is greater than 0.,5 and (2) the errors in predictions of the base level classifiers are independent.

Algorithm 1 enlists the majority filter (MF) algorithm as a representative EnFilter algorithm. In Step 1, it initializes with the n disjoint subset of the training set E. In Step 2; it initializes the empty set A to reflect the noisy examples. The main loop in Steps 3–6 processes each subset E_i, in an iterative manner. Step 4 establishes subset E_t having all examples from E except the one existing in E_i. These examples from E_t are used in an arbitrary inductive learning algorithm in Step 6 to induce a hypothesis (a classifier) H_j. In Step 14, all those examples from E_i are added to A as potential noisy examples for which the majority of the hypothesis does not hold. CF is more conservative than MF because of the severer condition for noise identification, which ultimately results in fewer instances being eliminated from the training set. With a such property, the CF differs from MF in Step 14, and thus, it considers examples in E_i as noisy only if all of them are classified incorrectly by the hypothesis. Furthermore, CF has the risk of retaining bad data.

As Algorithm 1 shows, the core of EnFilter is adopting a voting mechanism to recognize noises. Training data x, in subset E_i after data partitioning on E, will be voted on by the multiple classifiers, which are trained based on the data in $E \setminus E_i$. Suppose y(x) is the function to determine whether x is mislabeled, then y(x) = vote(classifiers($E\setminus E_i$), x). Like MF and CF, the conventional EnFilter decides y(x) only based on one-time voting and therefore, is a single-voting based filter (SVFilter).

As pointed out in [40], the SVFilter suffers from an instability problem. For data x, if the SVFilter runs twice, the first data random partitioning might assign x to subset E_i, while the second time, it assigns x to subset E_k. Therefore, we have y(x) = vote(classifiers($E\setminus E_i$)) and y(x) = vote(classifiers($E\setminus E_k$)). Note that as there is diversity between $E\setminus E_i$ and $E\setminus E_k$, the voting results of two SVFilters might be different. Therefore, instead of one-time voting, multiple-voting based filters (MVFilter) have been proposed to address this instability problem.

MVFilter consists of t SVFilters. Each SVFilter generates its own decision about suspected mislabeled data index A_i. Finally, all the different decisions $A_{i(i=1:t)}$ will be combined by the MVFilter to output the final decision about which data are mislabeled. Therefore, the decision function of MVFilter can be described as y(x) = $\text{vote}_2(\text{vote}_1(E\setminus E^1), \text{vote}_1(E\setminus E^2), \ldots, \text{vote}_1(E\setminus E^t))$. In this function, vote_1 is the voting policy used by each SVFilter; vote_2 is the voting policy used by MVFilter; E^i is the subset containing x obtained from the ith SVFilter. Usually, the vote_1 policy can either be based on majority voting or consensus voting. For the vote_2 policy, we have developed three policies: majority voting, consensus voting, and one-time veto. One-time vote means that if at least one SVFilter tags

data as mislabeled, then the MVFilter will tag these data as mislabeled. In the MVFilter, different $vote_1$ and $vote_2$ policies can be combined to make various algorithms. As the example of MVFilter, the MF_{MF} [40] algorithm is presented in Algorithm 2, which utilizes majority voting for both $vote_1$ and $vote_2$.

Algorithm 1 Majority filtering algorithm.

Algorithm: majority filtering (MF)
Input: E (training set)
Parameter: n (number of subsets), y (number of learning algorithms),
$A_1, A_2, \ldots, A_y$ (y kinds of learning algorithms)
Output: A (detected noisy subset of E)
(1) form n disjoint almost equally sized subset of E_i, where $\bigcup_i E_i = E$
(2) $A \leftarrow \varnothing$
(3) **for** i=1, ..., n **do**
(4)   form $E_t \leftarrow E \backslash E_i$
(5)   **for** j = 1,...y **do**
(6)   induce H_j based on examples 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,...,y **do**
(11)   **if** H_j incorrectly classifies e
(12)   **then** $ErrorCounter \leftarrow ErrorCounter + 1$
(13)   **end for**
(14)   **if** $ErrorCounter > \frac{y}{2}$, **then** $A \leftarrow A \cup \{e\}$
(15)   **end for**
(16) **end for**

Algorithm 2 MF_{MF} algorithm.

MajorityFiltering_MajorityFiltering (MF$_{MF}$)
Input: E (training set)
Parameter: n (number of subsets), y (number of learning algorithms),
t (number of times of subsets partitioning), $A_1, A_2, \ldots, A_y$ (y kinds of learning algorithms)
Output: A (detected noisy subset of E)
(1) **for** p = 1,..., t **do**
(2)   form n disjoint almost equally sized subset of E_{pi}, where $\bigcup_i E_{pi} = E$
(3) $A^p \leftarrow \varnothing$
(4)   **for** i = 1, ..., n **do**
(5)   form $E_t \leftarrow E \backslash E_{pi}$
(6)   **for** j = 1,...y **do**
(7)   induce H_{pj} based on examples in E_t and A_j
(8)   **end for**
(9)   **for** every $e \in E_{pi}$ **do**
(10) $ErrorCounter \leftarrow 0$
(11)   **for** j = 1,...,y **do**
(12)   **if** H_{pj} incorrectly classifies e
(13)   **then** $ErrorCounter \leftarrow ErrorCounter + 1$
(14)   **end for**
(15)   **if** $ErrorCounter > \frac{y}{2}$, **then** $A^p \leftarrow A^p \cup \{e\}$
(16)   **end for**
(17)   **end for**
(18) **end for**
(19) $A \leftarrow \varnothing$
(20) **for** every $e \in E$ **do**
(21) $ErrorCounter \leftarrow 0$
(22)   **for** j = 1,..., p **do**
(23)   **if** $e \in A^p$
(24)   **then** $ErrorCounter \leftarrow ErrorCounter + 1$
(25)   **end for**
(26)   **if** $ErrorCounter > \frac{p}{2}$, **then** $A \leftarrow A \cup \{e\}$
(27) **end for**

---

3. Analysis of Decision Point, Error Probability, and Cost for MVFilter

The multiple-voting based filter (MVFilter) consists of several single-voting based filters (SVFilter). The MVFilter treats data as mislabeled if at least m out of t SVFilters identify these data as mislabeled. Obviously, for different m values, the recognized noises by an MVFilter will be different. The selection of the m value plays an important role in an MVFilter. Because the m value decides the noise identifying

results, it is called the "decision point" in this work. Our goal is to find a way to decide the "optimal decision point" to maximize the performance of MVFilter.

When a filter works on a noisy training dataset, it is usually hard to recognize all the noises perfectly. The errors made by a filter include two types: The first type (E1) occurs when declaring a correctly labeled example as mislabeled and is subsequently discarded. The second type of error (E2) corresponds to declaring a mislabeled example as correctly labeled. For a well designed filter, it is desirable to avoid both E1 and E2 errors. However, conceptually, E1 and E2 are conflicting. To reduce E1 errors, the filter should make a more severe noise detection policy, which tends to increase E2 errors. In MVFilter, the selection of the decision point will influence the probability to make an E1 or E2 error.

3.1. Relationship between the Decision Point and Error Probability in MVFilter

An MVFilter fuses the noise detection results of multiple SVFilters, while an SVFilter fuses the classification results of multiple classifiers. Therefore, the errors made by each classifier are the basis to infer the errors made by an MVFilter.

Let $P(E1_i)$ and $P(E2_i)$ be the probability that classifier i makes an E1 and E2 error, respectively. To clarify the analysis, it is assumed that all the various classifiers in an SVFilter have the same probability of making an error. Therefore, we assume that $P(E1_i) = P(E1)$ and $P(E2_i) = P(E2)$. The most commonly used SVFilters include the majority filter (MF) and consensus filter (CF). The analysis here is based on MF, while a similar analysis can be conducted for CF.

