mathematics

MDPI

Article

On the Suitability of Bagging-Based Ensembles with Borderline Label Noise

José A. Sáez [1,*] and José L. Romero-Béjar [1,2,3]

[1] Department of Statistics and Operations Research, University of Granada, Fuentenueva s/n, 18071 Granada, Spain; jlrbejar@ugr.es
[2] Instituto de Investigación Biosanitaria (ibs.GRANADA), 18012 Granada, Spain
[3] Institute of Mathematics of the University of Granada (IMAG), Ventanilla 11, 18001 Granada, Spain
* Correspondence: joseasaezm@ugr.es

Abstract: Real-world classification data usually contain noise, which can affect the accuracy of the models and their complexity. In this context, an interesting approach to reduce the effects of noise is building ensembles of classifiers, which traditionally have been credited with the ability to tackle difficult problems. Among the alternatives to build ensembles with noisy data, bagging has shown some potential in the specialized literature. However, existing works in this field are limited and only focus on the study of noise based on a random mislabeling, which is unlikely to occur in real-world applications. Recent research shows that other types of noise, such as that occurring at class boundaries, are more common and challenging for classification algorithms. This paper delves into the analysis of the usage of bagging techniques in these complex problems, in which noise affects the decision boundaries among classes. In order to investigate whether bagging is able to reduce the impact of borderline noise, an experimental study is carried out considering a large number of datasets with different noise levels, and several noise models and classification algorithms. The results obtained reflect that bagging obtains a better accuracy and robustness than the individual models with this complex type of noise. The highest improvements in average accuracy are around 2–4% and are generally found at medium-high noise levels (from 15–20% onwards). The partial consideration of noisy samples when creating the subsamples from the original training set in bagging can make it so that only some parts of the decision boundaries among classes are impaired when building each model, reducing the impact of noise in the global system.

Keywords: borderline noise; label noise; bagging; ensembles; robust learners; classification

MSC: 62R07

check for updates

Citation: Sáez, J.A; Romero-Béjar, J.L. On the Suitability of Bagging-Based Ensembles with Borderline Label Noise. *Mathematics* **2022**, *10*, 1892. https://doi.org/10.3390/math10111892

Academic Editors: Weihua Xu, Jinhai Li, Xibei Yang and José Antonio Sanz

Received: 1 May 2022
Accepted: 26 May 2022
Published: 1 June 2022

1. Introduction

Data acquisition and processing in statistical and data-mining applications are often subject to imperfections [1]. This fact may lead to the presence of errors or noise in datasets [2,3]. In classification [4], creating models from noisy data has several drawbacks, including the need for more time and samples to build the classifier [5,6]. Furthermore, both the accuracy and complexity of classifiers can be affected by modeling corrupted data [7,8].

Given the inconveniences caused by noise, previous works have raised the need for techniques to deal with it [9,10]. Thus, in the classification literature, two main options are contemplated for the treatment of noise: (i) the so-called robust learners [9,11], which involve modifications of existing algorithms to deal with errors; and (ii) the preprocessing of datasets with the aim of handling the noisy samples [2,10]. Despite the good results that both types of approaches can provide, they have some disadvantages [12,13]. The former require redesigning the algorithm associated with known classification methods, which in some cases is complex to perform. Furthermore, since the adaptation depends on such methods, it is not immediately applicable to other classification techniques [9]. On the other hand, methods following the second approach are often designed to detect noise with certain characteristics, and therefore the resulting data may be imperfect [12]. These facts show the importance of investigating other alternatives to reduce the impact of noise that allow satisfactory results when the previous approaches are impaired.

When dealing with complex datasets, several works have demonstrated that ensemble methods [14,15], which simultaneously use several classifiers, are an accurate way to overcome some of the difficulties in building models from the data. One of the best known approaches to create ensembles is bootstrap aggregating or, as it is more commonly known, bagging [16,17]. Given a dataset, it generates different versions using a bootstrap resampling procedure and builds a model on each of these subsets. Then, the outputs of all the available classifiers are combined to obtain a single final prediction for each sample [18].

Nevertheless, despite the popularity of bagging schemes to build ensembles, research works studying their behavior with noisy data are limited and use specific features [19,20]. Abellán et al. [19] focused on studying decision trees, analyzing the application of bagging of trees with imprecise probabilities compared to bagging of traditional decision trees. On the other hand, Khoshgoftaar et al. [20] compared the performance of several boosting and bagging techniques dealing with noisy datasets, only focusing on imbalanced and binary classification data. Furthermore, the above studies dealt with noise based on random mislabeling [20,21]: the samples to corrupt were chosen randomly, which represents an unlikely situation in real applications [22]. Recent works [23,24] have proposed other more advanced noise introduction models, which better represent the corruptions occurring in real-world datasets. They are based on the mislabeling of samples close to decision boundaries, where errors are more prone to occur [22]. These types of errors are more common in practice and more difficult for classification algorithms to detect and deal with [23]. There are works showing that, in real-world applications based on collaborative labeling, most of the differences between the labelers occur in the proximity of the decision boundaries [25,26]. A study on coronary disease classification [27] revealed that noise was generally caused by equipment measurement errors, which generated altered values in the proximity of the decision boundaries and led to incorrect labeling of the samples. Other works also reinforce the importance of labeling errors at decision limits [12,28]. Thus, Garcia et al. [28] analyzed a dataset in the field of ecology, in which they observed that certain alterations produced small errors in environmental characteristics, ultimately leading to mislabeling of the collected data. The importance of noisy samples at decision limits was also reflected in the field of noise filtering [12], in which the efficiency of noise filters was more notable when the dataset presented overlapping between classes.

This research delves into the above aspects, analyzing the behavior of bagging schemes when decision boundaries are affected by labeling errors. An experimental study based on the comparison of bagging against its baseline models dealing with borderline noise will be developed. For this, four robust classification methods will be considered: `C4.5` [11], `RIPPER` [29], `PART` [30], and `C5.0` [31]. Although the performance of these techniques with noise is usually known [32,33], research works analyzing their real behavior with borderline label noise are scarce, particularly if bagging is used. This work aims at verifying how these algorithms are affected by borderline noise and whether their robustness can be further increased by using bagging schemes. These methods, with and without bagging, will be used to create classifiers over 36 real-world datasets, both binary and multi-class, with different natures and characteristics. Recent borderline label noise models [23] will be employed in order to inject errors into these datasets considering nine noise levels (ranging from 0% to 40%, in steps of 5%), resulting in an experiment involving a total of 612 noisy datasets. The disparity between the individual models and bagging-based ensembles will be explored considering both their accuracy and robustness on each noisy dataset created. As support for the conclusions drawn from this study, the corresponding statistical tests [34] will be used on the results obtained. The datasets and the results of the experimentation carried out in this paper can be accessed through the webpage https://joseasaezm.github.io/bagbln/ (accessed on 1 May 2022). As a summary, the following are the main contributions offered by this research:

- Deepening the understanding of the impact of borderline label noise, which is more frequent in practice than the random label noise that is commonly studied, on the efficacy of traditional classification methods.
- Analysis of the behavior of bagging-based ensembles versus not considering them when dealing with borderline label noise, which has usually been overlooked in the literature.
- Study of the improvement of robustness to noise, through the use of specific measures [35], of methods traditionally considered robust when included in a bagging ensemble.
- Establishing the noise levels in the data where the use of bagging is most recommended, as well as the hypotheses that explain its good behavior with borderline label noise.

