Hindawi
Discrete Dynamics in Nature and Society
Volume 2024, Article ID 6671395, 10 pages
https://doi.org/10.1155/2024/6671395

*Research Article*

# Classification Performance Analysis of Decision Tree-Based Algorithms with Noisy Class Variable

**Abdulmajeed Atiah Alharbi**

*Department of Mathematics, College of Science, Taibah University, Al-Madinah Al-Munawara, Saudi Arabia*

Correspondence should be addressed to Abdulmajeed Atiah Alharbi; aahharbi@taibahu.edu.sa

Received 14 June 2023; Revised 13 January 2024; Accepted 24 January 2024; Published 1 February 2024

Academic Editor: Mijanur Rahaman Seikh



Class noise is a common issue that affects the performance of classification techniques on real-world data sets. Class noise appears when a class variable in data sets has incorrect class labels. In the case of noisy data, the robustness of classification techniques against noise could be more important than the performance results on noise-free data sets. The decision tree method is one of the most popular techniques for classification tasks. The C4.5, CART, and random forest (RF) algorithms are considered to be three of the most used algorithms in decision trees. The aim of this paper is to reach conclusions on which decision tree algorithm is better to use for building decision trees in terms of its performance and robustness against class noise. In order to achieve this aim, we study and compare the performance of the models when applied to class variables with noise. The results obtained indicate that the RF algorithm is more robust to data sets with noisy class variable than other algorithms.

## 1. Introduction

In the area of data mining and machine learning, classification is one of the most commonly used techniques. The aim of classification is to predict classes of instances whose attribute values are known, but their classes are unknown. The variable to be predicted is known as the class variable, and the other variables are the attribute variables or features. Many classification methods have been introduced in the literature such as decision trees, naive Bayes, logistic regression, and discriminant analysis. Decision trees (also called classification trees) are one of the most preferable approaches to use in classification because of their interpretational simplicity. Among the different algorithms to build decision trees, the C4.5, CART, and random forest (RF) algorithms are the most studied and commonly used for tree construction. In terms of interpretability, single trees such as the C4.5 and CART algorithms are easy to interpret, whereas ensemble methods such as the RF algorithm are not easily interpretable.

Real-world data sets, which are used as input for classification algorithms, are never perfect and could be affected by various factors. One of these factors is the presence of noise. Data noise is an unavoidable problem, which may hinder the interpretations, decisions, and performance of classification algorithms built from such noisy data sets. One of the data noise types is class noise, which occurs when data sets have incorrect class labels. Several studies have been published that test the performance of different classifiers, including decision trees when applied to class variables with noisy instances. This research focuses only on class noise; however, handling attribute noise is more difficult as the impact of attribute noise on the overall performance is unclear. This could be because of the dependence among attribute variables and with class variables as well [1]. The performance of different classification algorithms depends crucially on the quality of data sets; hence, the performance of classification algorithms may be negatively affected when developed using data sets with noisy class variables. However, some algorithms may be more robust to class noise than others. As a consequence, studying the performance of classification algorithms in the presence of noisy data is a significant issue in data mining and machine learning. Many studies discussed class and attribute noise and their

impact on the performance of classification algorithms [2–4].

In this paper, we investigate the performance of three machine learning algorithms: C4.5, CART, and RF, on data sets with varying levels of class noise. In order to evaluate classifiers with noisy data sets, we require a technique to introduce noise into data sets. One of the most commonly used and successful methods in the literature is to add random noise to the class variable. We use this method in our experimental analysis by adding random noise with different percentages to the class variable. The performance of the C4.5, CART, and RF algorithms with noisy class variable has been evaluated using two common evaluation measures, which are the overall classification accuracy and $F$-measure rates.

The rest of this paper is structured in the following way. Section 2 provides a brief background on decision trees and the most common algorithms for building trees. Section 3 presents an introduction to data noise, discusses data noise impact on classification algorithms, and describes different techniques for introducing noise into data sets. In Section 4, we discuss the findings of the experimental analysis conducted to test and compare the performance of the C4.5, CART, and RF algorithms on data sets with varying levels of class noise. Finally, Section 5 concludes the final remarks and suggests potential topics for future research.

## 2. Decision Trees

Classification is a data mining technique that assigns a new instance to predefined classes based on attribute variables. The decision tree method is one of the most commonly used methods of classification. Decision trees are attractive due to their interpretational simplicity, enabling the prediction of possible class by simple partitions. A decision tree is a model that can be used in classification and regression tasks. A classification task can be considered when the class variable is nominal, whereas in the situation that the class variable is numerical, regression task can be used. In this paper, we consider a decision tree within the classification tasks.

The decision tree algorithm is used to classify new instances into a set of predefined classes based on their attribute values. A decision tree consists of three types of nodes: a root node, which is the highest node in the tree and has no incoming edges; an internal node, which only has one incoming edge but two or more outgoing edges; and a leaf node, which has no outgoing edges. In a decision tree, each nonleaf node expresses an attribute variable, each branch expresses the outcome of an attribute variable, and each leaf specifies the predicted label of the class variable based on the information available in the training set. Once a decision tree is built, classifying a new instance of the test data set is a straightforward task. Instances are classified by following the path in the tree starting from the root until a leaf node, based on the attribute values of the variables along the path.

There are a number of approaches that have been published in the literature to construct a decision tree. Three of the most commonly used are the C4.5 [5], CART [6], and RF [7] algorithms. The C4.5, CART, and RF algorithms are summarized in Sections 2.1–2.3, respectively.

