On the Performance of Nonparametric Predictive Inference-Based Classification Trees under Noisy Data

Masad A. Alrasheedi[a], Tahani Coolen-Maturi[b,*], Frank P.A. Coolen[b]

[a]*Department of Management Information Systems, College of Business Administration, Taibah University, Madinah, Saudi Arabia*
[b]*Department of Mathematical Sciences, Durham University, Durham, UK*

Abstract

In data mining, classification assigns new observations to one of a set of predefined classes based on their attributes. Among classification methods, classification trees are widely used because their rules are straightforward to understand and interpret. These trees are constructed recursively in a top-down manner through repeated splits of the training dataset, which is a subset of the full dataset.

In recent years, Nonparametric Predictive Inference (NPI) has been introduced for selecting optimal thresholds in two- and three-class classification problems. NPI expresses inferences explicitly in terms of a specified number of future observations and target proportions. These target proportions allow the assignment of weights that reflect the relative importance of one class over another. The NPI-based threshold selection method has previously been applied to the construction of classification trees.

In this study, we present applications of the NPI_2-Tree and NPI_3-Tree classification algorithms to noisy datasets. Noise arises when datasets contain incorrect values in either attribute variables or the class variable. We evaluate the performance of the NPI_2-Tree and NPI_3-Tree algorithms under varying levels of noise added to the class variable. The results show that our algorithms perform well under noisy conditions and remain robust across most noise levels compared with other classification methods.

Keywords: Nonparametric Predictive Inference, Classification, Classification Trees, Optimal Thresholds, Noisy Data

1. Introduction

Classification is a fundamental task in supervised machine learning and data mining, where the aim is to assign new observations to one of a set of predefined classes based on their attribute values. Among the many available classification methods, classification trees remain widely used because they provide transparent decision rules and are easy to interpret, even for users with limited technical expertise. Their simple hierarchical structure also

*Corresponding author
Email addresses: mrshedi@taibahu.edu.sa (Masad A. Alrasheedi), tahani.maturi@durham.ac.uk (Tahani Coolen-Maturi), frank.coolen@durham.ac.uk (Frank P.A. Coolen)

makes them attractive in practical applications where interpretability is important alongside predictive performance.

In many real-world applications, however, the data used to train classifiers are imperfect. A common imperfection is noise, which may affect either the attribute values or the class labels. Noise in class labels, often referred to as class noise or label noise, can arise from measurement errors, incorrect data entry, subjective human judgement, ambiguous cases, or inconsistent annotation processes. Previous studies have shown that class noise often has a more harmful effect on classification performance than attribute noise because each observation typically contains many predictors but only one class label, and this label plays a central role in the learning process.

A variety of approaches have been proposed in the literature to address classification under noisy data. One direction focuses on designing classifiers that are inherently more robust to mislabeled observations. Another direction evaluates how the predictive performance of existing algorithms deteriorates as the level of noise increases. Within classification trees, this problem has been studied for classical methods such as ID3, C4.5, and CART, as well as for methods based on imprecise probabilities, including IDM- and NPI-M-based trees. More recently, related robustness issues have also been investigated outside the tree framework, for example through fuzzy and weighted k-nearest neighbour methods that aim to reduce the influence of noisy or unreliable observations.

Recently, new classification-tree algorithms based on Nonparametric Predictive Inference (NPI) have been introduced for selecting optimal thresholds in two-class and three-class problems, leading to the NPI_2-Tree and NPI_3-Tree algorithms. These methods provide an alternative framework for tree construction by using the NPI approach to support threshold selection and class discrimination. Although these algorithms have been proposed previously, their behaviour in the presence of class noise has not yet been systematically studied. In particular, it remains unclear how robust they are relative to established classical and imprecise-probability-based tree methods when class labels are corrupted.

The present paper addresses this gap through a systematic experimental investigation of the NPI_2-Tree and NPI_3-Tree algorithms under controlled class-noise settings. We artificially introduce different levels of class noise into a range of datasets in order to evaluate the effect of mislabeling on predictive performance. The NPI-based algorithms are compared with several well-known tree-based classifiers, namely C4.5, CART, NPI-M, and IDM1. In addition to experiments with randomly introduced class noise, we also examine structured class-dependent noise scenarios for the three-class case. The contribution of this paper is therefore not the proposal of a new classification algorithm, but rather a detailed robustness study of recently developed NPI-based classification trees under noisy data, together with a comparison against relevant tree-based baselines.

The remainder of the paper is organised as follows. Section 2 reviews previous work on classification under noisy data and positions the present study within the literature. Section 3 introduces the NPI framework and the threshold-selection method for two- and three-class problems. Section 4 presents the NPI-based tree-construction procedure. Section 5 reports the experimental results for random class noise. Section 6 examines the performance of the NPI_3-Tree algorithm under class-dependent noise scenarios for ordered three-class problems. Section 7 concludes the paper and outlines directions for future research.

2. Related Work on Classification Under Noisy Data

Classification trees are among the most widely used supervised learning methods because of their interpretability and straightforward decision structure. Classical tree algorithms such as ID3 [33], C4.5 [34], and CART [18] differ mainly in the criteria used to select splits, but all recursively partition the feature space in order to improve class separation. These methods have been applied successfully in many domains; however, their performance can deteriorate when the training data contain mislabeled observations. In particular, incorrect class labels may distort impurity measures and lead to suboptimal splits during tree induction.

To address uncertainty more explicitly, several tree-construction methods based on imprecise probabilities have been proposed. For example, Imprecise Information Gain (IIG) was introduced by Abellán and Moral [4] as a split-selection criterion that replaces precise probabilities and entropy measures with imprecise counterparts. This framework can be derived from either the Imprecise Dirichlet Model (IDM) [3] or Nonparametric Predictive Inference for multinomial data (NPI-M) [8], leading to tree algorithms that aim to be more cautious in the presence of uncertainty. More recently, Direct Nonparametric Predictive Inference (D-NPI) classification trees have been proposed by Alharbi et al. [11] as a fully NPI-based tree-induction approach. These methods are particularly relevant in noisy environments because they attempt to reduce overconfident decisions when the available information is limited or uncertain. Although these methods provide useful alternatives to classical tree induction, the literature remains limited with respect to the behaviour of recently introduced NPI threshold-based trees under label noise. In particular, the robustness of the NPI_2-Tree and NPI_3-Tree algorithms has not yet been systematically assessed across multiple levels and structures of class noise. This is one of the main issues addressed in the present paper.

The problem of classification with noisy data has also been studied outside the decision-tree setting. In recent years, several studies have proposed more robust variants of the k-nearest neighbour (kNN) classifier [2], including fuzzy, local-mean-based [15], and weighted kNN approaches [1]. These methods generally aim to reduce the harmful effects of noisy labels, class imbalance, and outlying observations by assigning different weights to neighbours or by replacing individual neighbours with more stable local summaries. Such approaches represent recent attempts to improve classification robustness under difficult data conditions. At the same time, they differ fundamentally from classification-tree methods in terms of model structure, decision mechanism, interpretability, and computational procedure. kNN-based classifiers are instance-based and rely on neighbourhood comparisons at prediction time, whereas classification trees produce an explicit hierarchical rule structure through recursive partitioning of the training data. As a result, strengths demonstrated in one framework do not automatically transfer to the other. For this reason, the present study focuses on the robustness of tree-based classifiers, and especially on NPI-based tree construction, rather than attempting to unify all classifier families within a single modelling framework.

The existing literature therefore highlights two important points. First, robustness to class noise is an established and important topic in classification research. Second, while both classical and imprecise-probability tree methods have been studied, and while robust non-tree methods have also been proposed, the specific robustness properties of NPI_2-Tree and NPI_3-Tree under controlled noisy-label settings remain insufficiently understood. The

present paper is positioned within this gap. Its goal is not to introduce a new classifier, but to provide a structured robustness study of recently proposed NPI-based tree algorithms under noisy data. To do so, the paper evaluates NPI_2-Tree and NPI_3-Tree under multiple random noise levels and, for the three-class case, under class-dependent noise scenarios, and compares their performance with established tree-based baselines. In this way, the study contributes new empirical evidence on the behaviour of NPI-based classification trees in the presence of mislabeled observations.

3. Nonparametric Predictive Inference (NPI) Framework

3.1. NPI for real-valued observations

Nonparametric Predictive Inference (NPI) is a statistical methodology based on Hill's assumption $A_{(n)}$ [28], which gives direct probabilities for one or more future observations based on n observed values of related random quantities. Inference based on $A_{(n)}$ is nonparametric and predictive. It was introduced particularly for situations where there is no prior information about the probability distribution for a random quantity of interest, or in cases where one explicitly does not want to use any such information. Let $X_1, \ldots, X_n, X_{n+1}$ be exchangeable real-valued random quantities. Suppose the ordered observed values of $X_1, X_2, \ldots, X_n$ are denoted by $x_1 < x_2 < \ldots < x_n$, where we assume that no ties occur among observations. For notational convenience, define $x_0 = -\infty$ and $x_{n+1} = \infty$. Note that $x_{n+1} = \infty$ is introduced purely for notation and does not represent an observation of the variable X_{n+1}.