Note that this paper primarily focuses on analyzing the accuracy and robustness of classification methods with and without bagging when errors affect the labels of samples at decision boundaries. However, a study of the specific characteristics of the data (such as the overlapping level among classes, the imbalance ratio, and the dispersion degree of the samples, among others [36]) leading to a better behavior of bagging is not carried out (except for the level and type of noise, which are injected into the datasets in a controlled way).

The rest of this work is organized as follows. Section 2 contemplates the background associated with this paper, introducing the problem of noisy data in classification and the creation of ensembles using bagging. Then, Section 3 details the characteristics of the experiment that is carried out, and Section 4 focuses on the analysis of results. Finally, Section 5 concludes this paper, providing some ideas about future research.

2. Background

This section presents the background related to this paper. Section 2.1 introduces classification with noisy data, while Section 2.2 focuses on using bagging schemes for building ensembles.

2.1. Noisy Data in Classification

Because the source and input of data in real-world applications are often subject to imperfections, the data associated with them usually suffer from corruptions [6,37]. In classification problems, noise can impair classifiers by affecting their accuracy, complexity, and construction time [5,7]. In this context, there are two main types of noise found in the specialized literature [2,8]:

- Label noise [2,38]. This occurs when samples are labeled with incorrect class labels. Its origin is commonly associated with subjectivity during the labeling process, errors in data collection, or the use of inaccurate information for labeling [39,40].
- Attribute noise [41,42]. This is related to the imperfections occurring in the attributes of a classification dataset. This type of noise can come from various sources, such as streaming restrictions, detection device failures, and transcription errors [8].

Note that, although only two types of noise are distinguished, each of them can appear in multiple ways [10,42]. For example, label noise may occur only between certain classes [10], affect each of the classes unequally [43], or be located in certain areas [22], such as the decision boundaries analyzed in this paper. Something similar applies to attribute noise, as it can appear as small errors in the data following a Gaussian distribution [42] or more pronounced errors that can have a larger impact [44]. Among both types of noise, label noise, which is the focus of this research, is often more harmful to classifier performance than attribute noise because labels usually have more influence on model construction [32,33].

Since error is inherent in human nature and in most measurement instruments, there are many real-world applications where noisy data are typically present [45,46]. For example, label noise is common in medical applications [45], in which the information used to label each case may come from different tests whose results are unknown or imprecise. Another common application where label noise occurs is spam filtering [47], in which accidental clicks can cause samples to be mislabeled. On the other hand, attribute noise can be present in other types of applications, such as those involving voice recognition in call routing [48] or in the field of software engineering [46], where it can affect the software quality metrics.

In the context of noisy data, robustness [35] is the ability of a classification technique to create classifiers that are less affected by imperfect data. This fact implies that models created by robust algorithms from clean and noisy data are more similar. Robustness is a relevant issue when studying noisy data because it allows estimation of the performance of a technique when the characteristics of noise in a dataset are unknown. Examples of robust learners are `C4.5` [11], `RIPPER` [29], `PART` [30], and `C5.0` [31], which are considered in this paper. These algorithms incorporate pruning schemes to avoid overfitting the classifiers to errors. One of the contributions of this research is to analyze whether the usage of bagging improves the behavior of these algorithms traditionally considered robust to noise when dealing with borderline label noise.

2.2. Building Classification Ensembles Using Bagging

Ensemble methods [14,15] are based on creating several models from the training data. They have been postulated as an efficient alternative for complex problems, where the construction of different models from the data, in such a way that they complement each other, usually brings some advantages [49]. Thus, the usage of ensembles with respect to each of their components often implies improvements in classification performance, dynamic adaptation, and parallelization [50,51]. One of the best known and widely used approaches to building ensembles is bagging [16,17] (see Figure 1).

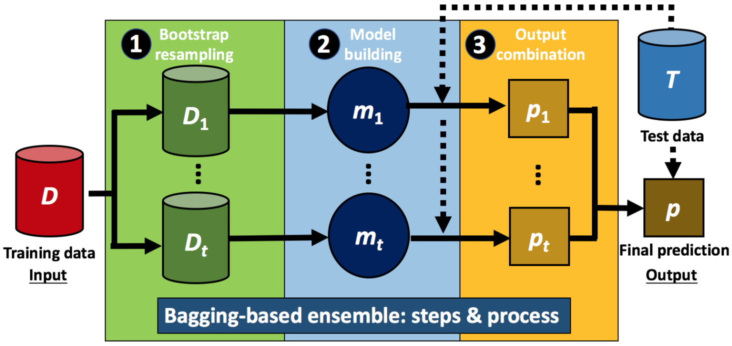

Figure 1. Main steps of bagging-based ensembles.

The operation of bagging-based ensembles is described below. Let D be a classification dataset with n samples. Bagging generates t different subsets $D_1, \ldots, D_t$ from D using a bootstrap resampling procedure [52]. Each subset D_k, $k \in \{1, \ldots, t\}$, is usually created by means of a random selection with replacement of n samples from the initial data D. This sampling procedure ensures that each subset D_k is independent of the others. Then, a model m_k, $k \in \{1, \ldots, t\}$, is built on each of these subsets D_k using a base classification algorithm. A phase of output combination is carried out to determine the class labels for new samples [52], in which each sample is evaluated by all the available classifiers obtaining t distinct predictions $p_1, \ldots, p_t$. The most used approach for output combination in the specialized literature is majority voting [14,19]. This is a simple but effective procedure in which each model within the ensemble casts a vote for one of the classes, and the most voted class is chosen as the final prediction.

3. Experimental Framework

This section details the characteristics of the experimental framework designed to analyze the efficacy of bagging schemes with borderline label noise. They are influenced by the experimental framework of other recent research works published in the field of classification with noisy data [10,42,53]. Sections 3.1 and 3.2 focus on describing the real-world datasets used and how labeling errors are induced in them. Then, Section 3.3 focuses on the classification methods. Section 3.4 presents the methodology employed for the analysis of results.