MF makes an E1 (or E2) error when more than half of these classifiers make an E1 (or E2) error. If the number of classifiers in MF is y, then we have:

$$P(E1_{MF}) = \sum_{j>y/2}^{j=y} (P(E1)^j (1 - P(E1))^{y-j}) \begin{pmatrix} y \\ j \end{pmatrix}$$

$$P(E2_{MF}) = \sum_{j>y/2}^{j=y} (P(E2)^j (1 - P(E2))^{y-j}) \begin{pmatrix} y \\ j \end{pmatrix}$$

Suppose an MVFilter consists of t majority filters (MMF). Let $P(E1_{MF_i})$ and $P(E2_{MF_i})$ denote the probability that each MF makes an E1 and E2 error, respectively. To simplify the analysis, it is assumed that $P(E1_{MF_i}) = P(E1_{MF})$ and $P(E2_{MF_i}) = P(E2_{MF})$. The decision rule of an MVFilter is "if at least m of the t SVFilters think data is mislabeled, then these data are identified as mislabeled". This m value, called the decision point, will influence the probabilities of making an error for an MVFilter. Let MMF represent an MVFilter consisting of multiple majority filters, then the following relationships can be found:

$$P(E1_{MMF}) =$$

$$\sum_{j \geq m}^{j=t} (P(E1_{MF})^j (1 - P(E1_{MF}))^{t-j}) \begin{pmatrix} t \\ j \end{pmatrix}$$

$$P(E2_{MMF}) =$$

$$\sum_{j > t-m}^{j=t} (P(E2_{MF})^j (1 - P(E2_{MF}))^{t-j}) \begin{pmatrix} t \\ j \end{pmatrix}$$

The decision point value m can be any number between one and t. Among all possible values, the representative decision points include $m = 1$, $m = t/2$, $m = t$. When m = 1, data will be identified as mislabeled if at least one SVFilter thinks these data are mislabeled. When $m = t$, data will be identified as mislabeled only if all the t SVFilters think these data are mislabeled. Conceptually, the noise detection rule is too loose if the decision point is one, while the rule is too strict if the decision point is

t. In this sense, $m = t/2$ is usually moderate. For these three representative decision points, we have the following relationships:

$$P(E1_{MMF}|m = 1) = 1 - (1 - P(E1_{MF}))^t$$

$$P(E2_{MMF}|m = 1) = P(E2_{MF})^t$$

$$P(E1_{MMF}|m = t/2) =$$

$$\sum_{j \geq t/2}^{j=t} (P(E1_{MF})^j (1 - P(E1_{MF}))^{t-j}) \binom{t}{j}$$

$$P(E2_{MMF}|m = t/2) =$$

$$\sum_{j > t/2}^{j=t} (P(E2_{MF})^j (1 - P(E2_{MF}))^{t-j}) \binom{t}{j}$$

$$P(E1_{MMF}|m = t) = P(E1_{MF})^t$$

$$P(E2_{MMF}|m = t) = 1 - (1 - P(E2_{MF}))^t$$

For the above relationships, normally we have:

$$P(E1_{MMF}|m = t) < P(E1_{MMF}|m = t/2) <$$
$$P(E1_{MMF}|m = 1)$$

$$P(E2_{MMF}|m = 1) < P(E2_{MMF}|m = t/2) <$$
$$P(E2_{MMF}|m = t)$$

As $P(E1_{MMF})$ and $P(E2_{MMF})$ are conflicting, the optimal decision point should make a trade-off between these two probabilities. Therefore, if the probability of making errors is the only concern of MVFilter, the optimal decision point (ODP) is $ODP = \underset{m=1:j}{\arg\min}(P(E1_{MMF}) + P(E2_{MMF}))$.

3.2. Relationship between the Decision Point and Error Cost

In Section 3.1, for an MVFilter, the relationships between the optimal decision point and probabilities of making errors are analyzed. In this section, the costs of misrecognitions are considered. We will further analyze the relationships between the decision point and expected costs.

Misrecognition/error costs allow us to specify the relative importance of different kinds of errors. In fact, many applications have unequal misrecognition costs. In our previous work [41] while studying the behaviors of the supervised feature selection algorithm, we noticed a trade-off of a smaller and bigger number of noise-free data preferences among various algorithms. As a consequence of this trade-off, different costs should be determined for different errors. A smaller number of noise-free data yields a higher type 1 error cost compared to the type 2 error cost.

The various misrecognition costs are defined by a cost matrix. The cost matrix reflects the domain specific costs representing the cost sensitive model in the critical medical domain. Therefore, associative costs for a different type of error are finalized by the domain expert keeping the clinical context and consequences in mind.

As shown in Table 1, cost matrix C usually has the following structure, wherein the cost matrix rows correspond to predicted results, while columns correspond to actual results, i.e., row/column = predict/actual.

Table 1. Cost matrix of the mislabeled data filter.

	Actual Mislabeled	Actual Noise-Free
Predict mislabeled and eliminate	C(0,0) = C_{00}	C(0,1) = C_{01}
Predict noise-free and retain	C(1,0) = C_{10}	C(0,0) = C_{11}

For correctly classified mislabeled (or noise-free) data, the cost is zero, and hence, it is normally assumed that $C_{00} = C_{11} = 0$ in the above matrix. With this assumption, the expected cost of an MVFilter is:

$$ExpectedCost_{MVFilter} =$$

$$P(E1_{MVFilter})C_{01} + P(E2_{MVFilter})C_{10}$$

As Section 3.1 shows, $P(E1_{MVFilter})$ and $P(E2_{MVFilter})$ are correlated with the decision point value. Therefore, $ExpectedCost_{MVFilter}$ is determined by both the decision point value and the cost matrix. If the cost matrix is fixed, then $ExpectedCost_{MVFilter}$ is only influenced by the decision point value. Therefore, the cost concerned optimal decision point should be:

$$ODP = \arg\min_{m=1:j}(P(E1_{MVFilter})C_{01} + P(E2_{MVFilter})C_{10})$$

In this equation, if $C_{01} \gg C_{10}$, $P(E1_{MVFilter})$ will be the dominant factor to determine ODP value. The ODP is the decision point that can minimize $P(E1_{MVFilter})$. From the analysis in Section 3.1, we know that it is highly probable that $ODP = t$. On the other hand, if $C_{10} \gg C_{01}$, $P(E2_{MVFilter})$ will be the dominant factor to determine the OPD value. In this case, it is likely that $ODP = 1$.

It should be noted that the ODP can be determined from the above analysis only in some extreme cases (for example, when $C_{01} \gg C_{10}$ or $C_{10} \gg C_{01}$). However, for the other cases, directly calculating the ODP is extremely difficult. In addition, the above equation of the ODP is obtained by making several assumptions. Therefore, it is not very useful to calculate the ODP value through mathematically inferring since the calculated ODP is influenced by the assumptions.

4. Novel Approach to Determine the Optimal Decision Point

In this section, we present our approach that can select the optimal decision point for an MVFilter by considering both cost information and the dataset itself.

Given a noisy training dataset, we define that the ODP is the value that can minimize the expected cost of an MVFilter. As pointed out in Section 3, mathematically inferring the ODP is difficult. Therefore, instead of directly inferring, we try to estimate the ODP implicitly.

For a noisy training dataset E, if we already know which data in E are mislabeled, it is trivial to decide the ODP. We just need to explore all the possible decision points. The OPD will be the point that minimizes the overall costs of misrecognitions.

Of course, the mislabeled data distribution in E is unknown since our mission is to identify mislabeled data from E. However, if there exists another noisy dataset E′ similar to E and with a known mislabeled data distribution, then we could implicitly estimate ODP from E′ instead of E since their ODPs should be similar.

This actually is the key idea of our approach. Given a noisy dataset E to handle, we will generate another dataset E′. The new generated E′ requires: (1) E′ and E are from the same/similar data distribution, and (2) the mislabeled data distributions in E and E′ are similar. If such an E′ could be generated, we can easily get the ODP based on E′ since the mislabeled data distribution in E′ is known.

In many real applications, in addition to the noisy dataset E, usually another validation dataset E_{nf} is available. E_{nf} contains only noise-free data and coming from the same data distribution as E. As there are no mislabeled data in E_{nf}, the artificial erroneous labels are put into E_{nf}. Here, we assume that the prior knowledge of the noise ratio in E is available, which is used to determine the erroneous

labels put into E_{nf}. Through the above procedures, E_{nf} can be converted to E'. The optimal decision point from E' can be used to estimate the actual decision point in E.