These n ordered observations divide the real-line into $n+1$ open intervals $I_j = (x_{j-1}, x_j)$, for $j = 1, 2, \ldots, n+1$. The assumption $A_{(n)}$ states that the future observation X_{n+1} falls equally likely in any interval (x_{j-1}, x_j), for each $j = 1, 2, \ldots, n+1$, that is

$$P(X_{n+1} \in I_j) = \frac{1}{n+1} \tag{1}$$

It is important to emphasise that Hill's assumption $A_{(n)}$ does not make any further assumptions on the distribution of probability $\frac{1}{n+1}$ within an interval I_j. NPI uses $A_{(n)}$ for predictive inferences about future observations in the form of lower and upper probabilities, also known as imprecise probabilities. Augustin and Coolen [17] introduced predictive lower and upper probabilities for events of interest based on assumption $A_{(n)}$, which is essentially an application of De Finetti's fundamental theorem of probability [22]. The lower and upper probabilities for the event $X_{n+1} \in B$, with $B \subset \mathbb{R}$, based on the intervals I_j, $j = 1, 2, \ldots, n+1$, and Hill's assumption $A_{(n)}$, are given by

$$\underline{P}(X_{n+1} \in B) = \frac{1}{n+1} \sum_{j=1}^{n+1} \mathbf{1}\{I_j \subseteq B\} \tag{2}$$

$$\overline{P}(X_{n+1} \in B) = \frac{1}{n+1} \sum_{j=1}^{n+1} \mathbf{1}\{I_j \cap B \neq \emptyset\} \tag{3}$$

where $\mathbf{1}\{A\}$ is equal to 1 if A is true and equal to 0 else. The NPI lower probability (2) is obtained by taking only probability mass into account that is necessary within B, which is

only the case for the probability mass $\frac{1}{n+1}$ per interval I_j if this interval is totally contained within B. The NPI upper probability (3) is obtained by taking all probability mass into account that could possibly be within B, which is the case for the probability mass $\frac{1}{n+1}$ per interval I_j if the intersection of I_j and B is non-empty.

We are interested in $m \geq 1$ future observations, X_{n+i} for $i = 1, \ldots, m$ [19]. The data and future observations are linked by consecutive application of $A_{(n)}, A_{(n+1)}, \ldots, A_{(n+m-1)}$ [28]. These together are referred to as the $A_{(.)}$ assumptions, which can be considered a post-data version of a finite exchangeability assumption for $n + m$ random quantities. The $A_{(.)}$ assumptions imply that all possible orderings of n data observations and m future observations are equally likely, where the n data observations are not distinguished from one another, and neither are the m future observations. Let $S_j = \#\{X_{n+i} \in I_j, i = 1, \ldots, m\}$, then assuming $A_{(.)}$ we have [19]

$$P(\bigcap_{j=1}^{n+1} \{S_j = s_j\}) = \binom{n+m}{n}^{-1} \tag{4}$$

where s_j are non-negative integers with $\sum_{j=1}^{n+1} s_j = m$. Equation (4) implies that all $\binom{n+m}{n}$ orderings of m future observations among the n observations are equally likely. Let $X_{(r)}$, for $r = 1, \ldots, m$, be the r-th ordered future observation, so $X_{(r)} = X_{n+i}$ for one $i = 1, \ldots, m$ and $X_{(1)} < X_{(2)} < \cdots < X_{(m)}$. The probabilities given in Equation (5) are based on Equation (4) and derived by counting the relevant orderings, and hold for $j = 1, \ldots, n + 1$, and $r = 1, \ldots, m$ [19],

$$P(X_{(r)} \in I_j) = \binom{j+r-2}{j-1} \binom{n-j+1+m-r}{n-j+1} \binom{n+m}{n}^{-1} \tag{5}$$

For this $X_{(r)} \in I_j$, NPI gives a precise probability, as each of the $\binom{n+m}{n}$ equally likely orderings of n past and m future observations has the r-th ordered future observation in precisely one interval I_j [20]. The event that the number of future observations in an interval (x_α, x_β), with $1 \leq \alpha < \beta \leq n + 1$, and denoted by $S^m_{\alpha,\beta}$, is greater than or equal to a particular value $v \in \mathbb{N}$, has the following precise probability [12],

$$P(S^m_{\alpha,\beta} \geq v) = \sum_{i=v}^{m} \binom{n+m}{n}^{-1} \binom{\beta-\alpha-1+i}{i} \binom{n-\beta+\alpha+m-i}{m-i} \tag{6}$$

We will use these results in Sections 3.2 and 3.3 to select the optimal thresholds.

3.2. NPI-based threshold selection for two classes

The NPI method for selecting the optimal threshold t for a real-valued random quantity and for a two-class classification scenario has been introduced by Alabdulhadi [10] and Coolen-Maturi et al. [21]. This method is different from the classical methods as it selects the optimal threshold value, which focuses on a number of future observations to which the threshold will be applied. Assume that we have a continuous random variable X whose values belong to two classes, C_1 and C_2, and small values of X are more likely to belong to class C_1, i.e. $X \leq t$, and large values of X are more likely to belong to class C_2, i.e. $X > t$.

Let n_1 denote the number of observations for class C_1 and n_2 the number of observations for class C_2. It is assumed that there is full independence between the two classes, meaning that any information about the observations in one class does not contain information about the observations in the other class. In other words, any information about random quantities from one class does not affect any (lower and upper) probabilities for events involving only random quantities of the other class. Let $x_1^1 < x_2^1 < \cdots < x_{n_1}^1$ denote the ordered data from class C_1 and $x_1^2 < x_2^2 < \cdots < x_{n_2}^2$ denote the ordered data from class C_2. For ease of notation, let $x_0^1 = x_0^2 = -\infty$ and $x_{n_1+1}^1 = x_{n_2+1}^2 = \infty$. The data for C_1 partition the real-line into $n_1 + 1$ intervals $I_i^1 = (x_{i-1}^1, x_i^1)$, for $i = 1, 2, \ldots, n_1 + 1$, and the data for C_2 partition the real-line into $n_2 + 1$ intervals, for $I_j^2 = (x_{j-1}^2, x_j^2)$, for $j = 1, \ldots, n_2 + 1$. Throughout this paper, we assume that there are no ties among data observations, i.e., no two or more observations have the same value. When ties occur, they can be resolved by adding a very small amount—typically close to zero—to the tied observations. This is a common method for handling ties in statistics [27].

Since NPI inferences are based on multiple future observations, we consider m_1 future observations from class C_1, denoted by $X_{n_1+r}^1$ for $r = 1, \ldots, m_1$, and m_2 future observations from class C_2, denoted by $X_{n_2+s}^2$ for $s = 1, \ldots, m_2$. Let the ordered future observations be written as $X_{(1)}^1 < X_{(2)}^1 < \cdots < X_{(m_1)}^1$ for class C_1 and $X_{(1)}^2 < X_{(2)}^2 < \cdots < X_{(m_2)}^2$ for class C_2. Following [10, 21], the threshold value t is then chosen to provide the best classification based on these future observations.

For a given threshold t, let L_t^1 denote the number of correctly classified future observations from class C_1, i.e., those with values $X_{n_1+r}^1 \leq t$ for $r = 1, \ldots, m_1$. Similarly, let L_t^2 denote the number of correctly classified future observations from class C_2, i.e., those with values $X_{n_2+s}^2 > t$ for $s = 1, \ldots, m_2$. Let $a, b \in (0, 1]$ represent the relative importance of correct classification for classes C_1 and C_2, respectively. We are interested in the event that at least am_1 future observations from C_1 and at least bm_2 future observations from C_2 are correctly classified, that is $L_t^1 \geq am_1$ and $L_t^2 \geq bm_2$. The choice of a and b reflects the decision maker's judgment about the relative importance of the two classes. These values are unrestricted aside from lying in $(0, 1]$. Choosing $a = b$ implies equal importance, whereas assigning larger or smaller values can emphasize one class over the other and may influence classification performance.

Under the independence assumption between the two classes, the joint NPI lower and upper probabilities are obtained as the products of the corresponding NPI lower and upper probabilities for the events involving L_t^1 or L_t^2 as follows [10, 21]

$$\underline{P}\left(L_t^1 \geq am_1, L_t^2 \geq bm_2\right) = \underline{P}\left(L_t^1 \geq am_1\right) \times \underline{P}\left(L_t^2 \geq bm_2\right) \tag{7}$$

$$\overline{P}\left(L_t^1 \geq am_1, L_t^2 \geq bm_2\right) = \overline{P}\left(L_t^1 \geq am_1\right) \times \overline{P}\left(L_t^2 \geq bm_2\right) \tag{8}$$

The NPI lower and upper probabilities in Equations (7) and (8) are derived using NPI for multiple future observations as given in Section 3.1, in particular Equation (5), as shown below. It is noticed that the event $L_t^1 \geq am_1$ is equal to $X_{(\lceil am_1 \rceil)}^1 \leq t$, where $\lceil am_1 \rceil$ denotes the smallest integer greater than or equal am_1. Similarly, the event $L_t^2 \geq bm_2$ is equal to $X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t$. To show how to use the Equation (5) of the NPI results for multiple future observations, we first consider class C_1 and then class C_2.