3.1. Real-World Datasets

The experimentation is based on 36 real-world datasets of different natures taken from the *UCI machine learning* and *KEEL-dataset* repositories (https://archive.ics.uci.edu/ and http://www.keel.es (accessed on 1 May 2022)). These are shown in Table 1, where *sa* refers to the number of samples, *at* to the number of attributes, and *cl* to the number of classes. They cover a wide range of cardinalities regarding the number of samples (from 106 up to 20,000), attributes (from 2 up to 309), and classes (from 2 up to 37). The selection of the datasets has been made considering that all their attributes are numerical. This requirement is imposed by the models used in experimentation to introduce borderline label noise into the data [23], which compute the distance of the samples to the decision boundaries and need numerical attributes for that purpose.

Table 1. Datasets used, along with their number of samples (*sa*), attributes (*at*), and classes (*cl*).

Dataset	*sa*	*at*	*cl*	Dataset	*sa*	*at*	*cl*
`balance`	625	4	3	`lsvt`	126	309	2
`banana`	5300	2	2	`miceprotein`	552	77	8
`banknote`	1372	4	2	`pageblocks`	5473	10	5
`biodeg`	1055	41	2	`parkinson`	195	22	2
`breast`	106	9	6	`pendigits`	10,992	16	10
`bupa`	345	6	2	`pima`	768	8	2
`climatemuq`	540	18	2	`seeds`	210	7	3
`column2C`	310	6	2	`segment`	2310	19	7
`column3C`	310	6	3	`sonar`	208	60	2
`energyheat`	768	8	37	`spectf`	267	44	2
`glass`	214	9	6	`transfusion`	748	4	2
`haberman`	306	3	2	`userkw`	403	5	4
`ionosphere`	351	34	2	`wdbc`	569	30	2
`iris`	150	4	3	`wine`	178	13	3
`landsat`	6435	36	6	`wisconsin`	683	9	2
`leaf`	340	14	30	`wpbc`	194	33	2
`letter`	20,000	16	26	`wqred`	1599	11	6
`libras`	360	90	15	`wqwhite`	4898	11	7

3.2. Noise Introduction Models

In the above datasets, nine levels of borderline label noise ρ% are injected in order to control the characteristics of the errors: from 0% (clean datasets) up to 40%, by increments of 5%. The following two noise models are used to introduce noise [23]:

1. *Neighborwise borderline label noise*. This calculates a noise measure $N(x_i)$ for each sample x_i based on the distances to its closest samples from the same class and from a different one. The noise measure $N(x_i)$ has the following expression:

$$N(x_i) = \frac{d(x_i, x_j = NN(x_i) \mid x_{j,0} = x_{i,0})}{d(x_i, x_k = NN(x_i) \mid x_{k,0} \neq x_{i,0})}$$

where $NN(x_i)$ is the nearest neighbor of x_i, $d(x_i, x_j)$ the Euclidean distance between the samples x_i and x_j, and $x_{i,0}$ the class label of the sample x_i. Finally, the values $N(x_i)$ are ordered in descending order, and the first ρ% of them are chosen to be mislabeled.

2. *Non-linearwise borderline label noise*. This computes a noise metric for each sample based on its distance to the decision limit induced by a *support vector machine* (SVM) [54]. In order to achieve this, it first uses SVM with a radial basis kernel to compute the decision boundary in the data D. Then, for each sample x_i in D, its distance to the decision border is calculated, considered as the unsigned decision values of SVM for that sample. For multi-class problems, the one-vs-one approach is used, and the distance between the sample and the nearest decision boundary is selected. Finally, the values of the noise metric are ordered in ascending order, and the first ρ% of them are chosen to be altered.

For a given dataset D in Table 1, noise is introduced as follows. First, a noise level ρ% is injected into a copy D' of D using one of the above noise models. Then, both datasets, D and D', are split into five equivalent parts, maintaining the same samples per fold. Finally, the training sets are selected from D' (using four folds), and the test sets are built from D (using the remaining fold). Both noise models, *neighborwise* and *non-linearwise borderline label noise*, are independently considered. For each one, nine noise levels are analyzed. This fact implies the usage of a total of 612 different noisy datasets in the experiment. The accuracy of each algorithm in these datasets is computed by averaging its test results over five runs of a five-fold cross-validation.

3.3. Classification Algorithms

The choice of the classification techniques employed in the experimentation (C4.5 [11], RIPPER [29], PART [30], and C5.0 [31]) is based on two main aspects related to the research carried out. First, they are algorithms traditionally considered when creating bagging-based ensembles [17,55]. Even though bagging can be applied regardless of the classification method, those approaches based on decision trees and ruleset creation are generally recommended when building ensembles [18]. Among their advantages [55,56], we can highlight that they are non-parametric (no assumptions about the data distribution are made) and interpretable, and, what is even more important, when multiple models are built from the data, they provide good solutions in relatively short times. These types of techniques based on decision trees and rulesets are commonly used in some of the most popular ensembles, such as XGBoost [57] or *random forest* [58]. Second, the algorithms considered include mechanisms against overfitting and are commonly used in works on noisy data in classification [32,33]. This paper delves into this field, studying the effect of borderline label noise on the performance of these robust learners, comparing their results with and without bagging. The classification techniques used in the experimentation are briefly described below:

1. C4.5 [11]. It is based on the ID3 [59] algorithm, including some improvements, such as the handling of missing values, the possibility of treating continuous attributes, and the usage of pruning to avoid overfitting. C4.5 follows a top-down approach to build the decision tree. In order to determine the current node in each of its stages, the attribute that best separates the remaining samples among classes is selected.
2. RIPPER [29]. Its main goal is to create a set of crisp rules from the training data. The rules are learned one by one, until they cover all the samples of each of the classes according to their frequency. For this, a stopping criterion based on the *minimum description length* [60] metric is used. Each rule is then pruned to avoid the overfitting of the previous stage. After learning the ruleset for a given class, an optimization stage is run, in which the rules are improved by adjusting their antecedents.
3. PART [30]. It relies on a divide-and-conquer strategy to create a set of *if-else* rules from the construction of partial decision trees, which are those whose branches are not completely explored. Thus, when the children of a given node are obtained, it can be chosen to be pruned. At each stage, PART creates a partial decision tree and converts

its best branch, the one that covers the most samples, into a rule in the ruleset. The algorithm stops once all samples in the dataset have been covered.

4. `C5.0` [31]. It has been considered in the experimentation as a more recent and advanced version of the classic `C4.5` algorithm. Among the improvements that `C5.0` offers with respect to its predecessor, we can highlight lower temporal and spatial complexities (which are especially useful when building ensembles), the creation of smaller decision trees that maintain their accuracy, the introduction of sample and misclassification weighting schemes, and the filtering of irrelevant attributes for the classification task.

The parameter setting for each method is the default one recommended by the authors:

- `C4.5`, `PART`, `C5.0`: pruning confidence $c = 0.25$; min. samples per leaf $s = 2$.
- `RIPPER`: folds $f = 3$; optimizations $r = 2$; min. weights $w = 2$.