As the actual mislabeled data distribution in E is not available, we put erroneous labels in E_{nf} in a random manner based on the prior noise ratio information. Although the mislabeled data in E are also stochastic, the mislabeled data distribution in E and E' can have a great difference. In this case, the ODP value obtained from E' is actually not optimal for E. To solve this problem, the ODP is estimated several times. This method uses the *numIter* parameter to control the specified number of iterations. Each time E' changes since random erroneous labels are put into E_{nf}. For each time, all the possible decision points (from one to t) will be explored, and accordingly, the cost of misrecognition is recorded. The average cost of each decision point value is obtained by taking the mean value of this decision point multiple times the misrecognition costs. Finally, the decision point having the least average cost is selected as the optimal decision point. The details of our algorithm are shown in Algorithm 3.

Algorithm 3 Optimal decision point estimation for MVFilter.

Algorithm: Searching optimal decision point for MVFilter
Input: E (training set), E_{nf} (noise-free dataset)
Parameter: numIter (number of iterations to search ODP), noiseRatio (the noise ratio in E), MVFilter (the multiple-voting based filter algorithm), t (number of single-voting filters in MVFilter), C (cost matrix)
Output: ODP (optimal decision point)
(1) *costMatrix* $\leftarrow \emptyset$
(2) for i = 1,..., numIter do
(3)   randIndex$\leftarrow$RandomPermutation($\left|E_{nf}\right|$)
(4)   noiseIndex$\leftarrow$randIndex(1:$\left|E_{nf}\right| \times noiseRatio$)
(5) $E' \leftarrow generateNoise(E_{nf}, noiseIndex)$
(6)   *costVector* $\leftarrow \emptyset$
(7)   for m = 1,..., t do
(8)   noiseIndexDetected$\leftarrow$MVFilter(E', m)
(9)   index$\leftarrow$InterSection(noiseIndex, noiseIndexDetected)
(10)   indexE1$\leftarrow$noiseIndexDetected\index
(11)   indexE2$\leftarrow$noiseIndex\index
(12) $cost \leftarrow |indexE1| \times C_{01} + |indexE2| \times C_{10}$
(13)   costVector(m) $\leftarrow$cost
(14)   end for
(15)   costMatrix = [costrMatrix; costVector];
(16)   end for
(17) $ODP \leftarrow \underset{m=1:t}{\arg\min}(\text{costMatrix}(1:end, m))$

In Algorithm 3, it is assumed that another noise-free dataset E_{nf} exists, which has the same distribution as E. Usually in a training dataset, some labels are certainly correct. These partial noise-free data are also used as a validation dataset in many applications. However, for a few applications, if E_{nf} is unavailable, then this algorithm cannot be used directly. To solve this problem, we can directly use an MVFilter to mine the noise-free data from E. In this case, the loose noise detection policy is preferred by MVFilter. To generate E_{nf}, the main concern is to make less E2 errors. Therefore, a small decision point value (for example, one) should be used by MVFilter. By this method, E_{nf} can be collected from E. Then, Algorithm 3 can be used. The parameter noiseRatio in Algorithm 3 should also be noted. This parameter represents the noise ratio in E (mislabeled percentage of E). It is used to decide the number of erroneous labels to generate in E'. Here, we assume this is prior knowledge. For many applications, through years of experience, the rough noise ratio in a noisy training set is usually known. If this value is totally unknown, we also provide a solution. This parameter can be estimated from E by using an MVFilter. To estimate this parameter more accurately, MVFilter should select a decision point that considers the E1 and E2 error simultaneously. The value t/2 is a reasonable decision point since this decision point usually has a good trade-off between E1 and E2 errors.

5. Experimental Work

In this section, a set of experiments is conducted to verify the effectiveness of our proposed approach. To test its performance, several representative single-voting and multiple-voting based filters are used. SVFilters include the majority filter (MF) and consensus filter (CF) [27,28]. MF based MVFilters include MF_1, MF_{MF}, and MF_{CF} [40]. CF based MVFilters include CF_1, CF_{MF}, and CF_{CF} [40]. Suppose the number of SVFilters in an MVFilter is t. In MF_1 and CF_1, the decision point is 1. In MF_{MF} and CF_{MF}, the decision point is t/2, while in MF_{CF} and CF_{CF}, the decision point is t. When the decision point is determined by our approach, the MF based MVFilter is denoted by MF_{ODP} and the CF based MVFilter is denoted by CF_{ODP}. When filtering noises, the costs incurred by MF_{ODP} and CF_{ODP} will be compared to other methods. If our approach is effective, MF_{ODP} and CF_{ODP} should incur less cost compared to other methods.

Six bioinformatics datasets from the UCI repository were used in this experiment. Information on these datasets is tabulated in Table 2, where pos/neg presents the percentage of the number of positive examples against that of negative examples.

Table 2. Datasets used in the experiment. pos, positive; neg, negative.

Dataset	No. of Features	No. of Instances	pos/neg
Heart	14	270	55.6%/44.4%
Wdbc	30	569	62.7%/37.3%
Wpbc	33	198	76.3%/23.7%
Spect	22	267	79.4%/20.6%
Spect1	44	267	79.4%/20.6%
Promoter	57	106	50%/50%

An SVFilter (referring to Algorithm 1) is configured as follows: the number of subsets is 3 (n = 3); three learning algorithms are used (y = 3) including naive Bayes, decision tree, and 3-nearest neighbor. The configurations of an MVFilter (referring to Algorithm 2) are basically identical to the SVFilter configurations. One additional parameter in MVFilter is the number of SVFilters, which equals nine in the experiments (t = 9). Our proposed algorithm (referring to Algorithm 3) is based on MVFilter. Its additional parameter is the number of iterations to search for ODP. Here, it equals ten (numIter = 10).

The experiments were performed on each benchmark dataset by dividing it into a training set and test set. The filter algorithms were applied to each training set to remove the mislabeled data. Test data were only used by our algorithm, which is represented as E_{nf} in Algorithm 3. It is important to clarify here that domain experts were involved to establish the noise-free benchmark dataset, which included the desired labels finalized after coming to a common consensus.

Making the cost value as a baseline computation, the performance of each filter algorithm was evaluated against each dataset D using the following steps:

1. Evaluating the performance of each filter using three trials derived from the threefold cross-validation of D. For each trail, 2/3 of D or Tr were used for the training set. We purposely changed some correct labels in the Tr using the predefined mislabeled ratio to generate the mislabeled data. For this purpose, three different mislabeled ratios were used: 10%, 20%, and 30%. As an example, for a 10% mislabeled ratio, 10% of the samples from Tr were randomly selected and then the correct labels changed.
2. The average cost of each algorithm was calculated by taking the mean cost of errors for each filter of the three trails.
3. In order to avoid the influence of the partitioning of D on the generated mislabeled data, we considered ten cost values retrieved from each experiment conducted ten times (i.e., repeating the previous two steps ten times).

4. Finally, the reported cost value was obtained as the mean of these ten values.

5.1. Experimental Investigation

Next, the experimental results of each dataset will be presented. Table 3 shows the comparisons of each filter in terms of cost on the Heart dataset. This table consists of three parts corresponding to three noise ratios (10%, 20%, and 30%). Under each noise ratio, the experiments were based on nine different cost matrices. Here, it was assumed that $C_{00} = C_{11} = 0$, so only C_{01} and C_{10} were needed to define a cost matrix. For example, in the second row of Table 3, 1:1 means $C_{01} = C_{10} = 1$, while 1:20 means $C_{01} = 1, C_{10} = 20$. The last column in Table 3, Ave., represents the average cost of each filter based on all nine cost matrices.

Table 3 shows that for all three noise ratios, CF_{ODP} had the lowest average cost among all the CF based filters. Likewise, MF_{ODP} was the best one among all the MF based filters. Moreover, under all the noise ratios and cost matrices, CF_{ODP} and MF_{ODP} outperformed other filters in most cases. This was in contrast to the other filters that heavily depend on cost matrices. For example, CF_{CF} showed outstanding performance when $C_{01} > C_{10}$, but its performance decreased dramatically when C_{10} increased. When the correlation between the cost and noise ratio was considered, we found that the cost of all the filters increased with noise ratio growth. However, compared with other filters, the cost increases of CF_{ODP} and MF_{ODP} were slow. In detail, when the noise ratio grew from 10% to 30%, the cost increase of CF_{ODP} was 44, MF_{ODP} was 12, while the cost increase of other filters was fast (for example, 97 for CF_1 and 102 for MF_1). By further comparing CF_{ODP} and MF_{ODP}, we found that under this dataset, CF_{ODP} had a smaller average cost value. However, with the noise ratio increasing, the performance difference between them became small.