*2.1. C4.5 Algorithm.* The C4.5 algorithm was first introduced by Quinlan in 1993 [5] as a revised version of the ID3 algorithm [8]. The ID3 algorithm uses information gain as the split criterion, which employs entropy as an impurity measure. Entropy [9] of a training set $\mathscr{D}$ is given by the following equation:

$$E(\mathscr{D}) = -\sum_{i=1}^{m} p_i \log_2(p_i), \tag{1}$$

where $p_i$ represents the proportion of $\mathscr{D}$ that belongs to class $i$, $m$ represents the number of classes, and the logarithmic function with base 2 is used because information in computers is encoded in bits [10]. Entropy generally refers to the degree of uncertainty or impurity in a set of examples. The information gain of $A$ relative to $\mathscr{D}$ is given by

$$\text{Gain}(\mathscr{D}, A) = E(\mathscr{D}) - \sum_{j=1}^{n} \frac{|\mathscr{D}_j|}{|\mathscr{D}|} E(\mathscr{D}_j), \tag{2}$$

where the training set $\mathscr{D}$ is partitioned into $n$ partitions corresponding to the value of the attribute variable $A$, $\mathscr{D}_j$ represents the subset of $\mathscr{D}$ for which attribute variable $A$ has value $j$, and $|\mathscr{D}|$ is the cardinality of $\mathscr{D}$. The information gain handles only nominal attribute variables. The C4.5 algorithm is capable of handling both nominal and numerical attribute variables, which is not the case with the ID3 algorithm. The information gain tends to favor attribute variables that have a larger number of states, which may result in a biased analysis [8]. To address this issue, Quinlan [5] introduced the gain ratio split criterion. This criterion normalizes the information gain as follows:

$$\text{GR}(\mathscr{D}, A) = \frac{\text{Gain}(\mathscr{D}, A)}{\text{SI}(\mathscr{D}, A)}, \tag{3}$$

where $\text{Gain}(\mathscr{D}, A)$ is given by equation (2), and split information $\text{SI}(\mathscr{D}, A)$ is given by

$$\text{SI}(\mathscr{D}, A) = -\sum_{j=1}^{n} \frac{|\mathscr{D}_j|}{|\mathscr{D}|} \log_2 \frac{|\mathscr{D}_j|}{|\mathscr{D}|}. \tag{4}$$

$\text{SI}(\mathscr{D}, A)$ denotes the information gained by splitting the set $\mathscr{D}$ into $n$ subsets based on the values of the attribute $A$. The attribute with the maximum gain ratio split criterion (formula (3)) is selected by the C4.5 algorithm as the splitting attribute variable at each node when constructing the tree.

*2.2. CART Algorithm.* The classification and regression trees (CART) algorithm was introduced by Breiman et al. in 1984 [6]. The decision tree construction by the CART algorithm is based on binary splitting of the attribute variables. The CART algorithm employs the Gini Index splitting measure in choosing the best splitting attribute variable. The Gini Index measures how impure an

attribute variable is relative to its classes. It is given by the following equation:

$$\text{Gini}(\mathscr{D}) = 1 - \sum_{i=1}^{m} (p_i)^2, \tag{5}$$

where $p_i$ represents the relative frequency of class $i$ in the set $\mathscr{D}$, for $i = 1, \ldots, m$. The Gini Index reaches its minimum value when all the observations in the sample are of the same class and reaches its maximum value when all classes have an equal probability. After dividing the set $\mathscr{D}$ into two subsets $\mathscr{D}_1$ and $\mathscr{D}_2$ with sizes $n_1$ and $n_2$, respectively, the Gini Index of the split data is given by

$$\text{Gini}_{\text{split}}(\mathscr{D}) = \frac{n_1}{n}\text{Gini}(\mathscr{D}_1) + \frac{n_2}{n}\text{Gini}(\mathscr{D}_2). \tag{6}$$

In this way, the best Gini value split is chosen. The CART algorithm can handle both nominal and numerical attribute variables. Since the C4.5 and CART algorithms were published, they have been considered as standard models in classification.

*2.3. Random Forest.* The random forest (RF) algorithm was first introduced by Breiman in 2001 [7]. RF algorithm is a kind of ensemble approach that consists of multiple decision trees. In classification tasks, the RF algorithm makes a prediction by aggregating the majority vote of multiple independent decision trees. Each tree in RF contributes its vote for the classification, whereas those votes are used to make the final prediction of the RF classification algorithm.

To construct the RF algorithm, we choose a bootstrap sample of the training data (sample with replacement) and construct a decision tree on this sample using the following conditions: at each node, we randomly choose a small number of variables from the total number of attribute variables, and then, we pick the best splitting variable among these selections. Thereafter, another subset of variables is chosen for the subsequent node. After that, we repeat this process with another bootstrap sample from the training data to build many trees. Finally, a new instance is predicted by combining the prediction of these trees (i.e., majority vote) [11]. The RF algorithm reduces the correlation among the trees because the RF algorithm randomly chooses variables at each node, which helps to achieve an efficient prediction by this classifier [12]. The RF algorithm has many decision trees, which makes it a robust and efficient algorithm [13]. The C4.5, CART, and RF classifiers have been widely applied as data analysis tools in many fields, such as banking, medicine, and astronomy.

## 3. Data Noise

The presence of noise is a common issue in real-world data sets that may suffer from corruptions, thereby impacting the performance of classification algorithms constructed using such noisy data. Therefore, decisions based on models constructed from these noisy data sets may be negatively affected by data noise. Data noise refers to situations that appear when data sets have incorrect values in attribute variables or class labels. Noise in data sets can occur for a variety of reasons, including incorrect measurement of the inputs, experts' incorrect descriptions of the input values, the use of faulty measuring instruments, or data loss during data transmission and sorting [14]. In this paper, we consider only the class noise that occurs when an instance class is incorrectly labeled. The performance of models based on noisy data sets is a crucial issue for machine learning techniques. Classification algorithms based on noisy data sets are expected to be less accurate than those based on noise-free data sets [15].

This paper focuses on the effect of applying classification algorithms to noisy class variables. To test how well classification algorithms can handle noisy data, we compare their performance on a noise-free data set to their performance on the same data set with added noise. By doing this, we can assess the robustness of the algorithms. If the classification accuracy results for the noisy data are close to those for the clean data, the algorithm is considered robust. The robustness of classification algorithms depends on their ability to generate decision trees that are not affected by corrupt data sets. This method of assessing the robustness of classification algorithms in the presence of noise has also been utilized by Sáez et al. [16].

*3.1. Impact of Data Noise on Classification Algorithms.* This section reviews some studies which have explored the impact of class or attribute noise on classification algorithms are discussed. We provide a brief description of some of these studies and the concluded findings. Attribute noise has received less attention than class noise in the literature. Handling attribute noise is more complicated than class noise for several reasons. For example, the relationship between attribute noise and the classification accuracy is not clear, as the impact of noisy attribute variables depends on the dependence between attribute variables and the class variable [1]. Attribute variables could also have some correlations between each other; hence, this correlation may vary from one attribute to another, where the influences of adding noise to attribute variables can impact classification algorithm performance differently [17].