Note that, in real-world applications, it is interesting to find the optimal parameters for each algorithm on each dataset in order to obtain the highest possible classification accuracy for the specific problem addressed. However, this aspect is not the object of this research, whose main goal is to analyze whether there is an improvement in the behavior of ensembles based on bagging with respect to not considering them when dealing with borderline label noise. Because of this, finding the optimal parameter setup for each method is not essential, and the same parameters are set for all of them, regardless of whether they use bagging or not. In this way, the variation in accuracy of each algorithm before and after using bagging will be due to the use of bagging itself and not to the optimization of the parameters for each method and dataset.

3.4. Methodology of Analysis

The main goal of the experimentation is to compare the performance of each classification method when dealing with borderline label noise before and after using bagging. In order to do this, the analysis of results will be focused on four main aspects:

1. *Classification accuracy*. Classification accuracy is computed for each algorithm on each dataset, noise model, and noise level. Note that, even though this paper presents averaged results, the conclusions drawn are supported by the proper statistical tests with respect to each of them. On the other hand, the complete results are accessible through the webpage (https://joseasaezm.github.io/bagbln/ (accessed on 1 May 2022)) with complementary material of this research.
2. *Robustness to noise*. The *equalized loss of accuracy* (`ELA`) [35] metric is used to evaluate the noise robustness by measuring the performance deterioration with noisy data from a perfect classification weighted by the performance with clean data:

$$ELA_{\rho\%} = \frac{1 - A_{\rho\%}}{A_{0\%}}$$

   where $A_{0\%}$ and $A_{\rho\%}$ are, respectively, the classification accuracies without noise and with a noise level $\rho\%$. In this case, the lower the `ELA` value, the greater the robustness of the classification algorithm. It is important to point out that the conclusions reached when studying accuracy and robustness do not necessarily have to coincide: an algorithm can have a good accuracy, but deteriorate to a greater degree (being less robust) when considering higher levels of noise in the data.
3. *Box-plots of robustness results*. Box-plots allow completion of the analysis of the robustness to noise of the classification algorithms by analyzing the distribution of the `ELA` results. Lower medians and interquartile ranges will be an indicator of good robustness in all the datasets used, showing similar performances of the methods before and after introducing noise into the data.
4. *Datasets with the best result*. Along with the above metrics (accuracy and `ELA`), the number of datasets in which each approach (bagging or baseline method) obtains the best result at each noise level is computed.

Wilcoxon's test [61] will be employed to properly analyze both accuracy and `ELA` results and detect differences between two sample means, as suggested in the literature [62]. For each noise model and level, the baseline algorithm and that using bagging will be compared, and the corresponding p-values will be obtained. The p-value for each comparison will allow rejection of the null hypothesis of equality of means, implying that a given algorithm outperforms the other. This research considers a significance level $\alpha = 0.05$.

4. Addressing the Borderline Label Noise Problem with Bagging Ensembles

This section analyzes both the accuracy and robustness of the classification methods, with and without bagging, dealing with borderline label noise. Section 4.1 focuses on the impact of borderline noise on classification accuracy, whereas Section 4.2 focuses on the robustness against noise of each approach.

4.1. Impact of Borderline Label Noise on Classification Accuracy

Table 2 presents the accuracy results (rows *ACC*) of each classification method with and without bagging with each noise model and noise level. Additionally, the amount of datasets with the best result for each classification technique (rows *Best*) and the p-values obtained using *Wilcoxon*'s test (rows p_{Wil}) are provided.

The following observations emerge from the analysis of these results:

- The test accuracy results are higher for bagging than for non-bagging in both noise models, neighborwise and non-linearwise, at all the noise levels.
- The improvements using bagging are approximately between 2–4% in all cases.
- They are slightly larger for `RIPPER` and `PART` than for `C4.5` and `C5.0`.
- The largest improvements for each method are generally found at medium-high noise levels, that is, from 15–20% onwards.
- The rows *Best* show a clear advantage in favor of bagging, which provides the best accuracy in the majority of the datasets.
- The low p-values obtained with *Wilcoxon*'s test support the superiority of the bagging schemes in all the comparisons.

The results in Table 2 show that bagging schemes provide higher accuracies for all the classification algorithms studied when the data suffer from borderline label noise. Furthermore, this ensemble-building approach even improves the accuracy of algorithms traditionally considered robust to noise when dealing with this type of complex data. The improvement percentages obtained through the application of bagging (2–4%) represent significant amounts in classification problems. Note that these percentages of improvement occur in the borderline area among classes, where samples tend to be more confusing. In certain types of real-world applications, such as medical ones, these percentages can have a large impact on the classification system. On many occasions, the classification decision usually involves the health of patients who are difficult to classify, whose descriptive attributes place them on the border between two of the classes of the problem.

On the other hand, those classification methods generally providing worse performance results in some noise levels without using bagging, such as `RIPPER`, benefit the most from its usage, obtaining larger percentages of improvement on average. It is worth noting the behavior of `PART` at the levels 25–30% with non-linearwise borderline noise, where, despite obtaining good results among the methods that do not use bagging, it reaches high percentages of improvement when considered within the ensemble. The best results of improvement for all the classification algorithms, which are usually obtained at medium-high noise levels, show that the impact of bagging is potentially greater in the most complex classification problems.

Given that bagging provides better results in the majority of the datasets and that the statistical comparisons confirm its good behavior, its usage can be recommended when the data suffer from borderline label noise. Its better performance against label noise affecting decision boundaries can be explained by the fact that the bootstrap resampling procedure may cause each model to be affected by only some of the borderline samples. In this way, the separability between the classes can be increased, reducing the chances that decision limits induced by the classifiers overfit the noisy data.

Table 2. Accuracy results of baseline and bagging classifiers with borderline label noise.