Table 4 shows the cost comparisons of each filter based on the Wdbcdataset. The experimental conclusions in Table 4 are similar to those of Table 3. In most cases (under different noise ratios and cost matrices), CF_{ODP} and MF_{ODP} were the winners. In addition, their advantages were more obvious when the noise ratio and cost value increased. When the noise ratio was 10%, CF_{ODP} outperformed MF_{ODP}. However, they showed similar performance when the noise ratio grew.

Table 5 presents the experimental results based on the Wpbc dataset. Similar to the experimental conclusions from Tables 3 and 4, our approach could effectively improve the performance of the MF and CF based filters. In addition, CF_{ODP} and MF_{ODP} consistently worked well in different cases. Except for CF_{ODP} and MF_{ODP}, the performance of the other filters usually had a dramatic decline when the noise ratio increased. Moreover, other filters had obvious performance changes when the relationship of C_{01} and C_{10} changed. For example, MF_{MF} worked well when $C_{10} > C_{01}$, but its performance became poor when $C_{01} > C_{10}$.

Tables 6–8 show the experimental results on the datasets of Spect, Spect1, and Promoter. Similar to the above analysis, these three tables clearly indicate the superiority of CF_{ODP} and MF_{ODP}.

Table 3. Cost comparisons on Heart. Ave., average; CF, consensus filter; MF, majority filter.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
				C_{01}:C_{10} (10% Noise Ratio)						
CF	22	33	45	74	132	54	86	167	328	105
CF_1	34	39	44	56	81	98	161	320	638	164
CF_{MF}	22	34	46	77	138	52	83	161	315	103
CF_{CF}	17	40	64	123	241	27	37	62	112	80
CF_{ODP}	17	35	44	60	79	27	37	62	112	52
MF	34	41	48	65	100	96	157	311	619	164
MF_1	59	63	67	76	95	174	288	574	1146	282
MF_{MF}	31	37	43	59	90	87	143	282	561	148
MF_{CF}	20	32	44	74	134	49	78	150	295	98
MF_{ODP}	20	34	43	63	87	49	78	150	295	91
				C_{01}:C_{10} (20% Noise Ratio)						
CF	29	62	95	177	342	55	81	145	274	140
CF_1	39	53	67	102	172	104	169	331	656	188
CF_{MF}	26	55	83	153	295	51	76	137	261	126
CF_{CF}	29	82	136	269	536	33	38	49	71	138
CF_{ODP}	27	53	66	102	172	34	38	49	71	68
MF	41	58	75	118	203	105	170	331	653	195
MF_1	70	77	84	101	136	204	338	672	1341	336
MF_{MF}	36	52	67	106	184	93	151	293	579	174
MF_{CF}	27	58	89	166	320	50	73	131	247	129
MF_{ODP}	28	51	68	99	139	50	73	131	247	98
				C_{01}:C_{10} (30% Noise Ratio)						
CF	42	101	160	308	604	68	93	156	283	202
CF_1	50	73	96	153	268	128	205	399	787	240
CF_{MF}	39	97	156	301	592	59	80	130	231	187
CF_{CF}	48	142	236	471	941	51	53	59	71	230
CF_{ODP}	38	75	96	153	268	52	53	59	71	96
MF	49	80	111	189	344	115	181	347	678	233
MF_1	80	88	95	113	150	234	387	770	1537	384
MF_{MF}	44	72	100	169	308	104	164	315	616	210
MF_{CF}	41	105	168	328	647	59	76	121	210	195
MF_{ODP}	39	71	89	116	150	59	76	121	210	103

Table 4. Cost comparisons on Wdbc.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	16	29	41	72	134	36	56	105	204	77
CF_1	21	26	30	42	65	59	96	190	378	101
CF_{MF}	14	24	34	59	109	33	52	99	194	69
CF_{CF}	16	40	63	123	242	24	31	51	90	75
CF_{ODP}	14	23	28	43	70	25	33	56	100	43
MF	24	28	32	44	66	66	109	216	429	113
MF_1	38	40	43	49	61	112	185	369	737	182
MF_{MF}	20	25	29	40	63	57	93	184	367	98
MF_{CF}	15	23	31	51	91	36	57	111	217	70
MF_{ODP}	15	24	30	45	60	36	57	111	217	66
$C_{01}:C_{10}$ (20% Noise Ratio)										
CF	26	64	102	198	389	39	52	86	152	123
CF_1	23	36	49	82	147	55	87	167	327	108
CF_{MF}	24	57	91	175	342	37	51	86	155	113
CF_{CF}	50	148	246	490	979	54	57	64	80	241
CF_{ODP}	21	36	49	82	147	38	49	65	81	63
MF	27	40	52	83	146	70	112	218	431	131
MF_1	69	75	81	96	126	200	331	660	1316	328
MF_{MF}	22	30	39	62	106	56	90	176	347	103
MF_{CF}	23	55	88	170	333	35	47	79	141	108
MF_{ODP}	17	29	40	68	117	34	47	79	141	64
$C_{01}:C_{10}$ (30% Noise Ratio)										
CF	50	133	217	426	844	66	82	123	204	238
CF_1	28	48	69	121	224	62	96	183	355	132
CF_{MF}	43	116	190	373	740	56	68	100	163	205
CF_{CF}	90	267	445	888	1775	92	95	101	113	429
CF_{ODP}	28	48	69	121	224	52	71	100	111	91
MF	42	64	85	139	247	105	169	326	642	202
MF_1	85	91	97	110	138	251	416	829	1656	408
MF_{MF}	28	44	59	97	174	69	110	213	419	135
MF_{CF}	45	121	196	386	765	59	72	107	176	214
MF_{ODP}	26	43	57	90	147	51	76	107	176	86

Table 5. Cost comparisons on Wpbc.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
	$C_{01}:C_{10}$ (10% Noise Ratio)									
CF	22	39	56	98	182	49	76	143	277	105
CF_1	45	50	55	68	94	131	216	429	856	216
CF_{MF}	19	34	49	86	160	41	64	121	234	90
CF_{CF}	14	40	66	131	261	16	19	24	35	67
CF_{ODP}	14	35	52	65	91	16	19	24	35	39
MF	39	45	51	67	98	110	182	361	719	186
MF_1	80	82	83	87	95	239	398	795	1590	383
MF_{MF}	36	40	45	56	78	103	171	339	676	171
MF_{CF}	18	35	53	96	183	37	56	103	198	87
MF_{ODP}	18	35	45	63	90	37	56	103	198	72
	$C_{01}:C_{10}$ (20% Noise Ratio)									
CF	32	67	102	190	366	61	90	162	306	153
CF_1	50	68	86	131	221	131	212	416	822	237
CF_{MF}	28	65	101	192	374	48	68	117	216	134
CF_{CF}	27	79	130	260	519	29	31	36	46	128
CF_{ODP}	26	60	82	131	221	30	33	36	46	74
MF	45	65	84	132	229	117	188	366	723	217
MF_1	78	84	90	104	133	229	380	757	1512	374
MF_{MF}	42	60	79	126	219	106	170	332	654	199
MF_{CF}	27	67	107	207	407	42	57	94	169	131
MF_{ODP}	28	59	77	102	132	42	57	94	169	85
	$C_{01}:C_{10}$ (30% Noise Ratio)									
CF	40	95	150	286	560	67	93	158	290	193
CF_1	53	73	92	141	239	140	226	443	876	254
CF_{MF}	38	97	157	306	603	53	69	109	188	180
CF_{CF}	39	116	194	387	774	39	40	41	43	186
CF_{ODP}	37	77	92	141	239	39	40	41	43	83
MF	50	80	110	185	335	120	190	365	715	239
MF_1	75	79	83	94	115	220	365	728	1453	357
MF_{MF}	48	74	101	168	301	116	185	357	700	228
MF_{CF}	39	102	165	324	640	52	66	101	169	184
MF_{ODP}	39	75	82	94	115	52	66	101	169	88