For $I_i^1 = (x_{i-1}^1, x_i^1), i = 1, \ldots, n_1 + 1$, and $t \in I_{i_t}^1 = (x_{i_t-1}^1, x_{i_t}^1)$, where $i_t = 2, \ldots, n_1$ is defined as that interval $I_{i_t}^1$ which contains t, the NPI lower and upper probabilities for the event $L_t^1 \geq am_1$ are given as follow [10, 21]:

$$\underline{P}(L_t^1 \geq am_1) = \underline{P}(X_{(\lceil am_1 \rceil)}^1 \leq t) = \sum_{i=1}^{i_t - 1} P(X_{(\lceil am_1 \rceil)}^1 \in I_i^1) \tag{9}$$

$$\overline{P}(L_t^1 \geq am_1) = \overline{P}(X_{(\lceil am_1 \rceil)}^1 \leq t) = \sum_{i=1}^{i_t} P(X_{(\lceil am_1 \rceil)}^1 \in I_i^1) \tag{10}$$

where the precise probabilities on the right-hand sides of Equations (9) and (10) are obtained from Equation (5). For $i_t = 1$, we have $\underline{P}(L_t^1 \geq am_1) = 0$ and $\overline{P}(L_t^1 \geq am_1) = P(X_{(\lceil am_1 \rceil)}^1 \in I_1^1)$, and for $i_t = n_1 + 1$, we have $\underline{P}(L_t^1 \geq am_1) = 1 - P(X_{(\lceil am_1 \rceil)}^1 \in I_{n_1+1}^1)$ and $\overline{P}(L_t^1 \geq am_1) = 1$. If t is equal to one of the observations x_i^1, i.e. $t = x_{i_t}^1$, then this event has precise probability,

$$P(L_t^1 \geq am_1) = P(X_{(\lceil am_1 \rceil)}^1 \leq t) = \sum_{i=1}^{i_t} P(X_{(\lceil am_1 \rceil)}^1 \in I_i^1) \tag{11}$$

Of course, this implies that we have for such a value of t that $\underline{P}(L_t^1 \geq am_1) = \overline{P}(L_t^1 \geq am_1) = P(L_t^1 \geq am_1)$, in this case. Similarly, the NPI lower and upper probabilities for the event $L_t^2 \geq bm_2$ are derived. For $I_j^2 = (x_{j-1}^2, x_j^2), j = 1, \ldots, n_2 + 1$, and $t \in I_{j_t}^2 = (x_{j_t-1}^2, x_{j_t}^2), j_t = 2, \ldots, n_2$, the NPI lower and upper probabilities for the event $L_t^2 \geq bm_2$ are

$$\underline{P}(L_t^2 \geq bm_2) = \underline{P}(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t) = \sum_{i=j_t+1}^{n_2+1} P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 \in I_j^2) \tag{12}$$

$$\overline{P}(L_t^2 \geq bm_2) = \overline{P}(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t) = \sum_{i=j_t}^{n_2+1} P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 \in I_j^2) \tag{13}$$

For $j_t = 1$, we have

$$\underline{P}(L_t^2 \geq bm_2) = 1 - P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 \in I_1^2) \text{ and } \overline{P}(L_t^2 \geq bm_2) = 1.$$

And for $j_t = n_2 + 1$, we have

$$\underline{P}(L_t^2 \geq bm_2) = 0 \text{ and } \overline{P}(L_t^2 \geq bm_2) = P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 \in I_{n_2+1}^2)$$

Furthermore, when $t = x_{j_t}^2$

$$P(L_t^2 \geq bm_2) = P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t) = \sum_{i=j_t+1}^{n_2+1} P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 \in I_j^2) \tag{14}$$

so

$\underline{P}(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t) = \overline{P}(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t) = P(X_{(m_2 - \lceil bm_2 \rceil + 1)}^2 > t)$, if $t = x_{j_t}^2$.

The optimal threshold t for the two classes is obtained by maximizing either the NPI lower probability in Equation (7) or the NPI upper probability in Equation (8). Since these represent different criteria, they may yield different optimal thresholds. In this paper, we focus on the threshold based on the NPI lower probability (Equation (7)) for building classification trees, as lower probabilities represent evidence in favour of events, whereas upper probabilities represent evidence against events.

3.3. NPI-based thresholds selection for three ordered classes

In Section 3.2, we presented the NPI method for selecting the optimal threshold for two classes. Here, we extend this approach to three ordered classes, as introduced in [10, 21]. A naive approach would be to apply the two-class NPI method twice: first to determine t_1 using classes C_1 and C_2, and then to determine t_2 using classes C_2 and C_3. However, due to the assumed ordering of the classes, this procedure does not guarantee the condition $t_1 < t_2$. Therefore, instead of applying the method separately, we use the NPI approach developed specifically for three ordered classes, which simultaneously determines the optimal thresholds t_1 and t_2. We begin by summarising the results of [10, 21], using the same notation as in Section 3.2, with additional notation for class C_3.

Let n_3 denote the number of observations in class C_3, the ordered data from this class are denoted by $x_1^3 < x_2^3 < \cdots < x_{n_3}^3$. For ease of notation, we define $x_0^3 = -\infty$ and $x_{n_3+1}^3 = \infty$. Again, the n_3 observations divide the real-line into $n_3 + 1$ intervals $I_l^3 = (x_{l-1}^3, x_l^3)$, for $l = 1, 2, \ldots, n_3 + 1$. Let m_3 denote the number of future observations in class C_3, with random variable $X_{n_3+d}^3$, for $d = 1, \ldots, m_3$. Let the m_3 ordered future observations from class C_3 be denoted by $X_{(1)}^3 < X_{(2)}^3 < \cdots < X_{(m_3)}^3$. To classify observations into one of the classes, C_1, C_2 or C_3, we want to find the two optimal thresholds t_1 and t_2, where $t_1 < t_2$, such that observations less than or equal to t_1 are classified as belonging to C_1, observations greater than t_1 and less than or equal to t_2 are classified as belonging to C_2 and observations greater than t_2 are classified as belonging to C_3. For particular values of t_1 and t_2, we denote the number of correctly classified future observations from classes C_1, C_2, and C_3 by $L_{t_1}^1$, $L_{(t_1,t_2)}^2$, and $L_{t_2}^3$, respectively. Let a, b, and c denote the target proportions that reflect the desired importance of correctly classifying each class. The choice of these values depends on the decision maker's beliefs about the relative importance of the classes, and they are only constrained to lie in $(0, 1]$. If equal importance is assumed, one may simply set $a = b = c$. The general event of interest for the three classes is that the number of correctly classified future observations from C_1 is at least am_1, from C_2 is at least bm_2, and from C_3 is at least cm_3; that is, $L_{t_1}^1 \geq am_1$, $L_{(t_1,t_2)}^2 \geq bm_2$ and $L_{t_2}^3 \geq cm_3$.

Using the assumption of independence between the three ordered classes, the NPI lower probability for the event of interest is

$$\begin{aligned} &\underline{P}(L_{t_1}^1 \geq am_1, L_{(t_1,t_2)}^2 \geq bm_2, L_{t_2}^3 \geq cm_3) = \\ &\underline{P}(L_{t_1}^1 \geq am_1) \times \underline{P}(L_{(t_1,t_2)}^2 \geq bm_2) \times \underline{P}(L_{t_2}^3 \geq cm_3) \end{aligned} \tag{15}$$

and the corresponding NPI upper probability is

$$\begin{aligned} &\overline{P}(L_{t_1}^1 \geq am_1, L_{(t_1,t_2)}^2 \geq bm_2, L_{t_2}^3 \geq cm_3) = \\ &\overline{P}(L_{t_1}^1 \geq am_1) \times \overline{P}(L_{(t_1,t_2)}^2 \geq bm_2) \times \overline{P}(L_{t_2}^3 \geq cm_3) \end{aligned} \tag{16}$$

For $I_i^1 = (x_{i-1}^1, x_i^1), i = 1, \ldots, n_1+1$, and $t_1 \in I_{i_{t_1}}^1 = (x_{i_{t_1}-1}^1, x_{i_{t_1}}^1)$, where $i_{t_1} \in \{2, 3, \ldots, n_1\}$ is defined as that interval $I_{i_{t_1}}^1$ which contains t_1, the NPI lower and upper probabilities for the event $L_{t_1}^1 \geq am_1$ are [10, 21]

$$\underline{P}(L_{t_1}^1 \geq am_1) = \underline{P}(X_{\lceil am_1 \rceil}^1 \leq t_1) = \sum_{i=1}^{i_{t_1}-1} P(X_{\lceil am_1 \rceil}^1 \in I_i^1) \tag{17}$$

$$\overline{P}(L^1_{t_1} \geq am_1) = \overline{P}(X^1_{\lceil am_1 \rceil} \leq t_1) = \sum_{i=1}^{i_{t_1}} P(X^1_{\lceil am_1 \rceil} \in I^1_i) \tag{18}$$

For $i_{t_1} = 1$, these NPI lower and upper probabilities are $\underline{P}(L^1_{t_1} \geq am_1) = 0$ and $\overline{P}(L^1_{t_1} \geq am_1) = P(X^1_{\lceil am_1 \rceil} \in I^1_1)$, and for $i_{t_1} = n_1 + 1$, they are $\underline{P}(L^1_{t_1} \geq am_1) = 1 - P(X^1_{\lceil am_1 \rceil} \in I^1_{n_1+1})$ and $\overline{P}(L^1_{t_1} \geq am_1) = 1$.

For $I^2_j = (x^2_{j-1}, x^2_j)$ with $j = 1, \ldots, n_2 + 1$ and $t_1 \in I^2_{j_{t_1}} = (x^2_{j_{t_1}-1}, x^2_{j_{t_1}})$, and $t_2 \in I^2_{j_{t_2}} = (x^2_{j_{t_2}-1}, x^2_{j_{t_2}})$, with $j_{t_1} \in \{1, 2, \ldots, n_1+1\}$ and $j_{t_2} \in \{1, 2, \ldots, n_2+1\}$, with $t_2 \geq t_1$ so $j_{t_2} \geq j_{t_1}$, the NPI lower and upper probabilities for the event $L^2_{(t_1,t_2)} \geq bm_2$ are [10, 21]

$$\underline{P}(L^2_{(t_1,t_2)} \geq bm_2) = P(L^2_{(x^2_{j_{t_1}}, x^2_{j_{t_2}-1})} \geq bm_2) \tag{19}$$

$$\overline{P}(L^2_{(t_1,t_2)} \geq bm_2) = P(L^2_{(x^2_{j_{t_1}-1}, x^2_{j_{t_2}})} \geq bm_2) \tag{20}$$

For $j_{t_1} = 1$ and $j_{t_2} = 2$, these NPI lower and upper probabiliites are $\underline{P}(L^2_{(t_1,t_2)} \geq bm_2) = 0$ and $\overline{P}(L^2_{(t_1,t_2)} \geq bm_2) = P(L^2_{(-\infty, x^2_{j_{t_2}})} \geq bm_2)$.