Neighborwise Borderline Label Noise										
	Method	**0%**	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**
ACC	C4.5	0.8120	0.8120	0.8044	0.7906	0.7725	0.7540	0.7296	0.7030	0.6716
	Bag-C4.5	**0.8399**	**0.8398**	**0.8321**	**0.8233**	**0.8049**	**0.7850**	**0.7595**	**0.7344**	**0.7004**
Best	C4.5	3	3	4	4	5	4	6	5	6
	Bag-C4.5	**33**	**33**	**32**	**32**	**31**	**32**	**30**	**31**	**30**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	RIPPER	0.7990	0.7926	0.7854	0.7839	0.7647	0.7523	0.7285	0.7015	0.6766
	Bag-RIPPER	**0.8301**	**0.8253**	**0.8239**	**0.8192**	**0.8072**	**0.7846**	**0.7621**	**0.7388**	**0.7093**
Best	RIPPER	6	5	4	4	2	6	6	6	8
	Bag-RIPPER	**30**	**32**	**32**	**33**	**34**	**30**	**30**	**30**	**28**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	PART	0.8136	0.8080	0.8009	0.7846	0.7677	0.7494	0.7207	0.6974	0.6644
	Bag-PART	**0.8428**	**0.8373**	**0.8338**	**0.8229**	**0.8106**	**0.7880**	**0.7641**	**0.7329**	**0.7040**
Best	PART	6	3	4	0	3	6	3	7	4
	Bag-PART	**30**	**34**	**32**	**36**	**33**	**30**	**33**	**29**	**32**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	C5.0	0.8135	0.8117	0.8075	0.7951	0.7790	0.7524	0.7315	0.7083	0.6815
	Bag-C5.0	**0.8371**	**0.8368**	**0.8320**	**0.8229**	**0.8068**	**0.7868**	**0.7589**	**0.7326**	**0.7025**
Best	C5.0	5	6	4	1	5	7	5	7	9
	Bag-C5.0	**31**	**30**	**32**	**35**	**31**	**29**	**31**	**29**	**27**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
Non-Linearwise Borderline Label Noise										
	Method	**0%**	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**
ACC	C4.5	0.8120	0.8058	0.7994	0.7828	0.7663	0.7449	0.7200	0.6963	0.6681
	Bag-C4.5	**0.8399**	**0.8344**	**0.8238**	**0.8102**	**0.7949**	**0.7779**	**0.7522**	**0.7291**	**0.7032**
Best	C4.5	3	5	4	5	8	2	6	6	6
	Bag-C4.5	**33**	**31**	**32**	**31**	**28**	**34**	**30**	**30**	**30**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	RIPPER	0.7990	0.7943	0.7845	0.7731	0.7590	0.7435	0.7236	0.7010	0.6851
	Bag-RIPPER	**0.8301**	**0.8253**	**0.8167**	**0.8071**	**0.7940**	**0.7750**	**0.7604**	**0.7367**	**0.7124**
Best	RIPPER	6	6	3	4	5	6	4	5	7
	Bag-RIPPER	**30**	**30**	**34**	**34**	**31**	**30**	**33**	**32**	**30**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	PART	0.8136	0.8071	0.7976	0.7826	0.7696	0.7471	0.7239	0.6907	0.6632
	Bag-PART	**0.8428**	**0.8385**	**0.8279**	**0.8140**	**0.7993**	**0.7835**	**0.7621**	**0.7305**	**0.7034**
Best	PART	6	4	3	5	6	3	3	4	8
	Bag-PART	**30**	**32**	**33**	**33**	**30**	**33**	**33**	**32**	**28**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ACC	C5.0	0.8135	0.8122	0.7999	0.7830	0.7672	0.7425	0.7205	0.6988	0.6680
	Bag-C5.0	**0.8371**	**0.8321**	**0.8247**	**0.8132**	**0.7947**	**0.7736**	**0.7530**	**0.7270**	**0.7026**
Best	C5.0	5	8	5	4	6	7	7	8	7
	Bag-C5.0	**31**	**28**	**31**	**32**	**30**	**29**	**29**	**28**	**29**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**

4.2. Analysis of Classification Robustness to Borderline Noise

Table 3 shows the robustness results using the ELA metric for each classification algorithm, with and without bagging, in each noise model and noise level.

Table 3. Robustness results of baseline and bagging classifiers with borderline label noise.

					Neighborwise Borderline Label Noise					
	Method	**0%**	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**
ELA	C4.5	0.2613	0.2591	0.2681	0.2833	0.3052	0.3258	0.3537	0.3844	0.4215
	Bag-C4.5	**0.2114**	**0.2104**	**0.2195**	**0.2288**	**0.2493**	**0.2711**	**0.3003**	**0.3284**	**0.3673**
Best	C4.5	3	3	3	4	5	5	5	4	5
	Bag-C4.5	**33**	**33**	**33**	**32**	**31**	**31**	**31**	**32**	**31**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	RIPPER	0.3015	0.3060	0.3154	0.3166	0.3396	0.3528	0.3819	0.4151	0.4413
	Bag-RIPPER	**0.2350**	**0.2405**	**0.2407**	**0.2450**	**0.2586**	**0.2852**	**0.3105**	**0.3366**	**0.3700**
Best	RIPPER	6	4	4	4	2	6	6	4	7
	Bag-RIPPER	**30**	**32**	**32**	**32**	**34**	**30**	**30**	**32**	**29**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	PART	0.2607	0.2659	0.2736	0.2920	0.3115	0.3316	0.3654	0.3926	0.4305
	Bag-PART	**0.2075**	**0.2139**	**0.2168**	**0.2291**	**0.2409**	**0.2674**	**0.2944**	**0.3303**	**0.3618**
Best	PART	6	4	4	2	4	6	3	7	5
	Bag-PART	**30**	**32**	**32**	**34**	**32**	**30**	**33**	**29**	**31**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	C5.0	0.2571	0.2574	0.2623	0.2759	0.2952	0.3259	0.3496	0.3758	0.4066
	Bag-C5.0	**0.2177**	**0.2162**	**0.2216**	**0.2311**	**0.2488**	**0.2715**	**0.3035**	**0.3324**	**0.3668**
Best	C5.0	5	7	4	2	5	4	4	8	8
	Bag-C5.0	**31**	**29**	**32**	**34**	**31**	**32**	**32**	**28**	**28**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
					Non-Linearwise Borderline Label noise					
	Method	**0%**	**5%**	**10%**	**15%**	**20%**	**25%**	**30%**	**35%**	**40%**
ELA	C4.5	0.2613	0.2676	0.2736	0.2934	0.3131	0.3384	0.3670	0.3968	0.4285
	Bag-C4.5	**0.2114**	**0.2182**	**0.2301**	**0.2454**	**0.2631**	**0.2816**	**0.3112**	**0.3374**	**0.3658**
Best	C4.5	3	4	4	3	6	2	5	5	6
	Bag-C4.5	**33**	**32**	**32**	**33**	**30**	**34**	**31**	**31**	**30**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	RIPPER	0.3015	0.3062	0.3167	0.3329	0.3484	0.3675	0.3899	0.4177	0.4334
	Bag-RIPPER	**0.2350**	**0.2402**	**0.2502**	**0.2624**	**0.2763**	**0.2987**	**0.3141**	**0.3418**	**0.3701**
Best	RIPPER	6	5	3	1	4	5	1	4	5
	Bag-RIPPER	**30**	**31**	**33**	**35**	**32**	**31**	**35**	**32**	**31**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	PART	0.2607	0.2671	0.2777	0.2950	0.3093	0.3372	0.3636	0.4025	0.4339
	Bag-PART	**0.2075**	**0.2122**	**0.2246**	**0.2406**	**0.2568**	**0.2739**	**0.2979**	**0.3349**	**0.3655**
Best	PART	6	3	4	6	6	4	3	3	6
	Bag-PART	**30**	**33**	**32**	**30**	**30**	**32**	**33**	**33**	**30**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**
ELA	C5.0	0.2571	0.2569	0.2716	0.2910	0.3091	0.3390	0.3650	0.3906	0.4256
	Bag-C5.0	**0.2177**	**0.2226**	**0.2311**	**0.2437**	**0.2649**	**0.2894**	**0.3124**	**0.3415**	**0.3691**
Best	C5.0	5	7	3	2	4	5	7	7	7
	Bag-C5.0	**31**	**29**	**33**	**34**	**32**	**31**	**29**	**29**	**29**
p_{Wil}	-	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**<0.001**