Table 6. Cost comparisons on Spect.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
				$C_{01}{:}C_{10}$ (10% Noise Ratio)						
CF	23	39	56	98	181	51	79	151	293	108
CF_1	39	50	60	86	139	108	176	347	690	188
CF_{MF}	19	37	55	100	190	40	61	113	218	93
CF_{CF}	16	42	68	133	263	22	28	43	73	76
CF_{ODP}	16	40	54	93	139	22	28	43	73	56
MF	42	50	58	78	118	117	192	380	755	199
MF_1	71	74	78	87	104	208	346	690	1377	337
MF_{MF}	36	43	51	70	108	100	164	325	646	171
MF_{CF}	20	37	54	98	184	42	64	120	231	94
MF_{ODP}	20	39	51	69	92	42	64	120	231	81
				$C_{01}{:}C_{10}$ (20% Noise Ratio)						
CF	34	71	108	201	386	65	95	172	325	162
CF_1	48	69	90	144	250	122	196	381	751	228
CF_{MF}	31	69	107	203	394	55	79	138	257	148
CF_{CF}	31	90	148	295	588	34	38	46	63	148
CF_{ODP}	32	68	95	144	250	36	38	46	63	86
MF	45	63	81	125	214	119	192	374	740	217
MF_1	77	83	89	104	134	226	375	747	1492	370
MF_{MF}	42	60	77	121	209	108	175	341	673	201
MF_{CF}	30	72	113	216	423	50	70	118	216	145
MF_{ODP}	31	60	78	108	134	50	70	118	216	96
				$C_{01}{:}C_{10}$ (30% Noise Ratio)						
CF	49	113	177	336	655	83	117	203	374	234
CF_1	59	89	118	193	342	147	234	454	893	281
CF_{MF}	45	109	174	335	657	71	96	160	288	215
CF_{CF}	51	148	246	490	978	55	59	69	89	243
CF_{ODP}	46	89	118	193	342	55	59	69	89	118
MF	56	89	122	203	367	136	216	416	816	269
MF_1	89	100	111	137	191	258	426	846	1688	427
MF_{MF}	54	88	123	208	379	128	203	388	759	259
MF_{CF}	45	115	186	361	712	65	86	136	237	216
MF_{ODP}	45	86	109	137	191	65	86	136	237	121

Table 7. Cost comparisons on Spect1.

	$C_{01}:C_{10}$ (10% Noise Ratio)									
	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	28	45	62	104	189	67	107	205	402	134
CF_1	50	57	64	82	117	142	234	464	924	237
CF_{MF}	23	41	60	106	198	50	77	145	280	109
CF_{CF}	19	48	78	153	301	26	33	52	88	89
CF_{ODP}	19	42	59	85	117	26	33	52	88	58
MF	47	56	65	89	135	131	215	426	847	223
MF_1	82	84	87	95	109	242	402	803	1604	390
MF_{MF}	41	50	59	80	124	115	189	373	742	197
MF_{CF}	23	42	60	106	198	51	79	148	287	110
MF_{ODP}	23	43	60	83	120	51	79	148	287	99
	$C_{01}:C_{10}$ (20% Noise Ratio)									
CF	35	76	116	217	419	65	95	170	320	168
CF_1	51	69	87	131	220	136	221	433	858	245
CF_{MF}	34	78	122	232	452	57	80	138	253	160
CF_{CF}	34	98	162	323	644	37	40	48	63	161
CF_{ODP}	33	66	88	131	220	37	40	48	63	81
MF	53	75	97	152	262	137	222	432	853	254
MF_1	86	92	99	114	145	252	419	834	1665	412
MF_{MF}	47	68	89	142	247	119	191	371	731	222
MF_{CF}	32	77	123	237	465	50	68	114	205	152
MF_{ODP}	32	65	85	112	143	50	68	114	205	97
	$C_{01}:C_{10}$ (30% Noise Ratio)									
CF	53	125	198	379	741	87	120	204	372	253
CF_1	71	101	131	205	354	184	296	577	1139	340
CF_{MF}	49	128	207	405	800	68	86	133	226	234
CF_{CF}	53	158	263	524	1048	55	57	61	70	254
CF_{ODP}	51	104	131	205	354	54	56	62	70	121
MF	66	105	143	240	433	159	252	485	950	315
MF_1	102	109	115	131	164	301	499	994	1985	489
MF_{MF}	61	99	136	229	416	147	232	445	872	293
MF_{CF}	52	136	221	432	854	71	90	138	233	247
MF_{ODP}	52	103	125	131	164	71	90	138	233	123

Table 8. Cost comparisons on Promoter.

	$C_{01}:C_{10}$ (10% Noise Ratio)									
	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	10	19	29	52	99	20	31	57	109	47
CF_1	23	26	29	38	54	65	107	212	422	108
CF_{MF}	8	19	29	54	106	15	22	38	72	40
CF_{CF}	7	22	37	73	147	7	7	7	7	35
CF_{ODP}	8	19	29	37	54	8	7	7	7	20
MF	20	24	29	41	64	55	90	177	351	94
MF_1	43	44	44	46	49	128	213	426	851	205
MF_{MF}	17	20	23	30	45	48	80	158	315	82
MF_{CF}	8	18	29	55	107	13	18	31	56	37
MF_{ODP}	8	18	23	32	41	13	18	31	56	27
	$C_{01}:C_{10}$ (20% Noise Ratio)									
CF	16	34	51	96	185	30	44	79	149	76
CF_1	26	32	39	54	85	72	118	233	463	125
CF_{MF}	13	33	53	101	199	21	28	46	83	64
CF_{CF}	15	44	72	145	289	15	16	17	19	70
CF_{ODP}	14	34	40	54	85	15	16	17	19	33
MF	24	32	41	63	106	63	102	199	393	114
MF_1	42	43	44	47	52	124	207	413	825	200
MF_{MF}	22	31	40	62	107	58	94	183	362	107
MF_{CF}	14	35	56	108	212	20	27	44	77	66
MF_{ODP}	14	29	44	46	50	20	27	44	77	39
	$C_{01}:C_{10}$ (30% Noise Ratio)									
CF	22	49	76	144	279	38	54	95	176	104
CF_1	32	44	57	88	150	83	133	261	515	151
CF_{MF}	20	52	84	164	324	29	37	58	100	97
CF_{CF}	21	64	106	212	424	21	21	21	21	101
CF_{ODP}	21	44	57	88	150	21	21	21	21	49
MF	27	43	58	98	177	65	103	198	388	128
MF_1	41	45	48	56	73	120	199	397	793	197
MF_{MF}	24	38	52	88	159	57	90	174	340	114
MF_{CF}	20	54	88	173	343	25	30	43	68	93
MF_{ODP}	20	40	48	56	73	25	30	43	68	45

Several important conclusions can be drawn by summarizing the above evaluation results:

(1) Selecting the optimal decision point by our approach could effectively improve the performance of an MVFilter. (2) CF_{ODP} and MF_{ODP} adapted to various noise ratios. In particular, even in a high noise ratio environment, the cost increases of CF_{ODP} and MF_{ODP} were not great. (3) Under different cost matrices, CF_{ODP} and MF_{ODP} consistently outperformed other filters. The advantages of CF_{ODP} and MF_{ODP} were more obvious when the difference between C_{01} and C_{10} was big. (4) Given a noisy training dataset, our proposed approach proved to be effective under different noise ratios and cost matrices if two conditions hold: (a) the noise ratio of this dataset is known; (b) there exists another noise-free training dataset that is drawn from the same distribution as this noisy dataset.

5.2. Extended Experimental Investigation

As pointed out above, our approach was verified to work well if the noise ratio and additional noise-free dataset were available. To further confirm the usability of our approach, we evaluated it in an environment where the two kinds of information were not available, that is the noisy training dataset E was the only available information.

The noise ratio was estimated by the CF_{MF} algorithm. As an MVFilter, CF_{MF} consists of t consensus filters. The decision point here equals t/2. In other words, if at least t/2 CFs identify data as mislabeled, then CF_{MF} will regard that these data are mislabeled. For a noisy training dataset E, if n data are identified by CF_{MF}, then the estimated noise ratio is $n/|E|$. The parameter configurations of CF_{MF} were consistent with before (referring to the beginning of Section 4).

The noise-free dataset was obtained by applying CF_1 algorithm on E. CF_1 consists of t consensus filters. If at least one of CF identify data as mislabeled, then CF_1 will regard these data as mislabeled. Conceptually, the noise detection is loose, which aims to remove all the potential mislabeled data. Suppose the noises recognized by CF_1 are A. Then, the noise-free dataset is the subset of E, which excludes A. The configurations of CF_1 were in accordance with above experiments.

Tables 9–14 show the experimental results on the benchmark datasets. Under all five datasets and all the noise ratios, CF_{ODP} and MF_{ODP} still showed outstanding performance, which defeated other filters in most cases. When compared to the experiment results in Tables 3–8, we found that the performances of CF_{ODP} and MF_{ODP} had a certain extent of degradation in a few cases. However, in general, the performance change was moderate. This indicates that even without a noise ratio and an additional noise-free dataset, our proposed approach still worked well. One of the reasons was that in our approach, the mislabeled data distribution was estimated multiple times. Although the estimated mislabeled distribution for each time might be distorted, their fusion approached the real distribution. Then, the estimated ODP was also close to the real optimal decision point.

The two independent experimental evaluations in Sections 5.1 and 5.2 proved that our proposed approach was effective and able to improve the performance of any MVFilter by selecting the optimal decision point. In particular, in the high noise ratio and high cost values, our approach showed significant improvements compared to other filters.

Table 9. Cost comparisons on Heart in the case that the noise ratio and noise-free dataset are unavailable.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
	$C_{01}:C_{10}$ (10% Noise Ratio)									
CF	23	37	50	82	148	57	91	175	344	112
CF_1	34	39	45	57	83	98	161	320	638	164
CF_{MF}	21	34	47	79	144	51	80	154	302	101
CF_{CF}	16	39	62	119	233	24	33	55	98	75
CF_{ODP}	26	35	41	56	83	51	69	106	185	72
MF	36	42	48	63	93	101	166	330	656	170
MF_1	61	64	67	73	87	181	301	600	1199	293
MF_{MF}	31	36	41	55	81	87	143	283	563	147
MF_{CF}	21	32	44	73	131	51	81	156	306	99
MF_{ODP}	23	33	41	57	85	53	85	160	306	94
	$C_{01}:C_{10}$ (20% Noise Ratio)									
CF	28	57	86	158	302	55	82	149	283	133
CF_1	40	51	62	91	147	107	175	345	683	189
CF_{MF}	28	60	92	172	332	53	78	139	263	135
CF_{CF}	30	85	141	280	558	34	38	48	68	142
CF_{ODP}	29	52	66	96	146	53	69	81	98	77
MF	39	52	65	98	164	104	169	331	655	186
MF_1	67	72	77	88	112	197	327	651	1300	321
MF_{MF}	36	48	60	90	150	97	158	309	613	174
MF_{CF}	27	60	93	174	338	50	72	127	239	131
MF_{ODP}	30	48	60	84	119	55	78	127	239	93
	$C_{01}:C_{10}$ (30% Noise Ratio)									
CF	43	102	162	312	610	68	93	157	283	203
CF_1	45	68	91	147	261	114	182	352	694	217
CF_{MF}	40	99	158	307	603	59	79	129	227	189
CF_{CF}	48	143	239	476	952	50	51	55	63	231
CF_{ODP}	38	76	99	147	261	56	65	80	84	101
MF	51	79	108	178	319	125	199	384	754	244
MF_1	76	83	91	108	144	222	367	731	1459	365
MF_{MF}	43	69	95	160	290	103	162	312	611	205
MF_{CF}	40	103	167	326	644	56	72	112	192	190
MF_{ODP}	37	75	96	127	178	59	78	112	192	106

Table 10. Cost comparisons on Wdbc in the case that the noise ratio and noise-free dataset are unavailable.

	$C_{01}:C_{10}$ (10% Noise Ratio)									
	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	17	32	47	86	162	35	53	99	190	80
CF_1	21	29	36	53	89	57	93	182	361	102
CF_{MF}	16	29	42	75	140	34	52	98	189	75
CF_{CF}	19	50	81	159	315	27	34	52	89	92
CF_{ODP}	17	27	35	52	88	37	55	89	154	62
MF	22	28	34	49	79	61	100	196	390	107
MF_1	38	41	44	52	68	111	184	367	733	182
MF_{MF}	20	25	30	43	68	54	88	174	345	94
MF_{CF}	16	26	37	64	117	36	57	109	212	75
MF_{ODP}	16	25	31	43	67	36	57	109	212	66
	$C_{01}:C_{10}$ (20% Noise Ratio)									
CF	26	64	102	197	387	41	56	93	168	126
CF_1	22	34	46	75	134	54	86	167	328	105
CF_{MF}	22	52	83	159	311	35	48	81	146	104
CF_{CF}	47	138	228	454	906	51	55	65	85	226
CF_{ODP}	21	35	46	75	134	37	49	80	121	66
MF	28	39	50	78	133	72	116	226	446	132
MF_1	60	64	69	81	104	174	289	576	1149	285
MF_{MF}	22	31	39	59	100	59	95	186	369	107
MF_{CF}	21	48	74	142	276	35	50	86	158	99
MF_{ODP}	21	30	40	61	96	40	54	86	158	65
	$C_{01}:C_{10}$ (30% Noise Ratio)									
CF	56	152	248	489	970	71	86	125	201	266
CF_1	32	60	88	158	298	67	103	192	370	152
CF_{MF}	49	134	219	431	856	63	77	112	182	236
CF_{CF}	92	274	456	911	1821	93	95	99	107	438
CF_{ODP}	34	60	88	158	298	59	79	119	137	115
MF	46	78	109	188	346	106	167	318	620	220
MF_1	96	102	109	125	157	282	467	931	1859	459
MF_{MF}	30	49	67	113	205	72	114	219	429	144
MF_{CF}	48	131	214	422	838	61	74	107	173	230
MF_{ODP}	30	50	68	103	166	56	74	112	173	92

Table 11. Cost comparisons on Wpbc in the case that the noise ratio and noise-free dataset are unavailable.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
	C_{01}:C_{10} (10% Noise Ratio)									
CF	23	40	58	102	189	50	78	148	287	108
CF_1	43	51	59	78	117	121	199	395	786	205
CF_{MF}	21	39	57	103	194	44	68	127	245	100
CF_{CF}	15	40	66	131	260	18	22	31	48	70
CF_{ODP}	23	42	57	79	118	30	38	53	64	56
MF	41	49	56	74	111	117	192	380	757	197
MF_1	79	80	81	85	91	235	391	781	1561	376
MF_{MF}	38	46	54	73	112	107	176	348	693	183
MF_{CF}	19	38	56	102	194	39	59	108	207	91
MF_{ODP}	22	42	51	73	101	40	59	108	207	78
	C_{01}:C_{10} (20% Noise Ratio)									
CF	30	63	95	175	336	59	87	158	301	145
CF_1	52	65	79	113	181	142	232	457	907	247
CF_{MF}	29	66	102	193	375	51	73	128	238	140
CF_{CF}	26	77	128	256	512	28	29	32	39	125
CF_{ODP}	30	63	86	123	181	42	42	52	81	78
MF	45	61	77	116	195	121	196	383	759	217
MF_1	81	85	89	99	119	238	396	790	1578	386
MF_{MF}	44	59	74	112	188	118	191	374	741	211
MF_{CF}	27	64	102	195	382	44	61	103	188	130
MF_{ODP}	30	61	83	109	169	46	64	103	188	95
	C_{01}:C_{10} (30% Noise Ratio)									
CF	39	97	154	297	584	61	82	135	242	188
CF_1	53	81	109	178	317	132	210	406	798	254
CF_{MF}	37	96	155	301	595	53	69	108	187	178
CF_{CF}	40	117	194	388	774	41	43	48	56	189
CF_{ODP}	38	87	125	208	333	45	47	55	60	111
MF	51	82	113	191	346	121	192	368	720	243
MF_1	74	80	86	101	131	217	360	716	1430	355
MF_{MF}	49	80	110	187	340	116	183	351	686	234
MF_{CF}	38	101	164	320	634	52	66	101	171	183
MF_{ODP}	39	85	114	180	240	53	68	101	171	117

Table 12. Cost comparisons on Spect in the case that the noise ratio and noise-free dataset are unavailable.

	C$_{01}$:C$_{10}$ (10% Noise Ratio)									
	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	19	34	49	87	163	42	65	122	236	91
CF$_1$	37	45	54	75	117	102	168	331	658	176
CF$_{MF}$	21	37	53	93	173	46	72	136	264	99
CF$_{CF}$	17	43	70	137	270	24	31	48	82	80
CF$_{ODP}$	20	38	51	76	117	33	43	64	100	60
MF	40	48	56	77	118	110	181	359	713	189
MF$_1$	69	72	75	81	95	205	341	681	1361	331
MF$_{MF}$	36	45	53	74	116	100	164	323	642	173
MF$_{CF}$	20	36	52	91	170	44	68	129	250	96
MF$_{ODP}$	20	38	51	75	100	44	68	129	250	86
	C$_{01}$:C$_{10}$ (20% Noise Ratio)									
CF	35	75	115	215	415	64	94	168	316	166
CF$_1$	48	69	90	143	249	123	198	385	759	229
CF$_{MF}$	31	71	111	212	413	53	75	129	238	148
CF$_{CF}$	32	92	152	302	602	35	39	48	66	152
CF$_{ODP}$	32	69	95	149	260	38	42	48	65	89
MF	45	66	86	138	241	115	184	358	706	216
MF$_1$	78	86	93	112	150	226	375	746	1488	373
MF$_{MF}$	44	61	77	119	202	115	186	364	719	210
MF$_{CF}$	32	74	116	221	431	53	74	128	234	151
MF$_{ODP}$	32	62	82	112	164	53	74	128	234	105
	C$_{01}$:C$_{10}$ (30% Noise Ratio)									
CF	51	113	175	331	642	90	129	227	422	242
CF$_1$	61	87	112	175	302	159	256	499	986	293
CF$_{MF}$	45	105	166	319	623	73	101	172	313	213
CF$_{CF}$	51	150	250	498	995	53	56	62	74	243
CF$_{ODP}$	47	88	118	175	302	67	64	67	84	113
MF	61	91	122	198	350	153	244	473	931	291
MF$_1$	92	97	101	113	136	271	451	899	1796	440
MF$_{MF}$	50	79	108	180	324	120	191	368	721	238
MF$_{CF}$	45	117	189	369	729	63	80	125	214	214
MF$_{ODP}$	44	81	110	136	192	64	80	125	214	116

Table 13. Cost comparisons on Spect1 in the case that the noise ratio and noise-free dataset are unavailable.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
					C_{01}:C_{10} (10% Noise Ratio)					
CF	25	43	60	103	190	59	93	176	344	122
CF_1	45	51	56	70	98	130	216	428	854	217
CF_{MF}	24	43	61	109	203	52	81	152	294	113
CF_{CF}	17	45	73	144	285	23	29	43	72	81
CF_{ODP}	27	45	56	71	101	41	47	63	96	61
MF	45	53	61	82	123	126	207	410	815	213
MF_1	84	86	89	96	109	248	413	825	1648	400
MF_{MF}	41	49	57	77	117	115	190	375	746	196
MF_{CF}	23	41	59	104	194	50	77	145	280	108
MF_{ODP}	28	45	58	80	109	51	79	150	280	98
					C_{01}:C_{10} (20% Noise Ratio)					
CF	38	83	127	239	462	69	100	178	333	181
CF_1	52	70	89	134	225	138	224	439	869	249
CF_{MF}	35	79	123	234	455	59	84	147	271	165
CF_{CF}	35	102	169	336	670	38	41	49	65	167
CF_{ODP}	36	70	97	134	225	55	63	67	68	91
MF	55	78	102	161	278	140	226	441	870	261
MF_1	91	96	102	116	143	266	442	882	1761	433
MF_{MF}	46	66	87	138	240	117	189	367	724	219
MF_{CF}	34	79	124	236	460	57	80	137	251	162
MF_{ODP}	36	68	94	136	197	59	80	137	251	118
					C_{01}:C_{10} (30% Noise Ratio)					
CF	57	134	211	405	791	93	129	220	401	271
CF_1	68	105	141	232	414	168	268	517	1016	325
CF_{MF}	51	128	205	397	781	76	101	163	287	243
CF_{CF}	53	158	262	522	1044	56	59	65	79	255
CF_{ODP}	51	105	149	232	414	62	62	65	79	136
MF	65	106	147	249	454	155	245	470	920	313
MF_1	96	104	113	133	174	280	464	924	1844	459
MF_{MF}	58	93	127	213	385	140	222	427	837	278
MF_{CF}	49	129	209	409	809	67	84	129	218	234
MF_{ODP}	48	102	131	190	254	67	84	129	218	136

Table 14. Cost comparisons on Promoter in the case that the noise ratio and noise-free dataset are unavailable.

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
	C_{01}:C_{10} (10% Noise Ratio)									

	1:1	1:3	1:5	1:10	1:20	3:1	5:1	10:1	20:1	Ave.
CF	9	19	29	53	102	17	25	46	87	43
CF_1	22	25	29	38	56	62	102	203	404	105
CF_{MF}	7	15	24	47	91	11	15	26	47	31
CF_{CF}	7	21	34	69	138	7	7	7	7	33
CF_{ODP}	7	16	24	41	68	8	8	8	9	21
MF	18	24	29	43	71	49	79	156	309	87
MF_1	43	44	45	49	55	127	211	422	843	204
MF_{MF}	17	21	25	36	57	46	75	148	293	80
MF_{CF}	7	16	25	48	93	11	15	26	47	32
MF_{ODP}	7	15	22	35	53	11	15	26	47	26
	C_{01}:C_{10} (20% Noise Ratio)									
CF	15	34	53	101	197	26	37	64	118	72
CF_1	27	34	42	59	95	75	122	241	479	131
CF_{MF}	13	33	53	102	201	19	25	41	72	62
CF_{CF}	14	42	71	141	282	14	14	14	14	67
CF_{ODP}	13	30	43	65	100	15	14	14	14	34
MF	22	30	38	57	96	58	94	185	366	105
MF_1	42	44	46	51	61	123	204	408	814	199
MF_{MF}	20	29	37	58	101	53	85	165	326	97
MF_{CF}	13	33	53	102	201	20	26	42	74	63
MF_{ODP}	14	28	39	50	72	20	26	42	74	41
	C_{01}:C_{10} (30% Noise Ratio)									
CF	21	49	77	148	289	34	47	80	145	99
CF_1	28	40	51	79	136	73	118	231	457	135
CF_{MF}	19	50	82	160	317	26	32	49	82	91
CF_{CF}	21	64	106	212	425	22	22	22	23	102
CF_{ODP}	21	44	61	84	137	23	22	22	23	49
MF	27	42	58	96	173	65	104	200	392	128
MF_1	38	40	42	46	55	113	187	373	745	182
MF_{MF}	25	40	56	94	171	60	94	181	354	119
MF_{CF}	21	55	88	173	342	29	36	56	95	99
MF_{ODP}	21	46	55	83	93	29	36	56	95	57

6. Conclusions and Future Works

In mislabeled data detection, the multiple-voting based filter (MVFilter) is generally superior to the conventional single-voting based filter (SVFilter). However, one important unsolved issue in the MVFilter is how to choose the optimal decision point (ODP) to maximize its noise detection performance.

In this paper, a novel approach was proposed to solve this issue. This approach implicitly computed the ODP by estimating the mislabeled data distribution in the noisy training dataset. Our approach took a noisy dataset and a cost matrix as input, then output an ODP, which aimed to minimize the expected cost of errors. Note that minimizing cost was one important contribution of this work, because most existing works were not aware of the importance of cost. They just implicitly assumed that all errors were equally costly, but in most real applications, this is far from the case.

A set of experimental evaluations was conducted, which proved the effectiveness of our approach. With the aid of our approach, an MVFilter could effectively reduced the cost. In particular, in the difficult noise detection environment (when the noise ratio was high or cost was big), the advantages of our approach were more obvious. Furthermore, the proposed methodology could also be extended to a multi-class problem. One possible strategy is the naive way to divide the multi-class problem into several two-class problems, and then, the proposed approach can solve each two-class problem.

Although the clean dataset (i.e., validation cases) and cost matrix are available in most cases, the prior information of noise ratio is not easily available; therefore, the current solution needs to be improved to alleviate the prior information requirement. Therefore, in future work, we will focus on developing more elegant approaches to further improve the current proposed approach.

7. Availability of Data and Material

All the datasets are available at http://archive.ics.uci.edu/ml/datasets.html.

References

1. Guan, D.; Yuan, W.; Lee, Y.K. Nearest neighbor editing aided by unlabeled data. *Inf. Sci.* **2009**, *179*, 2273–2282. [CrossRef]
2. Van, J.; Khoshgoftaar, T.; Huang, H. The pairwise attribute noise detection algorithm. *Knowl. Inf. Syst.* **2007**, *11*, 171–190.
3. Van, J.; Khoshgoftaar, T. Knowledge discovery from imbalanced and noisy data. *Data Knowl. Eng.* **2009**, *68*, 1513–1542.
4. Zhu, X.Q.; Wu, X.D. Class noise vs. attribute noise: A quantitative study. *Artif. Intell. Rev.* **2004**, *22*, 177–210. [CrossRef]
5. Zhu, X.Q.; Wu, X.D.; Yang, Y. Dynamic classifier selection for effective mining from noisy data streams. In Proceedings of the Fourth IEEE International Conference on Data Mining, Brighton, UK, 1–4 November 2004; pp. 305–312.
6. Han, G.; Jiang, J.; Guizani, M.; Rodrigues, J.J.P.C. Green routing protocols for wireless multimedia sensor networks. *IEEE Wirel. Commun.* **2016**, *23*, 140–146. [CrossRef]
7. Han, G.; Que, W.; Jia, G.; Zhang, W. Resource Utilization-aware Energy Efficient Server Consolidation Algorithm for Green Computing in IIOT. *J. Netw. Comput. Appl.* **2017**. [CrossRef]
8. Jia, G.; Han, G.; Jiang, J.; Liu, L. Dynamic Adaptive Replacement Policy in Shared Last-Level Cache of DRAM/PCM Hybrid Memory for Big Data Storage. *IEEE Trans. Ind. Inform.* **2016**. [CrossRef]

9. West, M.; Blanchette, C.; Dressman, H.; Huang, E.; Ishida, S.; Spang, R.; Zuzan, H.; Olson, J.A., Jr.; Marks, J.R.; Nevins, J.R. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. USA* **2001**, *98*, 11462–11467. [CrossRef]

10. Hickey, R.J. Noise modelling and evaluating learning from examples. *Artif. Intell.* **2006**, *82*, 157–179. [CrossRef]

11. Pechenizkiy, M.; Tsymbal, A.; Puuronen, S.; Pechenizkiy, O. Class noise and supervised learning in medical domains: The effect of feature extraction. In Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems, Salt Lake City, UT, USA, 22–23 June 2006; pp. 708–713.

12. Bi, Y.; Jeske, D.R. The efficiency of logistic regression compared to normal discriminant analysis under class-conditional classification noise. *J. Multivar. Anal.* **2010**, *101*, 1622–1637. [CrossRef]

13. Nettleton, D.; Orriols-Puig, A.; Fornells, A. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif. Intell. Rev.* **2010**, *33*, 275–306. [CrossRef]

14. Zhang, J.; Yang, Y. Robustness of regularized linear classification methods in text categorization. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 190–197.

15. Opitz, D.; Maclin, R. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.* **1999**, *11*, 169–198. [CrossRef]

16. Dietterich, T.G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.* **2000**, *40*, 139–157. [CrossRef]

17. Ratsch, G.; Onoda, T.; Muller, K. Soft margins for AdaBoost. *Mach. Learn.* **2001**, *42*, 287–320. [CrossRef]

18. Bootkrajang, J.; Kaban, A. Classification of mislabelled microarrays using robust sparse logistic regression. *Bioinformatics* **2013**, *29*, 870–877. [CrossRef] [PubMed]

19. Gu, B.; Victor, S.S. A Robust Regularization Path Algorithm for v-Support Vector Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**. [CrossRef]

20. Saez, J.; Galar, M.; Luengo, J.; Herrera, F. A first study on decomposition strategies with data with class noise using decision trees. In *Hybrid Artificial Intelligent Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7209, pp. 25–35.

21. Beigman, E.; Klebanov, B.B. Learning with annotation noise. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1, Suntec, Singapore, 2–7 August 2009; pp. 280–287.

22. Sastry, P.S.; Nagendra, G.D.; Manwani, N. A team of continuousaction learning automata for noise-tolerant learning of half-spaces. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2010**, *40*, 19–28. [CrossRef]

23. Manwani, N.; Sastry, P.S. Noise tolerance under risk minimization. *IEEE Trans. Cybern.* **2013**, *43*, 1146–1151. [CrossRef]

24. Abellan, J.; Masegosa, A.R. Bagging decision trees on data sets with classification noise. In *Foundations of Information and Knowledge Systems*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 248–265.

25. Gu, B.; Sheng, V.S.; Tay, K.Y.; Romano, W.; Li, S. Incremental support vector learning for ordinal regression. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1403–1416. [CrossRef]

26. Abellan, J.; Moral, S. Building classification trees using the total uncertainty criterion. *Int. J. Intell. Syst.* **2003**, *18*, 1215–1225. [CrossRef]

27. Brodley, C.E.; Friedl, M.A. Improving autmated land cover mapping by identifying and eliminating mislabeled observations from training data. In Proceedings of the Geoscience and Remote Sensing Symposium, Lincoln, NE, USA, 31 May 1996; pp. 1379–1381.

28. Brodley, C.E.; Friedl, M.A. Identifying mislabeled training data. *J. Artif. Intell. Res.* **1999**, *11*, 131–167. [CrossRef]

29. Chaudhuri, B.B. A new definition of neighborhod of a point in multi-dimensional space. *Pattern Recognit. Lett.* **1996**, *17*, 11–17. [CrossRef]

30. Guan, D.; Yuan, W.; Lee, Y.K.; Lee, S. Identifying mislabeled training data with the aid of unlabeled data. *Appl. Intell.* **2011**, *35*, 345–358. [CrossRef]

31. John, G.H. Robust decision trees: Removing outliers from databases. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, Montréal, QC, Canada, 20–21 August 1995; pp. 174–179.

32. Marques, A.I. Decontamination of training data for supevised pattern recognition. In *Advances in Pattern Recognition*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1876, pp. 621–630.
33. Sánchez, J.S.; Barandela, R.; Marqués, A.I.; Alejo, R.; Badenas, J. Analysis of new techniques to obtain quality training sets. *Pattern Recognit. Lett.* **2003**, *24*, 1015–1022. [CrossRef]
34. Metxas, D.; Metaxas, D.; Fradkin, D.; Kulikowski, C.; Muchnik, I. Distinguishing mislabeled data from correctly labeled data in classifier design. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 15–17 November 2004; pp. 668–672.
35. Verbaeten, S.; Assche, A.V. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 317–325.
36. Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **1992**, *2*, 431–433. [CrossRef]
37. Wu, X.; Zhu, X.; Chen, Q. Eliminating class noise in large datasets. In Proceedings of the International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003; pp. 920–927.
38. Young, J.; Ashburner, J.; Ourselin, S. Wrapper methods to correct mislabeled training data. In Proceedings of the 2013 International Workshop on Pattern Recognition in Neuroimaging, Philadelphia, PA, USA, 22–24 June 2013; pp. 170–173.
39. Zhou, Z.H.; Jiang, Y. Editing training data for knn classifiers with neural network ensemble. In *Advances in Neural Networks*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3173, pp. 356–361.
40. Guan, D.; Yuan, W.; Ma, T.; Lee, S. Detecting potential labeling errors for bioinformatics by multiple voting. *Knowl.-Based Syst.* **2014**, *66*, 28–35. [CrossRef]
41. Yuan, W.; Guan, D.; Shen, L.; Pan, H. An empirical study of filter based feature selection algorithms using noisy training data. In Proceedings of the 4th IEEE International Conference on Information Science and Technology, Shenzhen, China, 26–28 April 2014; pp. 209–212.