For $I^3_l = (x^3_{l-1}, x^3_l), l = 1, \ldots, n_3+1$, and $t_2 \in I^3_{l_{t_2}} = (x^3_{l_{t_2}-1}, x^3_{l_{t_2}})$ where $l_{t_2} \in \{1, 2, \ldots, n_3\}$, the NPI lower and upper probabilities for the event $L^3_{t_2} \geq cm_3$ are [10, 21]

$$\underline{P}(L^3_{t_2} \geq cm_3) = \underline{P}(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} > t_2) = \sum_{l=l_{t_2}+1}^{n_3+1} P(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} \in I^3_l) \tag{21}$$

$$\overline{P}(L^3_{t_2} \geq cm_3) = \overline{P}(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} > t_2) = \sum_{l=l_{t_2}}^{n_3+1} P(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} \in I^3_l) \tag{22}$$

where $\lceil cm_3 \rceil$ is the smallest integer greater than or equal cm_3. For $l_{t_2} = 1$, these NPI lower and upper probabilites are $\underline{P}(L^3_{t_2} \geq cm_3) = 1 - P(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} \in I^3_1)$ and $\overline{P}(L^3_{t_2} \geq cm_3) = 1$, and for $l_{t_2} = n_3 + 1$, $\underline{P}\left(L^3_{t_2} \geq cm_3\right) = 0$ and $\overline{P}\left(L^3_{t_2} \geq cm_3\right) = P\left(X^3_{(m_3 - \lceil cm_3 \rceil + 1)} \in I^3_{n_3+1}\right)$.

We obtain the two optimal thresholds, t_1 and t_2, for the three ordered classes C_1, C_2, and C_3 by maximising either the NPI lower probability in Equation (15) or the NPI upper probability in Equation (16). The optimal values of t_1 and t_2 are found by searching for the pair that maximises the chosen probability within each of the $n_1 + n_2 + n_3 + 1$ intervals determined by the data observations.

4. NPI-Based Tree Algorithms for Noisy Data

Alrasheedi [14] and Alrasheedi et al. [13] introduced two classification algorithms for building classification trees based on the Nonparametric Predictive Inference (NPI) approach for selecting optimal thresholds [13, 14, 21]. The NPI_2-Tree algorithm is designed for binary classification problems (two classes), while the NPI_3-Tree algorithm handles classification with three ordered classes. In this study, we evaluate the performance of these algorithms on datasets with class noise to assess their robustness.

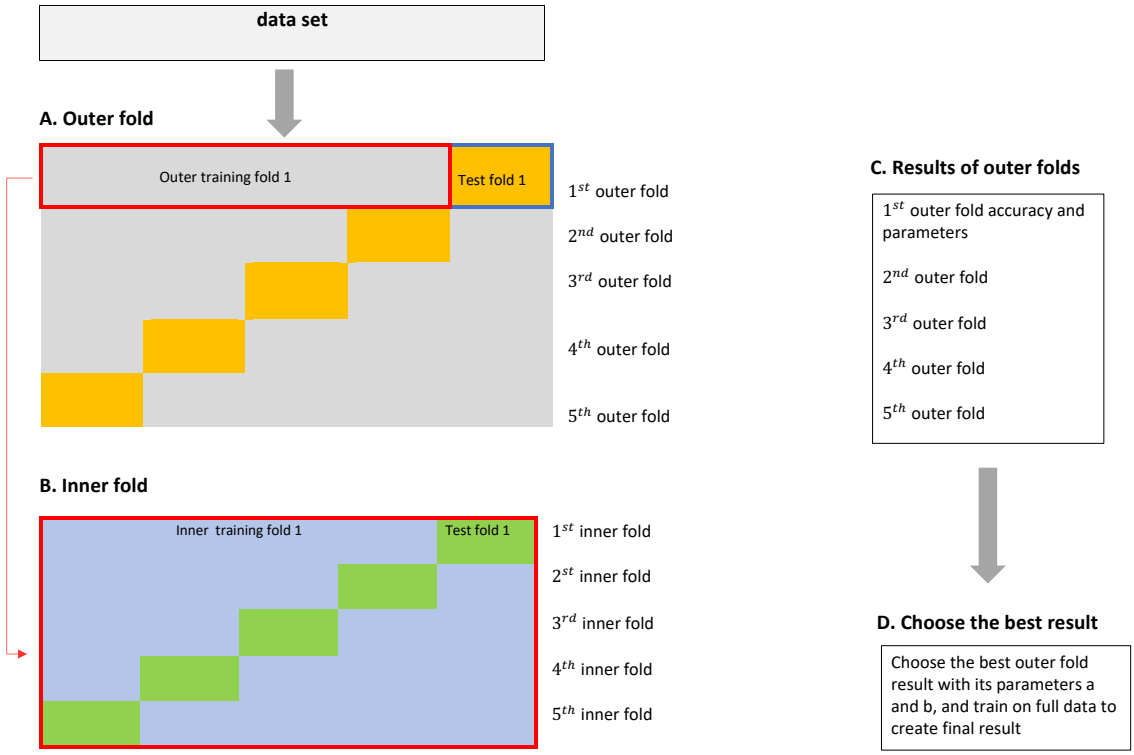


Figure 1: A diagram of a two-stage 5-fold cross-validation procedure to find the optimal values of the target proportions.

A key feature of these algorithms is their data-driven determination of target proportions a, b, and c, which represent the minimum required proportions of correctly classified future observations from each class. Rather than fixing these proportions a priori, the algorithms employ a two-stage k-fold cross-validation procedure (referred to as double 5-fold cross-validation) to select optimal values. The outer loop evaluates the overall classification performance, while the inner loop searches for values of a, b, and c that maximise classification accuracy. This procedure allows the algorithms to adapt to the characteristics of each dataset.

Figure 1 illustrates the proposed procedure, showing both stages of the 5-fold cross-validation. The outer loop validates the method for selecting parameters a and b for the NPI_2-Tree algorithm, or a, b, and c for the NPI_3-Tree algorithm. The inner loop optimises these parameters. The procedures for constructing the NPI_2-Tree and NPI_3-Tree classification trees are detailed in Algorithm 1 and Algorithm 2, given in Appendix A and Appendix B, respectively. For further details, we refer the reader to [13, 14].

Data set	N	Attr	Pro of class 1	Pro of class 2
Breast Cancer	116	9	0.45	0.55
Blood Transfusion	748	3	0.76	0.24
Liver Patients	583	9	0.28	0.72
Haberman's Survival	306	3	0.73	0.27
Cryotherapy	90	5	0.53	0.47
QSAR Biodeg	1055	14	0.66	0.34

Table 1: A brief description of the data sets used with the NPI_2-Tree algorithm.

Data set	N	Attr	Pro of class 1	Pro of class 2	Pro of class 3
Iris	150	4	0.33	0.33	0.33
Seeds	210	7	0.33	0.33	0.33
Wine	178	3	0.34	0.39	0.27
CMC	1473	2	0.42	0.24	0.34
Fitness	8020	10	0.36	0.41	0.23

Table 2: A brief description of the data sets used with the NPI_3-Tree algorithm.

5. Experimental analysis

This section examines the performance of the NPI_2-Tree and NPI_3-Tree algorithms compared to the C4.5, CART, NPI-M, and IDM1 algorithms on datasets with varying levels of noise. Six datasets for two-class problems were used for the NPI_2-Tree algorithm, and five datasets for three-class problems were used for the NPI_3-Tree algorithm, all taken from the UCI Machine Learning Repository [23]; other classification algorithms were applied to the same datasets accordingly. The characteristics of these datasets are summarised in Tables 1 and 2. Column 'N' lists the number of observations, 'Attr' gives the number of attribute variables, and 'Pro of class i' indicates the proportion of data in class i. Further details about these datasets are available in [23].

All experiments were carried out using the R statistical software [35]. The `RWeka` package [29, 38] was used for C4.5, `rpart` [37] for CART, and `imptree` [26] for both NPI-M and IDM1. Preprocessing steps included replacing missing values for continuous attributes with the mean and adding a small value to break tied observations; testing without breaking ties produced similar results. Since NPI-M and IDM1 can handle only categorical attributes, continuous variables in Table 1 were discretised using the `mdlp` package and its `discretization` function, which applies the Fayyad and Irani method [24, 25], selecting thresholds based on average class entropy.

To evaluate the performance of these classification algorithms on noisy data, we introduce noise into the class variables of the datasets. Noise is added because the original datasets may already contain unknown levels of noise, and no assumptions are made about its presence or magnitude; thus, the datasets are treated as noise-free. We add noise to the class variable at levels of 10%, 15%, 30%, and 50%. These noise levels are applied only to the training sets, while the testing sets remain unchanged. Although many studies in the literature add noise only up to 30% [6, 7, 16, 32], we include 50% noise to evaluate the algorithms under high-noise conditions, following [5]. The noise is introduced by randomly selecting x% of the observations in the training set (where x is the desired noise level) and replacing their class labels with a different class chosen uniformly from the remaining classes.

In our experiments, the primary evaluation metric is classification accuracy, defined as the ratio of correctly classified observations to the total number of observations in the test set. In practice, accuracy is calculated using k-fold cross-validation [30]. The dataset is randomly divided into k approximately equal-sized folds. Each fold is used once as a testing set, while the remaining $k-1$ folds are combined as a training set. This process is repeated k times, and the final classification accuracy is obtained by averaging the results across all k folds.

For the NPI_2-Tree and NPI_3-Tree algorithms, the double-5-fold cross-validation procedure described earlier is first used to determine the optimal target proportions with noisy data. Accuracy on the original, noise-free training sets (Noise level 0%) serves as a reference to assess the robustness of the algorithms under different noise levels. A classification algorithm is considered robust if it maintains similar accuracy on both noisy and noise-free datasets [36].

The experimental results are presented in two phases. In Section 5.1, we report the performance of the NPI_2-Tree, C4.5, CART, NPI-M, and IDM1 algorithms on the six datasets with two classes. In Section 5.2, we report the results of the NPI_3-Tree, C4.5, CART, NPI-M, and IDM1 algorithms on the five datasets with three classes. The best results in each table are highlighted in bold.