From the analysis of Table 3, the following results arise:

- The ELA values are better for the algorithms using bagging than for baseline classifiers at all the noise levels for the two borderline noise models studied.
- The most favorable advantages for each method using bagging generally occur at medium-high noise levels (from 20–25% onwards).
- These differences are usually more noticeable in the RIPPER algorithm, followed by PART, C4.5, and finally C5.0.

- The number of datasets in which each algorithm shows a greater robustness provides clear results in favor of bagging, being the best in most datasets.
- The *p*-values of *Wilcoxon*'s test confirm the robustness of the bagging schemes compared to their non-application.

Figure 2 shows the box-plots for the ELA values of each classification method on datasets with borderline label noise. Figure 2a–d show the plots for neighborwise borderline noise, whereas Figure 2e–h show the distributions for non-linearwise borderline noise. These plots show that the ELA medians and interquartile ranges of the methods that use bagging are generally lower compared to those that do not. It is also observed that the methods not using bagging often present more outliers in their robustness results at lower noise levels than if they consider it.

The above results show the efficacy of bagging when dealing with label noise at decision boundaries between classes. The bagging-based schemes obtain the highest accuracy and robustness with respect to their non-bagging counterparts. The partial consideration of mislabeled samples when creating the subsamples from the original training set in bagging can make it so that only some classifiers and some of the parts of the boundaries among classes are impaired. Because of this, the global system may be less affected than it is when creating a single model from the whole dataset containing errors.

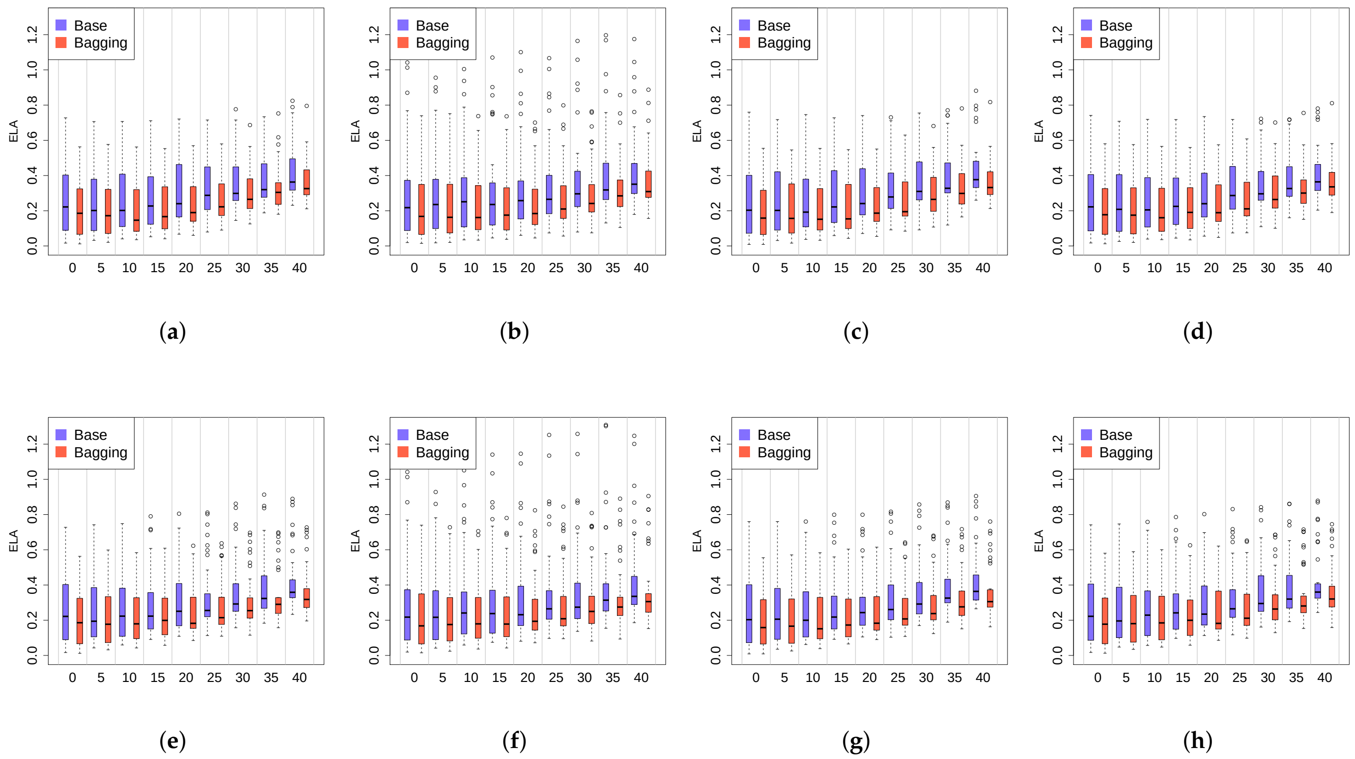

Figure 2. Distributions of robustness results (ELA) for each classification algorithm (C4.5, RIPPER, PART, and C5.0) at each noise level and borderline noise model (*neighborwise* and *non-linearwise*). (**a**) C4.5 (*neighborwise*); (**b**) RIPPER (*neighborwise*); (**c**) PART (*neighborwise*); (**d**) C5.0 (*neighborwise*); (**e**) C4.5 (*non-linearwise*); (**f**) RIPPER (*non-linearwise*); (**g**) PART (*non-linearwise*); (**h**) C5.0 (*non-linearwise*).

5. Conclusions

This research has focused on a comparison of the behavior of bagging-based ensembles against their individual components when the data is affected by borderline label noise. A total of 612 noisy datasets, considering various models and noise levels, have been used to analyze this comparison. On these datasets, the C4.5, RIPPER, PART, and C5.0 robust learners have been employed to create classifiers with and without the usage of bagging.

The results derived from the experimentation carried out have shown that bagging provides better accuracy and robustness results in the models and noise levels studied. The lowest improvements of average accuracy using bagging are around 2%, whereas the largest are around 4% and are usually obtained from the noise levels 15–20%. In these noise levels, a larger amount of noisy samples are available, producing greater advantages in favor of bagging. The quantity of datasets where bagging provides the highest accuracy is always above 27 (out of 36) at each noise level and noise model, regardless of the classification algorithm (the average being 31.13). However, the robustness results show a slightly greater superiority of the bagging-based methods, which are able to increase the number of datasets with the best result above 28 in all cases (with an average amount of 31.44). *Wilcoxon*'s test supports the good behavior of bagging, providing p-values below 0.001 in all the comparisons.

The main hypothesis to explain the better results of bagging-based methods with borderline noise is that the bootstrap resampling procedure may cause each model to be affected by only some of the borderline samples. Thus, the separability between classes can be increased, and the classifiers do not overfit the noisy data as much as in the case where bagging is not considered. Although it should be noted that the use of ensembles increases the computational cost, since several models are created from the training set, the advantages in accuracy and robustness offered by bagging in this scenario imply that it can be recommended as a simple and effective way to deal with borderline label noise.

Among the limitations and possibilities for improvement of this work, it may be interesting to analyze the imbalance ratio and other well-known characteristics of classification data, such as the dispersion of samples and the overlapping among classes [36], before and after introducing borderline label noise, determining those cases in which bagging provides better results. Another aspect to address is the analysis of the samples that are part of each subsample created by bagging, computing the number of clean and noisy samples at class boundaries in each one in order to deepen the understanding of the circumstances that make bagging work better with this type of complex data.

In future works, the synergy between bagging and preprocessing methods for the treatment of noisy data will be studied in order to test their joint operation when dealing with borderline label errors. Furthermore, the behavior of bagging when the data is affected by other types of noise in the borderline region, such as attribute noise, must be also studied.

Author Contributions: All authors have contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this research are taken from the *UCI machine learning* (https://archive.ics.uci.edu/ (accessed on 1 May 2022)) and *KEEL-dataset* (http://www.keel.es (accessed on 1 May 2022)) repositories.

Acknowledgments: J.L. Romero has been partially supported by grants MCIU/AEI/ERDF, UE PGC2018-098860-B-I00, grant A-FQM-345-UGR18 cofinanced by ERDF Operational Programme 2014–2020 and the Economy and Knowledge Council of the Regional Government of Andalusia, Spain, and grant CEX2020-001105-M MCIN/AEI/10.13039/501100011033.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Chen, W.; Yang, K.; Shao, Y.; Chen, Y.; Zhang, J.; Yao, J. A trace lasso regularized robust nonparallel proximal Support Vector Machine for noisy classification. *IEEE Access* **2019**, *7*, 47171–47184. [CrossRef]
2. Nematzadeh, Z.; Ibrahim, R.; Selamat, A. Improving class noise detection and classification performance: A new two-filter CNDC model. *Appl. Soft Comput.* **2020**, *94*, 106428. [CrossRef]

3. Martín, J.; Sáez, J.A.; Corchado, E. On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters. *IEEE Access* **2021**, *9*, 145800–145816. [CrossRef]
4. Pawara, P.; Okafor, E.; Groefsema, M.; He, S.; Schomaker, L.; Wiering, M. One-vs-One classification for deep neural networks. *Pattern Recognit.* **2020**, *108*, 107528. [CrossRef]
5. Tian, Y.; Sun, M.; Deng, Z.; Luo, J.; Li, Y. A new fuzzy set and nonkernel SVM approach for mislabeled binary classification with applications. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1536–1545. [CrossRef]
6. Yu, Z.; Wang, D.; Zhao, Z.; Chen, C.L.P.; You, J.; Wong, H.; Zhang, J. Hybrid incremental ensemble learning for noisy real-world data classification. *IEEE Trans. Cybern.* **2019**, *49*, 403–416. [CrossRef]
7. Liu, T.; Tao, D. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 447–461. [CrossRef]
8. Sáez, J.A.; Corchado, E. ANCES: A novel method to repair attribute noise in classification problems. *Pattern Recognit.* **2022**, *121*, 108198. [CrossRef]
9. Huang, L.; Shao, Y.; Zhang, J.; Zhao, Y.; Teng, J. Robust rescaled hinge loss twin support vector machine for imbalanced noisy classification. *IEEE Access* **2019**, *7*, 65390–65404. [CrossRef]
10. Li, J.; Zhu, Q.; Wu, Q.; Zhang, Z.; Gong, Y.; He, Z.; Zhu, F. SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution. *Knowl.-Based Syst.* **2021**, *223*, 107056. [CrossRef]
11. Quinlan, J. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Francisco, CA, USA, 2014.
12. Sáez, J.A.; Luengo, J.; Herrera, F. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. *Pattern Recognit.* **2013**, *46*, 355–364. [CrossRef]
13. Chaudhury, S.; Yamasaki, T. Robustness of adaptive neural network optimization under training noise. *IEEE Access* **2021**, *9*, 37039–37053. [CrossRef]
14. Cui, S.; Wang, Y.; Yin, Y.; Cheng, T.; Wang, D.; Zhai, M. A cluster-based intelligence ensemble learning method for classification problems. *Inf. Sci.* **2021**, *560*, 386–409. [CrossRef]
15. Xia, Y.; Chen, K.; Yang, Y. Multi-label classification with weighted classifier selection and stacked ensemble. *Inf. Sci.* **2021**, *557*, 421–442. [CrossRef]
16. Lughofer, E.; Pratama, M.; Škrjanc, I. Online bagging of evolving fuzzy systems. *Inf. Sci.* **2021**, *570*, 16–33. [CrossRef]
17. Sun, J.; Lang, J.; Fujita, H.; Li, H. Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates. *Inf. Sci.* **2018**, *425*, 76–91. [CrossRef]
18. Jafarzadeh, H.; Mahdianpari, M.; Gill, E.; Mohammadimanesh, F.; Homayouni, S. Bagging and boosting ensemble classifiers for classification of multispectral, hyperspectral and polsar data: A comparative evaluation. *Remote Sens.* **2021**, *13*, 4405. [CrossRef]
19. Abellán, J.; Castellano, J.; Mantas, C. A new robust classifier on noise domains: Bagging of credal C4.5 trees. *Complexity* **2017**, *2017*, 9023970. [CrossRef]
20. Khoshgoftaar, T.; Van Hulse, J.; Napolitano, A. Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2011**, *41*, 552–568. [CrossRef]
21. Wei, Y.; Gong, C.; Chen, S.; Liu, T.; Yang, J.; Tao, D. Harnessing side information for classification under label noise. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3178–3192. [CrossRef]
22. Bootkrajang, J. A generalised label noise model for classification. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Bruges, Belgium, 22–23 April 2015; pp. 349–354.
23. Garcia, L.P.F.; Lehmann, J.; de Carvalho, A.C.P.L.F.; Lorena, A.C. New label noise injection methods for the evaluation of noise filters. *Knowl.-Based Syst.* **2019**, *163*, 693–704. [CrossRef]
24. Bootkrajang, J.; Chaijaruwanich, J. Towards instance-dependent label noise-tolerant classification: A probabilistic approach. *Pattern Anal. Appl.* **2020**, *23*, 95–111. [CrossRef]
25. Du, J.; Cai, Z. Modelling class noise with symmetric and asymmetric distributions. In Proceedings of the 29th Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2589–2595.
26. Sáez, J.A.; Krawczyk, B.; Woźniak, M. On the influence of class noise in medical data classification: Treatment using noise filtering methods. *Appl. Artif. Intell.* **2016**, *30*, 590–609. [CrossRef]
27. Sluban, B.; Gamberger, D.; Lavrac, N. Ensemble-based noise detection: Noise ranking and visual performance evaluation. *Data Min. Knowl. Discov.* **2014**, *28*, 265–303. [CrossRef]
28. Garcia, L.P.F.; Lorena, A.C.; Matwin, S.; de Leon Ferreira de Carvalho, A.C.P. Ensembles of label noise filters: A ranking approach. *Data Min. Knowl. Discov.* **2016**, *30*, 1192–1216. [CrossRef]
29. Cohen, W. Fast effective rule induction. In Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 115–123.
30. Frank, E.; Witten, I. Generating accurate rule sets without global optimization. In Proceedings of the 15th International Conference on Machine Learning, Madison, WI, USA, 24–27 July 1998; pp. 144–151.
31. Rajeswari, S.; Suthendran, K. C5.0: Advanced Decision Tree (ADT) classification model for agricultural data analysis on cloud. *Comput. Electron. Agric.* **2019**, *156*, 530–539. [CrossRef]
32. 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]