Numerous studies have been conducted to evaluate the efficacy of classification algorithms in the presence of a noisy class variable [18–24]. Recent studies indicate that class noise has a more significant impact on the performance of classification algorithms than attribute noise [1, 21]. The study by Zhu and Wu [24] analyzed the impact of class noise on cost-sensitive classification models. Cost-sensitive classification aims to minimize the cost of misclassification instead of solely maximizing classification accuracy. The results of this study indicate that class noise significantly impacts the performance of cost-sensitive classification models, particularly when incorrectly predicting classes is extremely expensive.

Several experimental studies have been conducted by Mantas and Abellán [22] to compare the performance of imprecise probability-based credal-C4.5 classification

algorithm with classical classification algorithms such as the ID3 and C4.5 algorithms. Their results found that the credal-C4.5 algorithm outperforms other algorithms with noisy class variable, while without noisy class variable, similar performance has been given by all classification algorithms.

Zhu and Wu [1] present a systematic evaluation of the impact of noise on machine learning. They investigated the impact of class and attribute noise on the accuracy rate for different classification models, including the C4.5 algorithm. Mantas and Abellán [25] also tested the performance of decision tree algorithms with various levels of noise. Various studies have examined how attribute or class noise affects classification accuracy across different classification algorithms [15, 16, 26]. However, more attention has been given to noise in the class variable in the literature.

An application of bagging credal decision trees has been presented by Abellán and Masegosa [19, 20]. A bagging classifier generates multiple versions of classification algorithms and then uses these algorithms to produce an aggregated algorithm [27]. The results of this study suggested that bagging credal decision trees perform better than other Bagging approaches on data sets with class noise. It will be interesting to generalize our work in this paper to include bagging methods, but such work is left as a possible topic for future research.

*3.2. Adding Noise Methods.* We need a method to add noise to a data set to test classification algorithm performance and robustness with noisy data. Numerous methodologies have been proposed in the literature for introducing noise into data sets. By adding noise to our data sets, we can evaluate how it affects the performance of classification models. This helps us identify which models are robust enough to handle noisy data and enables us to explore ways to improve the performance of classification models when working with noisy data. In this section, we review some techniques used in the literature to add noise to data sets, not only to introduce them but also to justify our choice of noise introduction method.

Zhu and Wu [24] used two techniques to add noise to a class variable, namely, *total random corruption* and *proportional random corruption*. For the first method, they add noise to all classes randomly, with a previously chosen noise level. Therefore, classes of instances are mislabeled based on this noise level. For the second method, when noise is added, the distribution of the class remains unchanged. In this method, if there are $K$ classes, where the classes distribution is denoted by $P_1, P_2, \ldots, P_K$, where $P_1$ is the percentage of the most common class and $P_K$ is the percentage of the least common class, and $P_i \geq P_{i+1}$. To corrupt a noise level of $x.100\%$, random noise is added proportionally to the different classes, where an instance labeled as $i$ has $P_1/P_i.x.100\%$ chance of being changed. It is possible that the actual noise level is lower than the intended corruption level with this method. Zhu and Wu [24] provide additional information and explanations regarding these strategies for introducing noise to data sets.

Zhu and Wu [1] proposed another approach to adding noise to class and attribute variables. To add a particular noise level to the class variable, given a pair of classes and a noise percentage $x\%$, an instance belonging to the first class has a probability of $x\%$ of being changed to the second class, and the same applies to an instance of the second class. When adding noise to attribute variables, given a noise percentage $x\%$, an attribute's value is changed at random (approximately $x\%$ of the time) to other possible values, where each potential value has an equal chance of being selected. For continuous variables, a value is chosen at random from within the range of possible values, bounded by the minimum and maximum values. We refer to Zhu and Wu [1] for additional details regarding this technique.

Sáez et al. [16] have introduced four approaches to add $x\%$ noise level to data sets. For class noise, they introduce a *uniform class noise scheme*, which replaces instances' classes by randomly changing a class with another one from the available classes, and a *pairwise class noise scheme*, which changes instances of the largest class to the second largest class. They employ a *uniform attribute noise scheme* and the *Gaussian attribute noise scheme* for attribute noise. For the first method, to add a specific noise level $x\%$, $x\%$ of the instances are selected, and their values are changed by other possible values from the domain of the attribute. In this scheme, a uniform distribution is employed for choosing the replacement value. The second method is similar to the first one, but it employs a Gaussian distribution. Sáez et al. [16] provide more details and explanations about these methods. Another recent comprehensive review of different methods of adding noise to class variable, attribute variables, or both in combination is given by Sáez [28]. Sáez has also presented an R package which is called *noisemodel* [29]. This R package contains different ways for adding noise to class variable, attribute variables, and both in combination.

A widely used technique for adding noise to data sets is introduced by Abellán et al. [18], Abellán and Masegosa [19, 20], Alharbi [17], Gray and Fan [30], Mantas and Abellán [22, 25], and Mantas et al. [23]. In this technique, they add a particular percentage of random noise to the class variable in the training data only; hence, the test data are left unchanged. To introduce noise into the class variable, follow these steps: first, they randomly select a particular percentage of instances in the training data; then, the class labels for the chosen instances are randomly switched to other possible classes. In this paper, we employ this technique for adding noise to the class variable. Section 4 contains additional information about applying this technique in our work.

## 4. Experimental Analysis

In this section, we study and compare the performance of the C4.5, CART, and RF algorithms when they are applied to noisy data sets. We first describe how the experiments have been conducted and provide a brief overview of the used data sets. Next, we explain the process of adding noise to the class variable. Following that, we present and discuss the

results of the performance of the C4.5, CART, and RF classifiers with noisy class variables.

*4.1. Experimental Setup.* In our experiments, we have used broad and different sets of 20 data sets from the UCI Machine Learning Repository database [31]. The characteristics of these data sets are summarized in Table 1, where column "$N$" represents the total number of instances in the data sets, column "Att." represents the number of attribute variables, "Num." represents the number of numerical attribute variables, column "Nom." represents the number of nominal attribute variables, and column "Classes" represents the number of labels or states of the class variable. Different levels of random noise have been added to the class variable in each data set, and then, the C4.5, CART, and RF algorithms have been applied to the data sets. We use the statistical software R for our experimentations [32]. To implement the RF algorithm in R, we set the default value for the parameter mtry, which is the square root of the attribute variables. Note that mtry is the number of attribute variables randomly chosen as candidates at each split when building the tree. The parameter ntree (the number of built trees) is set to 500. This parameter should not be set to too small numbers to ensure that every instance can be predicted a few times.

Table 1: Data set description.

| Data set | $N$ | Att. | Num. | Nom. | Classes |
|---|---|---|---|---|---|
| Adult | 48842 | 14 | 8 | 6 | 2 |
| Banknote Authentication | 1372 | 4 | 4 | 0 | 2 |
| Blood Transfusion | 748 | 4 | 4 | 0 | 2 |
| Breast Cancer | 699 | 9 | 9 | 0 | 2 |
| Car | 1728 | 6 | 0 | 6 | 4 |
| CMC | 1473 | 9 | 2 | 7 | 3 |
| Congressional Voting | 435 | 16 | 0 | 16 | 2 |
| Dry Bean | 13611 | 16 | 16 | 0 | 7 |
| Ionosphere | 351 | 34 | 34 | 0 | 2 |
| Iris | 150 | 4 | 4 | 0 | 3 |
| Optical Digits | 3823 | 64 | 64 | 0 | 10 |
| Pen-Based Recognition | 10992 | 16 | 16 | 0 | 10 |
| Qualitative Bankruptcy | 250 | 6 | 0 | 6 | 2 |
| Raisin | 900 | 7 | 7 | 0 | 2 |
| Seeds | 210 | 7 | 7 | 0 | 3 |
| Sonar | 208 | 60 | 60 | 0 | 2 |
| Tic-Tac-Toe Endgame | 958 | 9 | 0 | 9 | 2 |
| Vertebral Column | 310 | 6 | 6 | 0 | 4 |
| Wine | 178 | 13 | 13 | 0 | 3 |
| Wireless Indoor | 2000 | 7 | 7 | 0 | 4 |

For these data sets, as in most of the real-world data sets, we do not know how much noise they contain, if any, or which instances may be noisy. Thus, we do not assume any particular level of noise in these data sets; hence, we consider these data sets as noise-free. Therefore, we implement a random corruption method in order to introduce some noise into these data sets. We add the following random noise levels to the class variable: 0%, 10%, 20%, and 30%. These random levels are selected following several researchers in the literature. It is reasonable to add noise up to 30% as in most cases data sets may not contain more noise. Many researchers in the literature have also added noise levels in their experiments up to 30% to either class or attribute variables [17–20, 22, 23, 25, 33].

The performance of the classification algorithms built on the original training set (0% noise) acts as a reference that could be directly compared with the performance of the classification algorithms obtained with different noisy levels of training data. In other words, in order to check the degree of robustness of the classification algorithms with noisy data sets, we compare the accuracy results of the classification algorithms from the original data sets with the performance of classification algorithms from data sets with different levels of noise. Thus, the most robust classification algorithm is the one that obtained the most similar results with noisy data sets, compared to its results with noise-free data sets. This method of comparing and analysing the degree of robustness has also been used by Sáez et al. [16].

To corrupt a class variable, i.e., adding noise into it, $x$% of the instances are selected, where $x$ refers to the noise level we want to add. For adding noise to the class variable, $x$% of the instances in the training set are randomly selected, then their class labels are replaced by another class from the available classes, excluding the original class label. The noise levels are added to the training sets only, while the test sets are left unchanged. Adding noise to only training sets enables us to check the effects of different noise levels of the training set on the performance of the classification algorithms which are based on the data with the noise level, but which are tested on a test data set without noise. This way of adding noise allows direct comparison between the performances of the classification algorithms on equivalent test sets, for increased levels of noise in the training sets. Moreover, the robustness of the classification algorithms can be better studied since the effects of noise are isolated in the training process. Unlike [1, 15], we exclude the original label from the random assignments for the class variable in order to ensure that $x$% of the training set will be changed.

In this experimental analysis, a 10-fold cross-validation scheme has been applied for each data set, and then, the average results have been reported. In order to study and compare the performance of the C4.5, CART, and RF algorithms when dealing with noisy data, we use two evaluation measures. First, we used classification accuracy rate which is the most commonly used method to measure the performance of classification algorithms. It is calculated as the ratio of the total number of correctly classified instances on the testing set to the total number of instances. However, in the case of imbalanced classes, we may use another measure to have more insight into the performance of classification algorithms. $F$-measure is one of the best metrics to consider in such a case. $F$-measure is defined as the harmonic mean of the algorithm's precision and recall. The precision is the total number of true positive instances divided by the total number of all positive instances, and the recall is the total number of true positive instances divided by the number of all instances that should have been identified as positive. Using this method, the $F$-measure can be calculated for binary class variables, but for multiclass

class variables, we use macroaverage $F$-measure (the average of the $F$-measures calculated for each class) as given in [34]. For simplicity, we use the term "$F$-measure" for both cases throughout the paper.

*4.2. Experimental Results.* This section presents the performance results of the C4.5, CART, and RF algorithms with noisy data sets. We compare their performances using the classification accuracy and $F$-measure rates. First, we discuss the classification accuracy for both algorithms, and then, we discuss their performances in terms of the $F$-measure. Finally, we depict the average results using both measures and give comments based on our results.

Table 2 shows the classification accuracy results for the C4.5, CART, and RF classifiers based on noisy data sets with percentages of added random noise equal to 0%, 10%, 20%, and 30%. The classification accuracy results for original data sets (0% noise level) indicate that the RF algorithm performs better than the other algorithms, where the C4.5 algorithm outperforms the CART algorithm in 14 out of 20 data sets. It is also clear that the RF algorithm outperforms the other algorithms with all noise levels. With 10% and 20% noise levels, the accuracy results are quite similar between the C4.5 and CART algorithms. However, the CART algorithm outperforms the C4.5 algorithm with 30% noise level. It is noticed that the CART algorithm tends to outperform the C4.5 algorithm as the noise levels increase. For example, for the Wine data set, the C4.5 achieved 92.15% accuracy rate while the CART algorithm gives only 87.06% accuracy rate. However, with 30% noise level, the CART algorithm is superior to the C4.5 algorithm with 80.35% accuracy rate compared to 68.61% accuracy rate for the C4.5 algorithm. Overall, the RF algorithm acts as the best performing classifier in all cases. By creating trees from multiple subsets of the training set, the RF algorithm decreases the correlation among different classification trees, which could be one of the reasons behind its robustness to noisy instances.

In order to examine the impacts of introducing noise into the class variable more comprehensively, we present $F$-measure results for the C4.5, CART, and RF algorithms in Table 3. Again, the RF algorithm is superior to the other algorithms with and without added noise based on $F$-measure results. When constructing a decision tree, the RF algorithm selects the best-splitting attribute variables from a randomly chosen subset of available attributes [7], this mechanism could enhance the RF algorithm's performance, including its performance on noisy data. For the C4.5 and CART algorithms, the C4.5 algorithm performs better than the CART algorithm on noise-free data sets. However, with added noise into the class variable, the CART algorithm outperforms the C4.5 algorithm. The CART algorithm slightly outperforms the C4.5 algorithm with 10% and 20% noise levels but performs clearly better than the C4.5 algorithm with 30% noise level. For some data sets, such as the Iris, Seeds, Wine, and Wireless Indoor data sets, the C4.5 algorithm outperforms the CART algorithm in terms of $F$-measure when no noise is added, but the CART algorithm performs better with added noise (10%, 20%, and 30% noise levels) to the class variable. This behavior has been also noticed with regard to the classification accuracy rate. The negative impact of class noise on the C4.5 algorithm was the highest. As the level of noise in the data set increased, the performance of the C4.5 algorithm clearly decreased. Generally speaking, the RF algorithm is the best performing with this measure followed by the CART algorithm.

Looking at the average results over all data sets is also interesting. In Figure 1, we can notice the comparative results for the average accuracy and $F$-measure of the C4.5, CART, and RF algorithms when they are applied to data sets with random class noise percentages equal to 0%, 10%, 20%, and 30%. The average results are graphically represented in solid lines for the C4.5 algorithm, in dashed lines for the CART algorithm, and in dotted lines for the RF algorithm. From an average perspective, the RF algorithm outperforms the C4.5 and CART algorithms based on both measures. The C4.5 algorithm performs better than the CART algorithm with an overall classification accuracy rate of 88.42% while the CART algorithm has an overall classification accuracy rate of 86.66% when they are applied to the original data sets. For 10% and 20% noise levels, both algorithms have similar classification accuracy rates. However, for 30% noise level, the CART algorithm has a better classification accuracy rate of 81.00% compared to an accuracy rate of 80.06% for the C4.5 algorithm.

The results of the $F$-measure also indicate a similar behavior for the performances of the C4.5, CART, and RF algorithms. First, it is clear that the RF algorithm is superior to the other algorithms with/without added noise to the class variable. The RF algorithm is a combination of nonrelated decision trees [35], which might enhance its performance on data sets with noisy instances. Second, the CART algorithm outperforms the C4.5 algorithm with all noise levels, while the C4.5 performs better than the CART algorithm only on the original data sets (0% noise level). It is noticed that the impact of adding noise to the class variable negatively affects the C4.5 algorithm more than its effects on the CART algorithm. For the C4.5 algorithm, the difference between its $F$-measure rate on the original data sets and with 30% added noise equals 9.96%, while the difference for the CART algorithm equals 6.3%. This indicates that the CART algorithm is more robust to the presence of noise than the C4.5 algorithm.

Table 4 shows the average time complexity (in seconds) for the C4.5, CART, and RF algorithms on all data sets at varying levels of class noise. The C4.5 and CART algorithms have similar execution time, with slightly less time taken by the CART algorithm. This is not surprising as the CART algorithm produces only binary splitting trees while the C4.5 algorithm might return multisplit trees. The RF algorithm is an ensemble method of trees; hence, it clearly takes more time to execute the algorithm. The time complexity for each data set is given in Table 5. Overall, the CART algorithm achieves comprehensive time efficiency in comparison with the C4.5 and RF algorithms over almost all data sets.

To compare all classification algorithms, we have used a Friedman test [36, 37], with a level of significance of $\alpha = 0.05$. Friedman test is a nonparametric test that is used to

Table 2: Accuracies for the C4.5/CART/RF algorithms on the data sets at varying noise levels.

| Data set | 0% noise | 10% noise | 20% noise | 30% noise |
|---|---|---|---|---|
| Adult | 86.19/84.45/86.64 | 85.82/84.46/85.98 | 85.39/84.34/84.68 | 83.47/83.17/81.00 |
| Banknote Authentication | 98.61/97.15/99.34 | 97.45/94.89/97.74 | 94.96/94.60/92.92 | 91.02/91.75/82.85 |
| Blood Transfusion | 78.19/78.78/74.73 | 77.26/76.47/75.67 | 77.00/76.99/70.72 | 75.26/75.00/67.23 |
| Breast Cancer | 95.14/94.35/96.81 | 92.56/94.57/95.85 | 93.00/94.14/93.71 | 92.13/91.99/89.27 |
| Car | 92.36/94.13/96.80 | 91.09/92.59/95.95 | 88.71/90.80/94.50 | 86.46/88.77/93.92 |
| CMC | 50.88/55.37/55.03 | 49.32/54.76/53.27 | 50.20/52.18/50.41 | 46.46/51.16/49.25 |
| Congressional Voting | 96.36/95.12/96.51 | 94.55/95.45/95.65 | 94.77/94.55/93.55 | 89.91/91.99/87.44 |
| Dry Bean | 91.13/87.06/92.44 | 90.18/87.13/92.15 | 88.41/87.03/91.65 | 83.37/86.41/90.90 |
| Ionosphere | 90.29/87.43/93.14 | 87.43/84.57/92.86 | 83.14/80.57/90.57 | 75.71/75.43/85.71 |
| Iris | 94.67/92.86/95.00 | 92.00/95.33/94.00 | 89.33/92.00/89.33 | 84.00/88.67/80.00 |
| Optical Digits | 89.87/76.91/98.09 | 87.83/75.13/98.04 | 84.16/74.11/98.12 | 75.42/72.51/97.83 |
| Pen-Based Recognition | 95.91/83.02/99.17 | 94.79/81.36/99.20 | 93.26/79.69/98.89 | 87.94/79.41/98.05 |
| Qualitative Bankruptcy | 98.00/98.33/100 | 98.40/98.00/99.20 | 97.60/98.40/94.40 | 95.20/92.40/84.00 |
| Raisin | 85.78/85.56/86.00 | 85.56/85.56/84.89 | 84.78/85.33/81.00 | 83.11/82.11/74.00 |
| Seeds | 91.90/91.50/93.50 | 87.62/90.00/91.43 | 84.76/90.48/90.00 | 80.48/87.62/83.33 |
| Sonar | 76.89/71.00/86.00 | 66.84/68.22/78.85 | 64.64/69.22/80.33 | 62.51/56.69/72.56 |
| Tic-Tac-Toe Endgame | 85.58/92.21/99.05 | 82.15/90.70/97.07 | 77.15/82.44/89.35 | 73.08/76.21/80.59 |
| Vertebral Column | 81.29/84.33/85.33 | 79.03/80.65/85.16 | 78.71/77.74/83.23 | 78.71/79.03/79.68 |
| Wine | 92.15/87.06/97.65 | 81.88/89.93/96.67 | 79.24/83.68/98.33 | 68.61/80.35/91.46 |
| Wireless Indoor | 97.30/96.55/98.40 | 96.20/96.35/98.35 | 95.70/96.00/97.75 | 88.40/95.45/97.00 |

Table 3: *F*-Measure for the C4.5/CART/RF algorithms on the data sets at varying noise levels.

| Data set | 0% noise | 10% noise | 20% noise | 30% noise |
|---|---|---|---|---|
| Adult | 91.13/90.26/91.40 | 90.90/90.27/90.95 | 90.63/90.10/90.03 | 89.32/89.19/87.36 |
| Banknote Authentication | 98.78/97.46/99.42 | 97.70/95.47/98.01 | 95.49/95.10/93.66 | 91.82/92.58/84.52 |
| Blood Transfusion | 86.25/86.90/84.13 | 85.78/85.49/84.59 | 86.33/85.77/81.06 | 85.43/84.17/77.87 |
| Breast Cancer | 96.36/95.72/97.64 | 94.40/95.85/96.86 | 94.67/95.55/95.17 | 94.10/93.83/91.83 |
| Car | 82.14/86.03/91.35 | 78.39/82.37/89.56 | 68.99/78.50/84.50 | 65.18/74.38/82.04 |
| CMC | 48.27/51.61/52.27 | 46.65/52.02/50.85 | 47.58/49.17/47.98 | 44.07/48.88/46.89 |
| Congressional Voting | 97.08/95.86/96.98 | 95.57/96.14/96.35 | 95.74/95.50/94.79 | 91.61/92.95/89.57 |
| Dry Bean | 92.27/88.06/93.57 | 91.18/88.14/93.26 | 88.59/88.01/92.64 | 82.44/87.16/91.65 |
| Ionosphere | 85.85/81.53/90.21 | 81.73/78.14/89.75 | 72.68/74.19/87.14 | 59.35/69.16/79.74 |
| Iris | 94.38/91.20/93.48 | 91.57/94.67/92.86 | 89.16/91.35/88.40 | 82.50/87.66/77.86 |
| Optical Digits | 89.88/76.54/98.07 | 87.74/74.84/97.99 | 83.86/74.12/98.09 | 75.04/71.90/97.83 |
| Pen-Based Recognition | 95.91/82.77/99.18 | 94.75/81.09/99.19 | 93.21/79.14/98.89 | 87.86/78.80/98.06 |
| Qualitative Bankruptcy | 96.99/97.32/100 | 97.89/97.13/98.70 | 96.40/97.51/92.87 | 94.01/90.95/82.52 |
| Raisin | 85.79/85.14/85.45 | 85.03/85.20/84.44 | 84.13/84.60/80.69 | 82.03/82.08/73.25 |
| Seeds | 91.88/91.86/92.79 | 86.96/90.17/91.49 | 84.04/90.08/90.02 | 80.01/87.04/82.81 |
| Sonar | 79.35/73.18/87.79 | 68.30/71.53/80.98 | 66.34/70.92/82.53 | 63.19/57.87/74.55 |
| Tic-Tac-Toe Endgame | 77.85/88.29/98.50 | 73.45/85.87/95.48 | 65.47/71.51/83.87 | 60.48/63.41/71.61 |
| Vertebral Column | 75.45/80.01/80.76 | 73.42/75.11/81.11 | 72.15/70.52/77.96 | 72.86/70.56/73.79 |
| Wine | 91.86/86.65/97.41 | 79.87/88.69/96.97 | 78.37/82.33/98.26 | 66.13/78.88/91.33 |
| Wireless Indoor | 97.28/96.48/98.36 | 96.20/96.29/98.34 | 95.64/95.94/97.73 | 88.17/95.39/97.01 |

compare multiple classification algorithms on multiple data sets. For each data set, the algorithms are ranked by the test, and then, their average ranks are compared. The best performing algorithm received a rank of 1, the second best received a rank of 2, and so on. The null hypothesis stated that all algorithms perform equally. If the null hypothesis is rejected, we may use the post hoc Nemenyi test to compare all the algorithms. For more details and further explanation about the Friedman test, see Demšar [38].

The Friedman ranks of the classification algorithms with different noise levels are shown in Table 6. The RF algorithm achieved the best Friedman rank for 0%, 10%, and 20% noise levels, while for 30% noise levels, the RF and CART algorithms have equal Friedman ranks. The null hypothesis, which assumes that the Friedman ranks of all the classifiers are similar, has been rejected with 0% and 10% noise levels. It has been found that the Friedman ranks of the RF classifier are significantly higher than the ranks of the other classifiers at a significance level of 5%. However, we fail to reject the null hypothesis with 20% and 30% noise levels.

In summary, the robustness of a classification algorithm to noisy data sets is measured by how close its results with added noise to data sets compared to its results with the original data sets. For both evaluation measures, the RF algorithm has the best performance with slightly lower results with added noise to class variable. For the classification

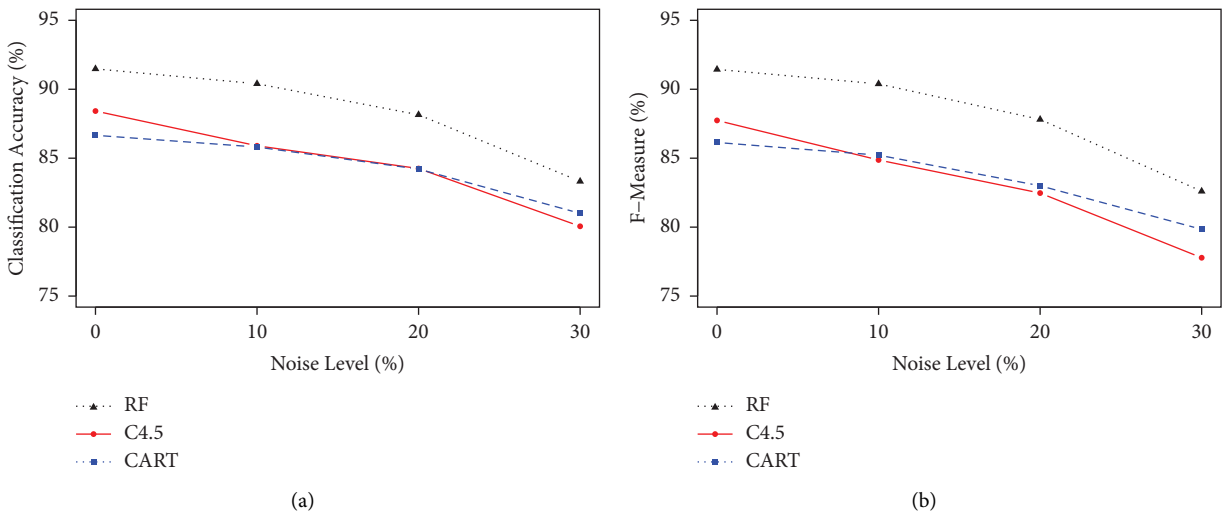


Figure 1: Average accuracy (a) and *F*-measure (b) for the C4.5, CART, and RF algorithms.

Table 4: Average execution time (seconds) for the C4.5/CART/RF algorithms on the data sets at varying noise levels.

| Algorithm | 0% noise | 10% noise | 20% noise | 30% noise |
|---|---|---|---|---|
| C4.5 | 0.134 | 0.179 | 0.21 | 0.223 |
| CART | 0.093 | 0.164 | 0.17 | 0.195 |
| RF | 2.797 | 3.803 | 4.302 | 4.355 |

Table 5: Time complexity for the C4.5/CART/RF algorithms on the data sets at varying noise levels.

| Data set | 0% noise | 10% noise | 20% noise | 30% noise |
|---|---|---|---|---|
| Adult | 0.92/0.96/27.99 | 1.41/1.2/38.98 | 1.85/1.25/47.01 | 1.71/1.45/46.85 |
| Banknote Authentication | 0.03/0.01/0.25 | 0.03/0.02/0.38 | 0.02/0.02/0.44 | 0.02/0.02/0.45 |
| Blood Transfusion | 0.03/0.01/0.19 | 0.02/0.01/0.21 | 0.02/0.01/0.23 | 0.02/0.01/0.22 |
| Breast Cancer | 0.03/0.01/0.16 | 0.03/0.01/0.26 | 0.03/0.02/0.29 | 0.03/0.02/0.29 |
| Car | 0.04/0.01/0.29 | 0.03/0.02/0.5 | 0.03/0.02/0.49 | 0.04/0.02/0.54 |
| CMC | 0.05/0.04/0.6 | 0.05/0.03/0.66 | 0.04/0.03/0.64 | 0.05/0.03/0.65 |
| Congressional Voting | 0.06/0.01/0.11 | 0.04/0.01/0.13 | 0.09/0.02/0.13 | 0.04/0.01/0.13 |
| Dry Bean | 0.62/0.8/10.62 | 0.76/1.01/13.71 | 0.97/1.22/15.36 | 1.32/1.42/15.77 |
| Ionosphere | 0.09/0.03/0.29 | 0.04/0.03/0.31 | 0.04/0.03/0.34 | 0.04/0.06/0.39 |
| Iris | 0.02/0.01/0.04 | 0.02/0.01/0.05 | 0.02/0.01/0.04 | 0.02/0.01/0.05 |
| Optical Digits | 0.29/0.36/8.62 | 0.48/0.58/11.34 | 0.41/0.39/10.81 | 0.46/0.41/11.18 |
| Pen-Based Recognition | 0.22/0.19/5.27 | 0.35/0.22/7.51 | 0.41/0.24/8.04 | 0.44/0.27/8.3 |
| Qualitative Bankruptcy | 0.03/0.01/0.04 | 0.03/0.01/0.06 | 0.03/0.01/0.07 | 0.03/0.01/0.07 |
| Raisin | 0.03/0.02/0.28 | 0.06/0.02/0.31 | 0.02/0.02/0.35 | 0.02/0.03/0.35 |
| Seeds | 0.03/0.01/0.06 | 0.02/0.01/0.07 | 0.02/0.01/0.08 | 0.02/0.01/0.08 |
| Sonar | 0.06/0.03/0.25 | 0.05/0.03/0.26 | 0.05/0.03/0.29 | 0.06/0.04/0.28 |
| Tic-Tac-Toe Endgame | 0.04/0.02/0.22 | 0.04/0.02/0.27 | 0.04/0.02/0.3 | 0.04/0.02/0.33 |
| Vertebral Column | 0.03/0.01/0.11 | 0.03/0.01/0.12 | 0.02/0.01/0.16 | 0.02/0.01/0.13 |
| Wine | 0.03/0.01/0.07 | 0.03/0.01/0.08 | 0.03/0.01/0.09 | 0.02/0.01/0.11 |
| Wireless Indoor | 0.03/0.02/0.47 | 0.05/0.03/0.85 | 0.06/0.03/0.88 | 0.06/0.03/0.92 |

Table 6: Friedman's average rankings of the algorithms on the data sets with different noise levels, for a level of significance of 0.05.

| Algorithm | 0% noise | 10% noise | 20% noise | 30% noise |
|---|---|---|---|---|
| C4.5 | 2.25 | 2.45 | 2.15 | 2.1 |
| CART | 2.65 | 2.2 | 2.05 | 1.95 |
| RF | 1.1 | 1.3 | 1.75 | 1.95 |

accuracy rate, the variance of accuracy rates with added noise levels does not decrease so quickly in the CART algorithm, while this is not the case for the C4.5 algorithm. This also corresponds to the *F*-measure rates, where we noticed that the C4.5 algorithm's performance sharply declines when adding noise levels to the class variable. In

consequence, this is the reason why the CART algorithm possesses more robustness to noisy class variable because it has a lower variance in these situations. Hence, we can say that the CART algorithm is more robust to noisy class variable than the C4.5 algorithm. The binary splitting technique performed by CART algorithm might be one of the reasons for its superiority over the C4.5 algorithm. Overall, the evaluation results prove that it is better to consider the RF algorithm in applications where noisy data could be present. The robustness of the RF algorithm to data noise compared to other traditional classification tree algorithms relies on that the RF algorithm only uses a subset of the available instances in the classification made by each of the single trees. Consequently, the probability that the trees could be affected by noise is lower than that of the algorithms using the entire data set [39]. By taking only the C4.5 and CART algorithms into account, it is better to use the C4.5 algorithm when constructing decision trees on data sets where it is expected that noise is not present. However, when data sets might contain some noise the CART algorithm is preferable to use for constructing decision trees.