5.1. Results for the NPI_2-Tree algorithm

Table 3 presents the classification accuracy results of the NPI_2-Tree algorithm and the other classification algorithms for each dataset at noise levels of 0%, 10%, 15%, 30%, and 50%. The table also reports the optimal values of the target proportions a and b for the NPI_2-Tree algorithm. Figure 2 provides a summary of classification accuracy results for all algorithms across all noise levels.

From Table 3, we observe that the performance of all classification algorithms generally decreases as the noise level increases. However, the NPI_2-Tree, NPI-M, and IDM1 algorithms are more robust to noise than C4.5 and CART across most noise levels. As shown in Figure 2, the nearly parallel lines for the NPI_2-Tree, NPI-M, and IDM1 algorithms indicate similar robustness.

For low noise levels (10% and 15%), the classification accuracies are similar, so these results are discussed together. At these levels, the NPI_2-Tree algorithm achieves the highest average accuracy, followed by NPI-M, IDM1, C4.5, and CART. The performance of NPI_2-Tree, NPI-M, and IDM1 remains close to their performance on the original datasets (Noise level 0%). For the Cryotherapy dataset, the NPI_2-Tree algorithm's accuracy slightly increases at 10% and 15% noise but decreases at 30% and 50% noise, likely due to randomness in small datasets with low noise. Similarly, the IDM1 algorithm shows a small increase at 10% noise for this dataset. Such behavior, where some algorithms perform slightly better with low noise, has been observed in previous studies [7, 31, 32].

At medium noise levels (30%), Table 3 shows that the NPI_2-Tree algorithm achieves the highest average classification accuracy on two datasets. For the Breast Cancer and Haberman's Survival datasets, the IDM1 algorithm slightly outperforms the others, while for the Cryotherapy and Liver Patients datasets, NPI-M performs slightly better. For the Blood Transfusion and QSAR Biodeg datasets, the NPI_2-Tree algorithm consistently outperforms

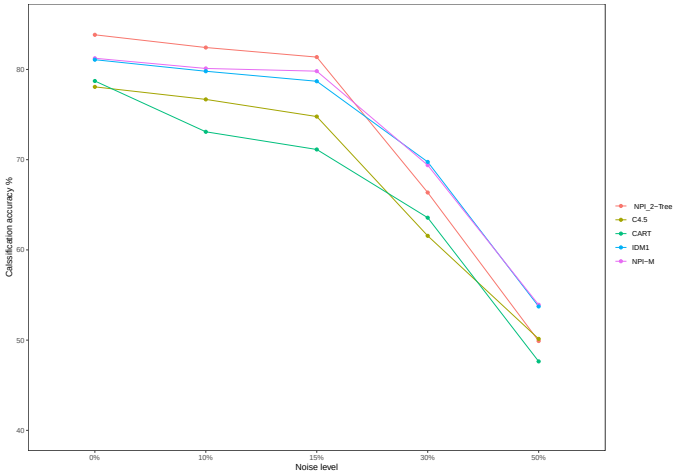


Figure 2: Classification algorithms performance with different levels of added noise, two classes case.

the other algorithms across all noise levels up to 30%, suggesting greater robustness for large datasets with minimal class overlap.

At high noise levels (50%), all algorithms show a notable drop in performance. While 30% noise is generally sufficient to assess robustness [6, 7, 16, 31, 32], we include 50% to explore algorithm behavior under extreme noise, following [5]. In binary classification with 50% noise, the dataset may contain little useful information, so differences between classifiers may largely reflect randomness. At this noise level, the NPI-M and IDM1 algorithms achieve the highest accuracies, while CART performs the worst.

5.2. Results for the NPI$_3$-Tree algorithm

Table 4 presents the classification accuracy results of all algorithms on datasets with class noise levels of 0%, 10%, 15%, 30%, and 50%. The table also shows the optimal target proportions a, b, and c for the NPI$_3$-Tree algorithm. Figure 3 summarises the average classification accuracy of each algorithm across all noise levels.

From Table 4, the NPI$_3$-Tree algorithm generally achieves the highest average accuracy for most noise levels, outperforming C4.5 and CART at all levels. For the original datasets (0%) and low noise levels (10% and 15%), NPI$_3$-Tree performs slightly better than the other algorithms, followed by NPI-M, IDM1, C4.5, and CART. The optimal target proportions a, b, and c are minimally affected by these low noise levels.

For the CMC dataset, all algorithms show lower performance across all noise levels, including the noise-free dataset. This may be due to the large size of the dataset (1474 observations, 2 attributes) combined with substantial class overlap, which makes classification more challenging. Similar observations were reported by Abellán et al. [9] and Mantas et al. [31]. For this dataset, the optimal target proportions a, b, and c are also relatively low.

At medium noise levels (30%), the IDM1 algorithm slightly outperforms the others, although its performance is very close to that of NPI$_3$-Tree and NPI-M. This aligns with findings by Mantas and Abellán [32], who observed IDM1 achieving the highest average accuracy for several datasets with 30% random noise. At this noise level, the NPI$_3$-Tree algorithm achieves the best accuracy on the Fitness dataset, the largest dataset in this

Data set	a	b	NPI_2-Tree	C4.5	CART	NPI-M	IDM1
Noise level 0%							
Breast Cancer	0.57	0.73	87.10	86.89	86.90	87.65	**87.86**
Cryotherapy	0.56	0.50	79.38	80.11	**83.48**	81.45	80.18
Blood Transfusion	0.79	0.64	**89.48**	75.43	74.76	79.56	79.56
Haberman's Survival	0.84	0.42	75.19	75.62	73.18	**76.39**	**76.39**
Liver Patients	0.32	0.65	**80.70**	77.25	76.43	80.28	80.28
QSAR Biodeg	0.65	0.82	**91.22**	73.13	77.86	82.16	82.16
Average	-	-	**83.84**	78.07	78.72	81.24	81.08
Noise level 10%							
Breast Cancer	0.82	0.48	85.13	83.32	70.35	**86.80**	84.12
Cryotherapy	0.52	0.65	80.10	79.56	78.38	80.23	**80.73**
Blood Transfusion	0.61	0.48	**87.32**	73.51	71.90	76.11	77.49
Haberman's Survival	0.87	0.61	74.37	**75.76**	73.16	75.32	73.81
Liver Patients	0.52	0.79	77.56	75.82	69.41	**79.12**	78.39
QSAR Biodeg	0.79	0.86	**90.14**	72.13	75.38	83.12	84.19
Average	-	-	**82.43**	76.68	73.09	80.11	79.80
Noise level 15%							
Breast Cancer	0.54	0.61	84.19	81.94	70.10	**85.91**	84.12
Cryotherapy	0.48	0.57	**80.54**	76.93	76.14	79.30	79.65
Blood Transfusion	0.44	0.49	**85.29**	72.19	68.81	78.15	75.11
Haberman's Survival	0.71	0.69	74.14	73.38	71.84	**74.59**	72.95
Liver Patients	0.43	0.18	75.33	73.16	66.25	**79.63**	78.06
QSAR Biodeg	0.82	0.54	**88.75**	71.13	73.68	81.93	82.27
Average	-	-	**81.37**	74.78	71.13	79.81	78.69
Noise level 30%							
Breast Cancer	0.43	0.27	70.28	60.74	65.82	**73.15**	70.53
Cryotherapy	0.31	0.46	70.39	65.57	71.28	72.93	**74.33**
Blood Transfusion	0.52	0.39	**71.81**	64.32	65.16	65.60	67.81
Haberman's Survival	0.62	0.46	55.22	63.69	58.96	**67.15**	64.55
Liver Patients	0.31	0.19	57.11	54.82	56.13	70.13	**74.18**
QSAR Biodeg	0.58	0.63	**73.35**	60.41	64.10	67.41	67.12
Average	-	-	66.36	61.56	63.57	69.39	**69.75**
Noise level 50%							
Breast Cancer	0.19	0.24	51.21	47.15	49.26	54.54	**56.30**
Cryotherapy	0.24	0.17	45.14	46.68	53.28	48.68	**51.98**
Blood Transfusion	0.43	0.49	58.91	52.35	48.40	**61.39**	53.71
Haberman's Survival	0.53	0.36	47.12	**58.16**	42.24	46.14	46.59
Liver Patients	0.30	0.72	43.80	41.33	40.71	55.43	**57.42**
QSAR Biodeg	0.46	0.61	53.22	55.18	51.97	**57.38**	56.43
Average	-	-	49.90	50.14	47.64	**53.92**	53.73

Table 3: Classification accuracy results obtained by the NPI_2-Tree, C4.5, CART, NPI-M and IDM1 algorithms, with different levels of added noise.

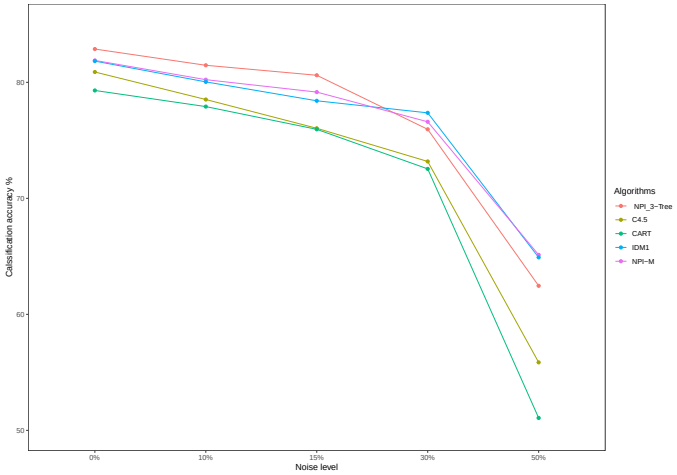


Figure 3: Classification algorithms performance with different levels of added noise, three classes case.

study. This pattern mirrors results from Section 5.1, where NPI_2-Tree also outperformed other algorithms on large datasets at 30% noise. These results suggest that NPI_3-Tree may perform particularly well on large datasets with medium noise. Future research could explore the impact of class-specific noise on threshold selection in NPI_3-Tree, e.g., by adding noise to one class at a time and evaluating the algorithm's performance.