33. Frenay, B.; Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 845–869. [CrossRef]
34. Singh, P.; Sarkar, R.; Nasipuri, M. Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets. *Int. J. Comput. Sci. Math.* **2016**, *7*, 410–442. [CrossRef]
35. Sáez, J.A.; Luengo, J.; Herrera, F. Evaluating the classifier behavior with noisy data considering performance and robustness: The Equalized Loss of Accuracy measure. *Neurocomputing* **2016**, *176*, 26–35. [CrossRef]
36. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 289–300.
37. Gupta, S.; Gupta, A. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Comput. Sci.* **2019**, *161*, 466–474. [CrossRef]
38. Zeng, S.; Duan, X.; Li, H.; Xiao, Z.; Wang, Z.; Feng, D. Regularized fuzzy discriminant analysis for hyperspectral image classification with noisy labels. *IEEE Access* **2019**, *7*, 108125–108136. [CrossRef]
39. Bootkrajang, J. A generalised label noise model for classification in the presence of annotation errors. *Neurocomputing* **2016**, *192*, 61–71. [CrossRef]
40. Yuan, W.; Guan, D.; Ma, T.; Khattak, A. Classification with class noises through probabilistic sampling. *Inf. Fusion* **2018**, *41*, 57–67. [CrossRef]
41. Adeli, E.; Thung, K.; An, L.; Wu, G.; Shi, F.; Wang, T.; Shen, D. Semi-supervised discriminative classification robust to sample-outliers and feature-noises. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 515–522. [CrossRef]
42. Koziarski, M.; Krawczyk, B.; Wozniak, M. Radial-Based oversampling for noisy imbalanced data classification. *Neurocomputing* **2019**, *343*, 19–33. [CrossRef]
43. Zhao, Z.; Chu, L.; Tao, D.; Pei, J. Classification with label noise: A Markov chain sampling framework. *Data Min. Knowl. Discov.* **2019**, *33*, 1468–1504. [CrossRef]
44. Shanthini, A.; Vinodhini, G.; Chandrasekaran, R.M.; Supraja, P. A taxonomy on impact of label noise and feature noise using machine learning techniques. *Soft Comput.* **2019**, *23*, 8597–8607. [CrossRef]
45. 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 International Symposium on Computer-Based Medical Systems, Salt Lake City, UT, USA, 22–23 June 2006; pp. 708–713.
46. Khoshgoftaar, T.M.; Hulse, J.V. Empirical case studies in attribute noise detection. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 379–388. [CrossRef]
47. Sculley, D.; Cormack, G.V. Filtering email spam in the presence of noisy user feedback. In Proceedings of the 5th Conference on Email and Anti-Spam, Mountain View, CA, USA, 21–22 August 2008; pp. 1–10.
48. Bi, J.; Zhang, T. Support vector classification with input data uncertainty. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2004; Volume 17, pp. 161–168.
49. Liang, D.; Yi, B. Two-stage three-way enhanced technique for ensemble learning in inclusive policy text classification. *Inf. Sci.* **2021**, *547*, 271–288. [CrossRef]
50. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258. [CrossRef]
51. Moyano, J.; Gibaja, E.; Cios, K.; Ventura, S. Review of ensembles of multi-label classifiers: Models, experimental study and prospects. *Inf. Fusion* **2018**, *44*, 33–45. [CrossRef]
52. Singhal, Y.; Jain, A.; Batra, S.; Varshney, Y.; Rathi, M. Review of bagging and boosting classification performance on unbalanced binary classification. In Proceedings of the 8th International Advance Computing Conference, Greater Noida, India, 14–15 December 2018; pp. 338–343.
53. Pakrashi, A.; Namee, B.M. KalmanTune: A Kalman filter based tuning method to make boosted ensembles robust to class-label noise. *IEEE Access* **2020**, *8*, 145887–145897. [CrossRef]
54. Baldomero-Naranjo, M.; Martínez-Merino, L.; Rodríguez-Chía, A. A robust SVM-based approach with feature selection and outliers detection for classification problems. *Expert Syst. Appl.* **2021**, *178*, 115017. [CrossRef]
55. Dietterich, T. Experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Mach. Learn.* **2000**, *40*, 139–157. [CrossRef]
56. Zhang, D.; Zhou, X.; Leung, S.; Zheng, J. Vertical bagging decision trees model for credit scoring. *Expert Syst. Appl.* **2010**, *37*, 7838–7843. [CrossRef]
57. Cherif, I.L.; Kortebi, A. On using extreme gradient boosting (XGBoost) machine learning algorithm for home network traffic classification. In Proceedings of the 2019 Wireless Days, Manchester, UK, 24–26 April 2019; p. 8734193.
58. Hansch, R. *Handbook of Random Forests: Theory and Applications for Remote Sensing*; World Scientific Publishing: Singapore, 2018.
59. Quinlan, J. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
60. Grunwald, P.; Roos, T. Minimum description length revisited. *Int. J. Math. Ind.* **2019**, *11*, 1930001. [CrossRef]
61. Baringhaus, L.; Gaigall, D. Efficiency comparison of the Wilcoxon tests in paired and independent survey samples. *Metrika* **2018**, *81*, 891–930. [CrossRef]
62. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]