## 5. Conclusions

In this paper, we have studied the performance of the C4.5, CART, and RF classification algorithms when different noise levels are added to the class variable. In order to do this, two evaluation measures have been used to evaluate and compare both algorithms which are the classification accuracy and $F$-measure rates. As real-world data sets often contain noise that negatively affects the classification performance, it is important to identify classification algorithms that can handle noise effectively. The results obtained have shown that the RF algorithm is the most robust algorithm with regard to class noise in the data sets followed by the CART algorithm. However, the results have shown that the C4.5 algorithm performs better than the CART algorithm with clean data sets. Overall, based on the averaged results of the accuracy and $F$-measure for the testing sets, the RF algorithm was shown to provide excellent results for predicting unknown instances with and without noisy class variables. Therefore, we strongly suggest using the RF algorithm for classifying instances that may contain some noise in the class variable.

In future work, it would be valuable to delve deeper into attribute noise's impact on these algorithms' performance. This is because there has been comparatively less focus and study on attribute noise in the literature than on class noise. It will be also of interest to extend this work by studying the classification performance when both attribute and class noise are introduced simultaneously. Another idea for future research is to consider other classification methods such as Naive Bayes and support vector machines by comparing their performances with the decision tree method with noisy data sets.

## Data Availability

The data sets used to support the findings of this study are available at https://archive.ics.uci.edu/.

## Conflicts of Interest

The author declares that there are no conflicts of interest.

## References


[1] X. Zhu and X. Wu, "Class noise vs attribute noise: a quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004.

[2] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[3] B. Frénay and A. Kabán, "A comprehensive introduction to label noise," in *Proceedings of the 22nd European Symposium on Artificial Neural Networks*, pp. 667–676, Citeseer, Bruges, Belgium, October, 2014.

[4] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial Intelligence Review*, vol. 33, no. 4, pp. 275–306, 2010.

[5] J. Ross, *Quinlan. C4.5: Programs for Machine Learning*, Morgan Kuafmann, Burlington, MA, USA, 1993.

[6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, USA, 1984.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[8] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[9] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, 1948.

[10] R. Changala, A. Gummadi, G. Yedukondalu, and U. N. P. G. Raju, "Classification by decision tree induction algorithm to learn decision trees from the class-labeled training tuples," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, pp. 427–434, 2012.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112, Springer, Berlin, Germany, 2013.

[12] A. Ghosh and A. Senthilrajan, "Comparison of machine learning techniques for spam detection," *Multimedia Tools and Applications*, vol. 82, no. 19, pp. 29227–29254, 2023.

[13] J. Cho and S. Kim, "Personal and social predictors of use and non-use of fitness/diet app: application of random forest algorithm," *Telematics and Informatics*, vol. 55, Article ID 101301, 2020.

[14] V. Vagin and M. Fomina, "Problem of knowledge discovery in noisy databases," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 3, pp. 135–145, 2011.

[15] C.-M. Teng, "Correcting Noisy Data," *Machine Learning*, vol. 99, pp. 239–248, 1999.

[16] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "Tackling the problem of classification with noisy data using multiple classifier systems: analysis of the performance and robustness," *Information Sciences*, vol. 247, pp. 1–20, 2013.

[17] A. A. H. Alharbi, *Direct nonparametric predictive inference classification trees*, PhD thesis, Durham University, Durham, UK, 2022.

[18] J. Abellán, C. J. Mantas, and J. G. Castellano, "Adaptativecc4.5: credal c4.5 with a rough class noise estimator," *Expert Systems with Applications*, vol. 92, pp. 363–379, 2018.

[19] J. Abellán and A. R. Masegosa, "Bagging decision trees on data sets with classification noise," in *Proceedings of the*

*International Symposium on Foundations of Information and Knowledge Systems*, pp. 248–265, Springer, Helsinki, Finland, June, 2010.

[20] J. Abellán and A. R. Masegosa, "Bagging schemes on the presence of class noise in classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6827–6837, 2012.

[21] C. Catal, O. Alan, and K. Balkan, "Class noise detection based on software metrics and roc curves," *Information Sciences*, vol. 181, no. 21, pp. 4867–4877, 2011.

[22] C. J. Mantas and J. Abellán, "Credal-c4.5: decision tree based on imprecise probabilities to classify noisy data," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4625–4637, 2014.

[23] C. J. Mantas, J. Abellán, and J. G. Castellano, "Analysis of credal-c4.5 for classification in noisy domains," *Expert Systems with Applications*, vol. 61, pp. 314–326, 2016.

[24] X. Zhu and X. Wu, "Cost-guided class noise handling for effective cost-sensitive learning," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, pp. 297–304, IEEE, Brighton, UK, November, 2004.

[25] C. J. Mantas and J. Abellán, "Analysis and extension of decision trees based on imprecise probabilities: application on noisy data," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2514–2525, 2014.

[26] T. M. Khoshgoftaar and J. Van Hulse, "Empirical case studies in attribute noise detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 4, pp. 379–388, 2009.

[27] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[28] J. A. Sáez, "Noise models in classification: unified nomenclature, extended taxonomy and pragmatic categorization," *Mathematics*, vol. 10, no. 20, p. 3736, 2022.

[29] J. A. Sáez, "Noise simulation in classification with the noisemodel r package: applications analyzing the impact of errors with chemical data," *Journal of Chemometrics*, vol. 37, no. 5, p. e3472, 2023.

[30] J. B. Gray and G. Fan, "Classification tree analysis using target," *Computational Statistics and Data Analysis*, vol. 52, no. 3, pp. 1362–1372, 2008.

[31] D. Dua and C. Graff, *UCI Machine Learning Repository*, University of California, School of Information and Computer Sciences, Irvine, CA, USA, 2019.

[32] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013.

[33] P. Li, X. Wu, X. Hu, Q. Liang, and Y. Gao, "A random decision tree ensemble for mining concept drifts from noisy data streams," *Applied Artificial Intelligence*, vol. 24, no. 7, pp. 680–710, 2010.

[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.

[35] L. Zhu, D. Qiu, D. Ergu, C. Ying, and K. Liu, "A study on predicting loan default based on the random forest algorithm," *Procedia Computer Science*, vol. 162, pp. 503–513, 2019.

[36] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[37] M. Friedman, "A comparison of alternative tests of significance for the problem of $m$ Rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[38] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[39] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 93–104, 2012.