At high noise levels (50%), NPI-M and IDM1 outperform the other algorithms, with NPI_3-Tree performing comparably. C4.5 and CART show substantial drops in accuracy at this noise level; for example, their average accuracies fall from 73.18% and 72.54% at 30% noise to 55.86% and 51.07% at 50% noise, respectively. Overall, NPI_3-Tree demonstrates strong robustness to class noise, performing best at 0%, 10%, and 15% noise and outperforming classical algorithms across all noise levels. The next section further analyses NPI_3-Tree under different scenarios of adding noise to specific classes.

6. Investigating Class-Dependent Noise Scenarios

In many real-world situations, observations are more likely to be misclassified into a neighbouring class than into a class further away. For example, in medical diagnostics, values close to a threshold are more likely to be misclassified than values further from the threshold. Consider three ordered temperature classes with two thresholds: below 37.5 indicates healthy, 37.5 to 38.5 indicates mild disease, and above 38.5 indicates serious disease. A person with an actual temperature of 37.4 is more likely to be misclassified as mildly diseased than as seriously diseased. From this perspective, it is useful to evaluate the performance of the NPI_3-Tree and other algorithms under such situations, where the probability of misclassification is higher for neighbouring classes and lower for classes further away. This analysis is only applicable to datasets with three ordered classes, as datasets with two classes allow only one type of misclassification. Therefore, this section focuses exclusively on the NPI_3-Tree algorithm.

To assess performance, we define three scenarios with different probabilities for misclassifying observations to neighbouring versus more distant classes. Table 5 presents these scenarios, where p_{ij} denotes the probability of replacing an observation's class label from

Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Noise level 0%								
Iris	0.77	0.75	0.80	94.61	94.22	94.38	**94.69**	94.52
Seeds	0.81	0.79	0.78	**93.43**	89.72	90.42	92.63	92.38
Wine	0.94	0.68	0.87	**96.54**	93.12	91.14	95.19	94.64
CMC	0.43	0.36	0.52	49.96	**50.10**	48.40	49.81	49.81
Fitness	0.53	0.64	0.49	**79.81**	77.31	72.19	77.60	77.82
Average	-	-	-	**82.87**	80.89	79.30	81.98	81.83
Noise level 10%								
Iris	0.81	0.75	0.68	93.60	91.70	92.16	93.33	**93.72**
Seeds	0.93	0.63	0.72	**92.13**	85.20	87.40	90.60	90.23
Wine	0.91	0.87	0.83	**94.28**	92.68	91.78	92.70	91.11
CMC	0.55	0.32	0.62	48.12	46.92	**48.30**	47.45	47.83
Fitness	0.68	0.45	0.47	**79.22**	76.12	69.94	77.10	77.31
Average	-	-	-	**81.47**	78.52	77.91	80.23	80.04
Noise level 15%								
Iris	0.79	0.66	0.72	**93.49**	86.35	85.29	91.64	91.87
Seeds	0.88	0.70	0.63	**92.22**	83.49	82.75	89.10	86.76
Wine	0.92	0.81	0.74	**92.10**	90.28	89.44	90.28	88.40
CMC	0.43	0.51	0.48	46.83	44.19	47.98	47.70	**48.08**
Fitness	0.59	0.43	0.55	78.43	75.91	74.28	77.12	76.94
Average	-	-	-	**80.61**	76.04	75.95	79.16	78.41
Noise level 30%								
Iris	0.74	0.59	0.60	88.19	84.14	83.14	89.93	**90.85**
Seeds	0.81	0.58	0.74	83.36	76.80	77.78	**82.42**	82.19
Wine	0.72	0.69	0.43	**91.13**	89.23	85.84	87.18	89.16
CMC	0.21	0.28	0.40	41.20	45.43	44.31	47.67	**49.55**
Fitness	0.37	0.56	0.31	**75.88**	70.34	71.67	75.82	75.11
Average	-	-	-	75.95	73.18	72.54	76.60	**77.37**
Noise level 50%								
Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Iris	0.63	0.48	0.81	**74.13**	61.40	59.31	69.87	69.13
Seeds	0.59	0.36	0.68	71.23	59.32	57.80	70.15	**72.67**
Wine	0.48	0.45	0.27	58.48	67.16	49.43	**73.84**	73.34
CMC	0.42	0.31	0.23	**53.12**	33.52	30.92	52.13	50.17
Fitness	0.35	0.45	0.29	55.36	58.40	57.91	**59.67**	59.24
Average	-	-	-	62.46	55.86	51.07	**65.13**	64.91

Table 4: Classification accuracy results obtained by the NPI_3-Tree, C4.5, CART, NPI-M and IDM1 algorithms, with different levels of added noise.

# Scenario	p_{ij} if $\lvert i-j\rvert = 1$	p_{ij} if $\lvert i-j\rvert = 2$
1	0.6	0.4
2	0.8	0.2
3	1	0.0

Table 5: Different scenarios of adding noise to the class variable.

class i ($i = 1, 2, 3$) to class j ($j = 1, 2, 3$), excluding the actual class label ($i \neq j$). Note that the middle class has two neighbouring classes and is misclassified to each with equal probability (i.e. with probability 0.5). To introduce noise according to these scenarios, we randomly select x% of the observations in the training set (where x is the desired noise level) and replace their class labels according to the probabilities in Table 5.

We tested the NPI$_3$-Tree algorithm on the five datasets listed in Table 2, adding noise to the training sets at levels of 10%, 15%, 30%, and 50%. The performance was compared to C4.5, CART, NPI-M, and IDM1 using the same procedure as in Section 5.2. Tables 6, 7, and 8 present the average classification accuracies for each dataset, scenario, and noise level. Figure 4 provides graphical representations of the results. All results were obtained using a 10-fold cross-validation procedure, and the best results are highlighted in bold.

For Scenario 1, the probability of misclassification to neighbouring classes is 0.60, and to more distant classes is 0.40. The results in Table 6 show that all algorithms perform similarly to their performances under random noise (Section 5.2). In Scenario 2, the probability of misclassification to neighbouring classes is increased to 0.80, and to distant classes is reduced to 0.20. Table 7 shows that at 10% and 15% noise, NPI$_3$-Tree slightly outperforms the other algorithms. At 30% noise, NPI$_3$-Tree achieves the highest accuracy in four out of five datasets and performs similarly to the imprecise algorithms on the CMC dataset, with an average accuracy of 75.64%, followed by NPI-M, IDM1, CART, and C4.5. For 50% noise, NPI-M and IDM1 outperform C4.5 and CART but are comparable to NPI$_3$-Tree. In Scenario 3, the probability of misclassification to neighbouring classes is 1.0, and to distant classes is 0. Table 8 shows that for 10% and 15% noise, NPI$_3$-Tree performs slightly better than the other algorithms, though all algorithms are fairly close. At 30% noise, NPI$_3$-Tree again achieves the highest average accuracy, and at 50% noise, all algorithms perform poorly, with IDM1 slightly outperforming the rest.

Overall, these scenarios illustrate how the position of noisy observations affects classification performance. Across all three scenarios, NPI$_3$-Tree generally performs well and often outperforms the other algorithms at low and medium noise levels. At 10% and 15% noise, NPI$_3$-Tree consistently achieves the best performance. At 30% noise, it remains superior in most scenarios, except in Scenario 1, where its performance is similar to NPI-M and IDM1. At 50% noise, NPI$_3$-Tree outperforms C4.5 and CART, while NPI-M and IDM1 perform slightly better.

7. Conclusions

This paper has presented applications of the NPI$_2$-Tree and NPI$_3$-Tree algorithms to the problem of classification with noisy data. We conducted an experimental analysis using multiple datasets and various levels of random noise to evaluate the performance of

Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
				Noise level 10%				
Iris	0.92	0.82	0.64	93.13	93.98	92.75	**93.52**	**93.52**
Seeds	0.87	0.55	0.74	**92.95**	89.63	87.42	91.32	91.58
Wine	0.74	0.63	0.96	**92.87**	91.50	91.18	91.64	91.43
CMC	0.52	0.31	0.64	49.30	48.63	**49.50**	48.71	48.19
Fitness	0.58	0.42	0.60	**79.38**	77.32	70.82	78.21	78.10
Average	-	-	-	**81.52**	80.21	78.33	80.68	80.56
				Noise level 15%				
Iris	0.80	0.78	0.71	**93.24**	89.74	87.22	91.32	91.15
Seeds	0.83	0.76	0.71	**92.28**	83.51	82.29	90.17	87.16
Wine	0.87	0.84	0.76	92.20	91.57	91.61	**92.84**	92.76
CMC	0.49	0.50	0.55	50.41	48.19	50.98	50.70	**51.18**
Fitness	0.50	0.45	0.66	**78.21**	74.63	73.90	77.19	77.23
Average	-	-	-	**81.26**	77.47	77.20	80.44	79.89
				Noise level 30%				
Iris	0.70	0.72	0.69	85.21	83.10	84.80	**89.87**	89.12
Seeds	0.81	0.58	0.74	**84.36**	80.80	79.21	83.50	83.85
Wine	0.72	0.69	0.43	**86.48**	83.23	85.84	85.54	84.46
CMC	0.21	0.28	0.40	42.63	44.21	44.61	**47.67**	47.53
Fitness	0.37	0.56	0.31	73.14	70.34	71.67	**73.38**	73.17
Average	-	-	-	74.36	72.34	73.22	**75.99**	75.62
				Noise level 50%				
Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Iris	0.45	0.30	0.61	**77.62**	72.19	57.50	75.87	74.13
Seeds	0.59	0.36	0.68	**75.63**	67.18	69.98	73.65	73.11
Wine	0.58	0.49	0.39	61.12	64.53	56.36	69.21	**69.84**
CMC	0.46	0.51	0.33	39.75	47.52	45.92	50.16	**51.98**
Fitness	0.38	0.65	0.26	54.65	53.12	51.98	**58.17**	59.14
Average	-	-	-	62.09	60.90	56.34	65.41	**65.64**

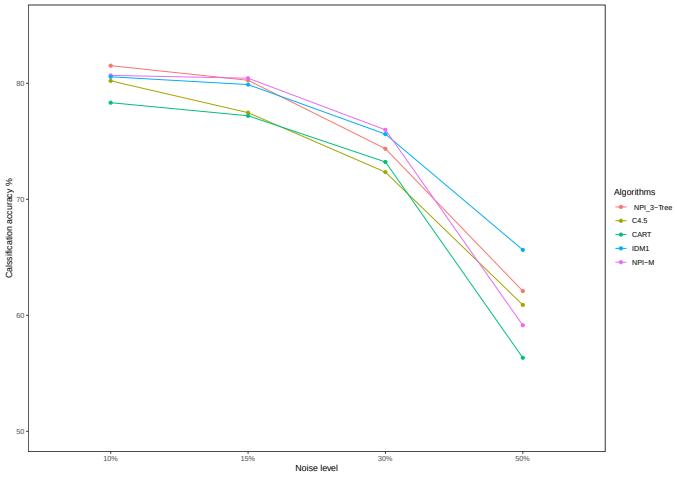
Table 6: Accuracy results of classification algorithms, first scenario.

Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Noise level 10%								
Iris	0.88	0.76	0.72	93.26	91.84	90.37	**93.58**	93.19
Seeds	0.93	0.61	0.80	**93.46**	85.29	87.40	91.12	90.23
Wine	0.85	0.67	0.92	92.16	92.75	91.88	92.37	**92.81**
CMC	0.52	0.30	0.64	49.34	49.34	48.50	**50.21**	**50.21**
Fitness	0.68	0.45	0.47	**79.76**	76.32	67.12	78.55	79.43
Average	-	-	-	**81.59**	79.10	77.05	81.16	81.17
Noise level 15%								
Iris	0.80	0.78	0.71	93.26	89.55	90.29	92.54	**92.87**
Seeds	0.88	0.70	0.63	**92.56**	86.49	86.75	89.10	90.76
Wine	0.92	0.81	0.74	**92.64**	90.28	89.44	92.81	92.81
CMC	0.43	0.51	0.48	50.19	**51.39**	47.98	51.33	51.21
Fitness	0.59	0.43	0.58	78.43	74.91	69.28	77.12	**79.14**
Average	-	-	-	**81.88**	78.10	76.74	80.58	81.35
Noise level 30%								
Iris	0.83	0.79	0.74	**86.58**	71.90	76.23	86.34	88.24
Seeds	0.85	0.53	0.78	**83.42**	78.85	82.21	82.93	83.16
Wine	0.70	0.60	0.79	**89.38**	80.23	82.84	82.63	**83.22**
CMC	0.38	0.45	0.44	45.51	44.21	41.61	**46.67**	46.55
Fitness	0.57	0.49	0.37	**73.33**	70.34	70.67	71.32	69.11
Average	-	-	-	**75.64**	69.10	70.71	73.97	73.85
Noise level 50%								
Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Iris	0.45	0.30	0.61	59.50	49.88	52.27	64.21	**65.47**
Seeds	0.59	0.36	0.68	70.98	65.18	64.98	72.65	**73.14**
Wine	0.58	0.49	0.39	63.79	65.73	59.12	**73.81**	69.34
CMC	0.46	0.51	0.33	47.98	40.19	46.92	50.76	**51.98**
Fitness	0.38	0.65	0.26	50.40	60.86	61.49	**63.17**	58.32
Average	-	-	-	62.40	56.36	56.95	**64.92**	63.65

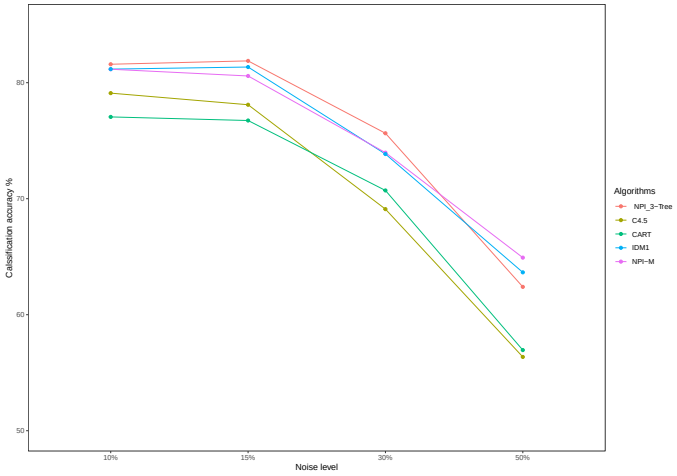
Table 7: Accuracy results of classification algorithms, second scenario.

Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Noise level 10%								
Iris	0.91	0.72	0.68	92.19	91.80	90.73	93.22	**93.34**
Seeds	0.86	0.64	0.84	**92.18**	86.11	89.23	91.18	90.17
Wine	0.85	0.71	0.84	**91.50**	90.78	90.92	91.33	89.41
CMC	0.56	0.32	0.69	50.19	49.63	48.50	48.36	**50.23**
Fitness	0.63	0.47	0.38	**79.64**	76.39	67.32	77.54	78.30
Average	-	-	-	**80.86**	78.94	77.34	80.32	80.29
Noise level 15%								
Iris	0.76	0.68	0.75	91.90	85.32	84.29	92.99	**93.10**
Seeds	0.81	0.56	0.60	**89.37**	84.49	83.75	87.12	86.77
Wine	0.90	0.69	0.73	**90.43**	88.46	89.61	90.32	89.70
CMC	0.61	0.39	0.42	50.77	48.92	46.67	51.39	**52.68**
Fitness	0.58	0.37	0.65	**80.10**	75.88	74.97	79.73	78.94
Average	-	-	-	**80.51**	76.61	75.85	80.31	80.23
Noise level 30%								
Iris	0.69	0.43	0.59	80.24	78.96	78.21	80.13	**80.67**
Seeds	0.83	0.51	0.63	**79.32**	76.19	76.27	78.43	78.98
Wine	0.66	0.49	0.50	78.62	76.25	76.12	79.28	**79.72**
CMC	0.42	0.29	0.61	**49.43**	47.21	48.69	46.14	47.50
Fitness	0.40	0.37	0.63	55.41	**55.54**	53.67	54.87	52.15
Average	-	-	-	**68.60**	66.83	66.59	67.77	67.80
Noise level 50%								
Data set	a	b	c	NPI_3-Tree	C4.5	CART	NPI-M	IDM1
Iris	0.32	0.58	0.43	**63.19**	62.40	61.31	62.87	62.14
Seeds	0.49	0.18	0.37	65.25	65.46	65.98	66.65	**66.77**
Wine	0.33	0.23	0.59	55.32	56.52	56.89	61.16	**58.24**
CMC	0.21	0.09	0.56	43.21	45.13	47.91	45.68	**48.33**
Fitness	0.32	0.28	0.41	49.15	45.17	47.68	**53.39**	52.58
Average	-	-	-	56.89	54.93	55.75	57.95	**57.61**

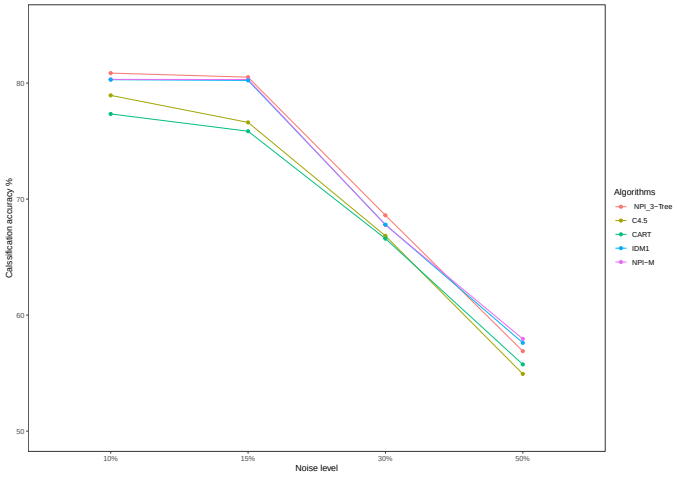
Table 8: Accuracy results of classification algorithms, third scenario.



(a) First scenario



(b) Second scenario



(c) Third scenario

Figure 4: Performance of the classification algorithms for different scenarios of the noise adding process.

these algorithms in the presence of class noise. Their performance was compared with the C4.5, CART, NPI-M, and IDM1 algorithms. The results demonstrate that the NPI_2-Tree and NPI_3-Tree algorithms perform consistently well and exhibit robustness to class noise. For most levels of randomly added noise, these algorithms slightly outperformed the other classification algorithms considered.

Furthermore, we investigated the performance of the NPI_3-Tree algorithm under different noise scenarios specifically designed for three ordered classes, where misclassification probabilities were higher for neighbouring classes and lower for classes further apart. Across these scenarios and various noise levels, the NPI_3-Tree algorithm generally maintained superior performance compared with the other algorithms, highlighting its robustness in more structured misclassification settings.

In this study, we focused exclusively on class noise, as it is prevalent in many real-world datasets and tends to have a more pronounced effect on classification performance than attribute noise. However, the impact of attribute noise on the NPI_2-Tree and NPI_3-Tree algorithms remains an open question and represents an interesting direction for future research. Additionally, investigating the combined effect of noise in both class and attribute variables may provide further insights into the resilience of these algorithms under more complex data imperfections.

In addition, although recent studies have proposed alternative classification approaches, such as improved fuzzy and weighted k-nearest neighbour methods, to enhance robustness under noisy and imbalanced conditions, a direct comparison between the NPI-based tree algorithms and such methods in the presence of noisy data has not been conducted in this work. A systematic comparative study including these, and possibly other, robust classification methods constitutes an important topic for future research. Such investigations would contribute to a broader understanding of the relative strengths, limitations, and practical applicability of the NPI-based tree algorithms in real-world noisy environments.

Overall, the results of this study suggest that the NPI_2-Tree and NPI_3-Tree algorithms are promising tools for classification tasks in noisy datasets, providing both accuracy and robustness, particularly when compared with classical and imprecise-probability-based classification methods.

Acknowledgements

Masad Alrasheedi would like to thank Taibah University and the Saudi government for their invaluable support, which made it possible for him to complete his PhD studies. The authors also wish to thank the reviewer for their constructive feedback.

Disclosure Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Funding

No funding was received.

Appendix A. NPI$_2$-Tree algorithm

Algorithm 1 Pseudocode NPI$_2$-Tree algorithm

1. Input:($\mathcal{D}$, C, Ω)
   $\mathcal{D}$: Data set
   C: Binary class variable $C = \{C_1, C_2\}$
   Ω: Set of continuous attributes $\Omega = \{X_1, \ldots, X_f\}$
2. Procedure NPI$_2$-Tree($\mathcal{D}$, C, Ω)
3. Create a Root node for the tree
4. If all observations in $\mathcal{D}$ have the same class C, then
5. Return the single-node tree with class C
6. If Ω is empty (i.e. there are no attributes available), then
7. Return the single-node tree with most common class C in $\mathcal{D}$
8. Otherwise
9. The data set $\mathcal{D}$ is divided into two subsets:
   S: training set
   T: testing set
10. Select the values of a, b and m_i for $i = 1, 2$
    Make the initial values of a and b equal to the class proportion in S,
    i.e. make $a = \frac{n_1}{n}$ and $b = \frac{n_2}{n}$
    Make the values of m_i equal to the number of observations in class C_i in T
11. For each attribute, X_i in Ω do
    Find the optimal threshold values that maximise the NPI lower probability, given in Equation (7)
    Compute the IGR value, given in Equation (**??**)
12. Choose attribute X from Ω, with the highest IGR value
13. Assign the attribute X for the Root node
14. Add a branch below the Root node, corresponding to $X \leq t$ and $X > t$
15. Let S_i, for $i = 1, 2$ be the subset of S that has $X \leq t$ and $X > t$, respectively
16. If S_i, for $i = 1, 2$ is empty (one of them), then
17. Add a leaf node below the branch with the most common class in S
18. Check the stopping criteria mentioned above
19. Else
20. Add the subset created by NPI$_2$-Tree (S_i, C, $\Omega - \{X\}$)
21. Return Root

Appendix B. NPI$_3$-Tree algorithm

Algorithm 2 Pseudocode NPI$_3$-Tree algorithm

1. Input:($\mathcal{S}$, C, Ω)
2. $\mathcal{S}$: Training data set
3. C: A class variable $C = \{C_1, C_2, C_3\}$
4. Ω: Set of continuous attributes $\Omega = \{X_1, \ldots, X_f\}$
5. Procedure NPI$_3$-Tree($\mathcal{S}$, C, Ω)
6. Create a Root node for the tree
7. if all observations in $\mathcal{S}$ have the same class C, **then**
8. Return the single-node tree with class C
9. if Ω is empty (i.e. there are no attributes available), then
10. Return the single-node tree with most common class C in $\mathcal{S}$
11. Otherwise
12. Select the values of a, b, c and m_i for $i = 1, 2, 3$
13. Make the initial values of a, b and c equal to the class proportion in S,
14. i.e. make $a = \frac{n_1}{n}$, $b = \frac{n_2}{n}$ and $c = \frac{n_3}{n}$
15. Make the values of m_i equal to the number of observations in class C_i in S,
16. i.e. make $m_1 = n_1,\ m_2 = n_2$ and $m_3 = n_3$
17. for each attribute, X_i in Ω, **do**
18. Find the threshold values t_1 and t_2 that maximise the NPI lower probability, given in Equation (15)
19. Compute the IGR value using Equation (**??**)
20. Choose attribute variable X from Ω, with the highest IGR value
21. Assign the attribute X for the Root node
22. Add a branch below Root, corresponding to $X \leq t_1$, $t_1 < X \leq t_2$ and $X > t_2$,
23. Let S_i, for $i = 1, 2, 3$, be the subset of S that has $X \leq t_1$, $t_1 < X \leq t_2$ and $X > t_2$, respectively
24. if S_i is not empty, **then**
25. Add the subset created by NPI$_3$-Tree (S_i, C, $\Omega - \{X\}$)
26. return Root

References

[1] Abdalla H. and Amer A. (2025). Enhancing data classification using locally informed weighted k-nearest neighbor algorithm. *Expert Systems with Applications*, 276, 126942.

[2] Abdalla, H., Amer, A. and Nassef, M. (2025). New fuzzy K-nearest neighbor algorithms for classification performance improvement. *Future Generation Computer Systems*, p. 108139.

[3] Abellán J. (2006). Uncertainty measures on probability intervals from the imprecise Dirichlet model. *International Journal of General Systems*, 35, 509–528.

[4] Abellán J. and Moral S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18, 1215–1225.

[5] Abellán, J. (2013). An application of non-parametric predictive inference on multi-class classification high-level-noise problems. *Expert Systems with Applications*, 40, 4585–4592.

[6] Abellán, J. and Masegosa, A. (2010). Bagging decision trees on data sets with classification noise. In *International Symposium on Foundations of Information and Knowledge Systems*, pp. 248–265. Springer.

[7] Abellán, J. and Masegosa, A. (2012). Bagging schemes on the presence of class noise in classification. *Expert Systems with Applications*, 39, 6827–6837.

[8] Abellán, J., Baker, R. and Coolen, F.P.A. (2011). Maximising entropy on the nonparametric predictive inference model for multinomial data. *European Journal of Operational Research*, 212, 112–122.

[9] Abellán, J., Baker, R., Coolen, F.P.A., Crossman, R. and Masegosa, R. (2014). Classification with decision trees from a nonparametric predictive inference perspective. *Computational Statistics and Data Analysis*, 71, 789–802.

[10] Alabdulhadi, M. (2018). *Nonparametric predictive inference for diagnostic test thresholds*. Ph.D. thesis, Durham University.

[11] Alharbi, A. A.,Coolen, F.P.A and Coolen-Maturi, T. (2026). Direct nonparametric predictive inference classification trees. *Journal of Applied Statistics*, pp. 1–27.

[12] Alqifari, H. (2017). *Nonparametric predictive inference for future order statistics*. Ph.D. thesis, Durham University.

[13] Alrasheedi M.A., Coolen-Maturi T. and Coolen F.P.A. (2025). Optimal Thresholds for Classification Trees using Nonparametric Predictive Inference. *Communications in Statistics – Theory and Methods*. To appear.

[14] Alrasheedi, M. (2023). *Optimal Thresholds for Classification Trees using Nonparametric Predictive Inference*. Ph.D. thesis, Durham University.

[15] Amer, A., Ravana, S. and Habeeb, R. (2025). On enhancing data classification using local mean-based fuzzy K-nearest neighbor algorithms. *Advances in Data Analysis and Classification*, pp. 1–46.

[16] Amri, N. (2009). *Classification techniques for noisy and imbalanced data*. Ph.D. thesis, Florida Atlantic University.

[17] Augustin T. and Coolen F.P.A. (2004). Nonparametric predictive inference and interval probability. *Journal of Statistical Planning and Inference*, 124, 251–272.

[18] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth International Group, Belmont.

[19] Coolen F.P.A., Coolen-Maturi T. and Alqifari H. (2018). Nonparametric predictive inference for future order statistics. *Communications in Statistics: Theory and Methods*, 47, 2527–2548.

[20] Coolen F.P.A. and Maturi T. (2010). Nonparametric predictive inference for order statistics of future observations. *In: Combining Soft Computing and Statistical Methods in Data Analysis*, pp. 97–104.

[21] Coolen-Maturi T., Coolen F.P.A. and Alabdulhadi M. (2020). Nonparametric predictive inference for diagnostic test thresholds. *Communications in Statistics-Theory and Methods*, 49, 697–725.

[22] De Finetti, B. (1974). *Theory of Probability*. Wiley, London.

[23] Dua, D. and Graff, C. (2019). UCI machine learning repository. *University of California, Irvine, School of Information and Computer Science*. Http://archive.ics.uci.edu/ml.

[24] Fayyad, U. and Irani, K. (1992). On the handling in decision tree of continuous-valued attributes generation. *Machine Learning*, 8, 87–102.

[25] Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learing. *In: Proceeding of the 13th International Joint Conference on Artificial Inteligence*, pp. 1022–1027.

[26] Fink P. (2018). *imptree: Classification Trees with Imprecise Probabilities*. R package version 0.5.1.

[27] Hill B. (1988). De Finetti's theorem, induction, and $A_{(n)}$ or Bayesian nonparametric predictive inference (with discussion). *Bayesian Statistics*, 3, 211–241.

[28] Hill, M. (1968). Posterior distribution of percentiles: Bayes' theorem for sampling from a population. *Journal of the American Statistical Association*, 63, 677–691.

[29] Hornik, K. and Buchta, C. and Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics*, 24, 225–232.

[30] James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer, New York.

[31] Mantas, C., Abellán, J. and Castellano, J. (2016). Analysis of Credal-C4. 5 for classification in noisy domains. *Expert Systems with Applications*, 61, 314–326.

[32] Mantas, C. and Abellán, J. (2014). Credal-C4. 5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41, 4625–4637.

[33] Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.

[34] Quinlan, J. (1993). *C4.5: Program for machine learning*. Morgan Kaufmann.

[35] R Core Team (2013). *R: A Language and Environment for Statistical Computing*.

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

[37] Therneau T., Atkinson B. and Ripley B. (2015). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.16.

[38] Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco.