IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 May 2024, accepted 17 June 2024, date of publication 19 June 2024, date of current version 27 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3416838

SURVEY

A Survey of Decision Trees: Concepts, Algorithms, and Applications

IBOMOIYE DOMOR MIENYE, (Member, IEEE), AND NOBERT JERE
Department of Information Technology, Walter Sisulu University, Buffalo City Campus, East London 5200, South Africa
Corresponding author: Ibomoiye Domor Mienye (imienye@wsu.ac.za)

ABSTRACT Machine learning (ML) has been instrumental in solving complex problems and significantly advancing different areas of our lives. Decision tree-based methods have gained significant popularity among the diverse range of ML algorithms due to their simplicity and interpretability. This paper presents a comprehensive overview of decision trees, including the core concepts, algorithms, applications, their early development to the recent high-performing ensemble algorithms and their mathematical and algorithmic representations, which are lacking in the literature and will be beneficial to ML researchers and industry experts. Some of the algorithms include classification and regression tree (CART), Iterative Dichotomiser 3 (ID3), C4.5, C5.0, Chi-squared Automatic Interaction Detection (CHAID), conditional inference trees, and other tree-based ensemble algorithms, such as random forest, gradient-boosted decision trees, and rotation forest. Their utilisation in recent literature is also discussed, focusing on applications in medical diagnosis and fraud detection.

INDEX TERMS Algorithms, CART, C4.5, C5.0, decision tree, ensemble learning, ID3, machine learning.

I. INTRODUCTION

Machine learning-based applications are revolutionising various industries and sectors, including healthcare, finance, and marketing [1], [2], [3], [4]. With the advancement of technology and the availability of large datasets, ML algorithms have become increasingly powerful and accurate in making predictions and informed decisions. These applications are transforming how organisations operate and paving the way for a more efficient and data-driven future.

Decision tree-based algorithms have been employed in diverse applications, including but not limited to classification, regression, and feature selection [5], [6], [7]. The basic idea behind decision tree-based algorithms is that they recursively partition the data into subsets based on the values of different attributes until a stopping criterion is met. This process results in a tree-like structure, where each node represents a decision or a split based on a specific attribute [8]. The algorithm determines the best attribute to use for each split based on certain criteria, such as information gain, gain ratio, and Gini index.

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

Furthermore, decision trees are known for their interpretability [9], [10]. The resulting tree structure allows users to understand and interpret the decision-making process easily. This is especially valuable in domains where transparency and explainability are crucial, making it easier for stakeholders to trust and validate the results. Another significance of decision tree-based algorithms is their ability to handle categorical and numerical data. Traditional statistical methods often struggle with categorical variables, requiring them to be converted into numerical values. Decision trees, on the other hand, can directly handle both types of data, eliminating the need for data preprocessing. This makes decision tree-based algorithms more versatile and efficient in a wide range of applications.

There are a few reviews of decision trees in the literature; for example, Che et al. [11] presented a review of decision trees and ensemble classifiers with specific applications to bioinformatics. The review focused on ID3, CART, and ensemble methods such as bagging, boosting, and stacked generalization. Cañete-Sifuentes et al. [12] reviewed multivariate decision trees (MDT) and compared the performance of several MDT induction classifiers. Anuradha and Gupta [13] presented a review of decision

tree classifiers, focusing on a high-level description of key concepts, such as node splitting and tree pruning. Meanwhile, Costa and Pedreira [14] reviewed recent decision tree-based classifier advances. The paper covered three main issues: how decision trees fit the training data, their generalization, and interpretability.

However, most of the existing surveys and reviews of decision trees focus on their applications in specific domains or a high-level overview of the decision tree concept. Therefore, the current literature lacks a comprehensive overview of decision tree algorithms, their early developments, succinct mathematical formulations, and algorithmic representations in a single peer-reviewed paper. Therefore, it is essential to have a review that fills this gap in view of the continuous use and prevalence of decision tree-based algorithms and their application in today's technological advancements. Hence, in this study, we present a detailed review of decision tree-based algorithms. Specifically, the paper aims to cover the different decision tree algorithms, including ID3, C4.5, C5.0, CART, conditional inference trees, and CHAID, together with other tree-based ensemble algorithms, such as random forest, rotation forest, and gradient boosting decision trees. The paper aims to present their mathematical formulations and algorithmic representations clearly and concisely.

The rest of the paper is structured as follows: Section II presents a comprehensive overview of the decision tree, covering key areas such as splitting criteria and tree pruning methods. Section III discusses different decision tree algorithms, their learning process, splitting criteria, and mathematical formulations. Section IV reviews decision tree applications in recent literature, including applications in medical diagnosis and fraud detection. Section V discusses key findings and future research directions, and Section VI concludes the paper.

II. OVERVIEW OF DECISION TREE

This section provides a comprehensive overview of decision trees, focusing on the main building blocks and splitting criteria. Decision trees, as a concept in ML, have a history that dates back to the mid-20th century. Initial decision tree studies were started by Charles J. Clopper and Egon S. Pearson in 1934, who introduced the concept of binary decision processes [15], [16]. However, the modern implementation of decision trees in the context of ML started decades later. Breiman [17] developed the CART algorithm in 1984, introducing concepts such as the Gini index and binary splitting, which are now widespread in decision tree designs. Quinlan [18] developed ID3, one of the first notable decision tree algorithms, in 1986. Furthermore, Quinlan [19] enhanced the ID3, introducing the C4.5 decision tree in 1993. These developments and integration of decision trees into ensemble methods like random forests and boosting algorithms have solidified their place as fundamental algorithms in machine learning.

The learning procedure of decision trees involves a series of steps where the data is split into homogenous subsets, as shown in Figure 1. The root node, which is the starting point of the tree, represents the entire dataset. The algorithm identifies the feature and the threshold that leads to the best split based on a specific criterion [20]. The process continues recursively, with each subset of the data being further split at each child node. This continues until a stopping criterion is reached, typically when the nodes are pure (i.e., all data points in a node belong to the same class) or when a predefined depth of the tree is reached. The nodes where the tree ends, called leaf node or terminal node, represent the outcomes or class labels. The decision to split at each node is made using mathematical formulations such as information gain, Gini impurity, or variance reduction.

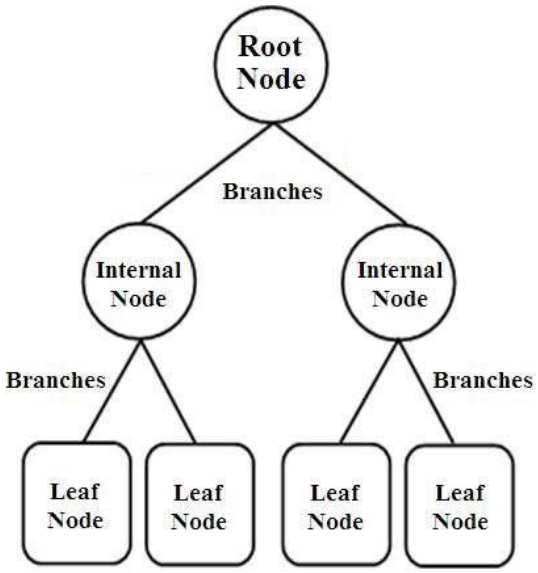

FIGURE 1. A decision tree example.

Furthermore, the success of decision tree techniques mainly depends on several factors contributing to their performance, interpretability, and applicability to a wide range of problems. These factors include data quality, tree depth, splitting criteria, and tree pruning method. According to Piramuthu [21], the effectiveness of decision trees is highly dependent on the training data quality. Hence, it is necessary to use clean or preprocessed data not containing missing values and outliers, which can significantly enhance the performance of the resulting models. Additionally, feature selection and feature engineering are necessary because inputting relevant and well-transformed features can lead to more efficient and accurate splits.

A. SPLITTING RULES

The term splitting criteria, or splitting rules, describes the methods used to determine where a tree should make a split in its nodes, effectively deciding how to divide the dataset into subsets based on different conditions [22], [23]. The choice of splitting criterion is crucial as it directly impacts the tree's structure and, ultimately, its performance. Different

decision tree algorithms use different criteria for this purpose, including the following:

1) GINI INDEX

Gini Index, also called Gini Impurity, is a well-known splitting criterion used in the CART algorithm. It measures the probability of a randomly chosen sample being incorrectly classified if it was randomly labelled [24]. It is used to evaluate the quality of a split in the tree and is calculated for each potential split in the dataset. The Gini Index for a set can be represented mathematically as:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{n} p_i^2 \tag{1}$$

where S, n, and p_i represent a set of samples, the number of unique classes in the set, and the proportion of the samples in the set that belong to class i, respectively. This formula calculates the probability of incorrectly classifying a randomly chosen element from the set S based on the distribution of classes in it. The value of Gini Impurity ranges from 0 (perfect purity) to 1 (maximal impurity) [25]. When the algorithm evaluates where to split the data, it calculates the Gini index for each potential split and typically chooses the split that results in the lowest weighted Gini Impurity for the resulting subsets.

2) INFORMATION GAIN

Information Gain (IG), a criterion used in ID3 and C4.5, is based on the notion of entropy in information theory. Entropy measures the unpredictability or randomness in a set of data [26]. The IG technique searches for a split that maximizes the difference in certainty or decreases uncertainty before and after the split. It determines the effectiveness of an attribute in splitting the training data into homogenous sets. Meanwhile, the entropy (E) of a set S is given by the formula:

$$E(S) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2}$$

where n is the number of unique classes in the set, and p_i is the proportion of the samples in the set that belong to class i. Therefore, the IG for a split on a dataset S with an attribute A can be computed as follows:

$$IG = E(S) - \sum_{v \in \ Values(A)} \frac{|S_v|}{|S|} E(S_v) \tag{3}$$

where $Values(A)$ are the different values that attribute A can take, and S_v is the subset of S for which attribute A has the value v [27]. This formula calculates the change in entropy from the original set S to the sets S_v created after the split. A higher IG indicates a more effective attribute for splitting the data, as it results in more homogeneous subsets.

3) INFORMATION GAIN RATIO

The information gain ratio (IGR), an extension of information gain, is a splitting criterion mainly used in the C4.5 decision tree to overcome the bias of information gain towards features that have several distinct values by considering the number and size of branches when choosing an attribute. The IGR normalises the information gain by dividing it by the intrinsic information or split information (SplitInfo) of the split. This normalisation reduces the bias towards the multi-valued attributes, resulting in more balanced and effective decision trees [26], [27]. The IGR criterion is calculated as:

$$IGR(S, A) = \frac{InformationGain(S, A)}{SplitInfo(S, A)} \tag{4}$$

4) CHI-SQUARE

The Chi-Square (χ^2) splitting criterion measures the independence between an attribute and the class [28]. The χ^2 test assesses whether the distribution of sample observations across different categories deviates significantly from what would be expected if the categories were independent of the class. Given an attribute A with different categories and a target class C, the χ^2 can be computed as:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{5}$$

where r is the number of categories of the attribute A, k is the number of classes, O_{ij} is the observed frequency in cell (i, j) that belong to class j), and E_{ij} is the expected frequency in cell (i, j) under the null hypothesis of independence, calculated as $E_{ij} = \frac{(row\_total_i \times column\_total_j)}{total\_samples}$. A high χ^2 value indicates a significant association between the attribute and the class, suggesting that the attribute is a good predictor for splitting the dataset [29], [30]. This criterion is useful for categorical data, and it identifies the most significant splits based on the chi-square test of independence.

B. TREE PRUNING METHODS

1) PRE-PRUNING

Pre-pruning or early stopping techniques are used to effectively limit the size of the tree and reduce the possibility of overfitting [31], [32]. The main benefit of pre-pruning is its simplicity and the reduction in computational cost due to the construction of smaller trees. However, setting the pre-pruning parameters too aggressively may lead to underfitting. Meanwhile, this strategy halts the tree's growth according to predefined criteria, such as maximum depth, minimum number of instances in a node, minimum information gain, and maximum number of leaf nodes [33].

2) POST-PRUNING

Post-pruning, also called backward pruning, is a technique used to trim down a fully grown tree to improve its generalization capabilities. Unlike pre-pruning, which stops the tree from fully growing, post-pruning allows the tree to first grow to its full size and then prunes it back [34]. Common post-pruning techniques include reduced error pruning, pessimistic error pruning, error-based pruning, minimum error pruning, and cost complexity pruning [33].

Post-pruning primarily removes sections of the tree that contribute little to predicting the target variable. It often requires a separate validation dataset to assess the impact of pruning [35]. This dataset tests the tree's performance as it undergoes pruning.

C. INTERPRETABILITY OF DECISION TREES

Decision trees are known for their inherent interpretability, making them valuable in various domains where understanding the decision-making process is crucial [14], [36]. Unlike many other ML algorithms that produce black-box models, decision trees offer transparency by representing the decision process as a sequence of simple, intuitive rules. Specifically, each node in a decision tree corresponds to a feature and a decision threshold, and the path from the root to a leaf node represents a series of decisions based on the feature values. This clear structure allows stakeholders to easily comprehend and interpret how the model arrives at its predictions.

Furthermore, while complex models such as deep neural networks and ensemble methods may achieve high accuracy, their black-box nature makes it challenging to understand how they arrive at their predictions [37], [38]. In contrast, decision trees provide a visual representation of the decision-making process, allowing stakeholders to trace each decision back to specific features and thresholds. For instance, in a medical diagnosis application, a decision tree model may reveal which symptoms or risk factors are most influential in predicting a particular disease. This transparency enables domain experts to validate the model's decisions and identify potential biases or errors, thereby improving trust in the model's predictions.

Additionally, decision trees can facilitate feature selection and variable importance analysis, aiding in feature engineering and model refinement [39], [40], [41]. By examining the splits in the tree and the associated feature importance scores, practitioners can identify the most influential features in the prediction process. This information can guide data preprocessing efforts and inform decisions about feature inclusion or exclusion in the model, leading to more efficient and interpretable models.

III. DECISION TREE ALGORITHMS

A. ITERATIVE DICHOTOMISER 3

The ID3 decision tree was first introduced in 1986 by Quinlan [18]. It is particularly noted for its simplicity and effectiveness in solving classification problems. The algorithm follows a top-down, greedy search approach through the given dataset to construct a decision tree. It begins with the entire dataset and divides it into subsets based on the attribute that maximizes the Information Gain (Equation 3), intending to efficiently classify the instances at each node of the tree. The ID3 is described in Algorithm 1.

The algorithm iterates through every unused attribute and calculates the Information Gain for a dataset split by the attribute's possible values. The attribute with the highest

Algorithm 1 ID3 Decision Tree Algorithm

Require: Training data set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
Ensure: Decision tree T.

```
function ID3(D)
    if D is empty then return a terminal node with default class c_default
    end if
    if all instances in D have same class label y then return a terminal node with class y
    end if
    if the attribute set J is empty then return a terminal node with the prevalent class in D
    end if
     Select the feature f that best splits the data using information gain.
     Create a decision node for f.
    for each value b_i of f do
        Create a branch for b_i.
        Let D_i be the subset of D where x_i = b_i.
        Recursively build the subtree for D_i.
        Attach the subtree to the branch for b_i.
    end for
    return the decision node.
end function
```

Information Gain is chosen to make the decision at the node, and the dataset is partitioned accordingly. This process is repeated recursively for each partitioned subset until one of the stopping criteria is met, such as when no further information can be gained, all instances in a subset belong to the same class, or there are no more attributes left to consider. Lastly, the ID3's limitations include its inability to directly handle continuous variables and overfitting.

B. 4.5 AND C5.0

Quinlan [19] proposed the C4.5 in 1993 as an extension of the ID3 algorithm and is designed to handle both continuous and discrete attributes. It introduces the concept of information gain ratio, described in Equation 4, to select the best attribute to split the dataset at each node, aiming to overcome the bias towards attributes with more levels found in the original Information Gain criterion used by ID3.

C5.0 is an improvement over C4.5, also proposed by Quinlan [42], designed to be faster and more memory efficient. It introduces several enhancements, such as advanced pruning methods and the ability to handle more complex types of data. C5.0 maintains the use of the information gain ratio for selecting attributes but optimises the algorithm's execution and the resulting decision tree's size.

C. CLASSIFICATION AND REGRESSION TREES

The CART decision tree was proposed in 1984 by Breiman [43]. Unlike C4.5, CART creates binary trees irrespective of the type of target variables. It uses different

splitting criteria for classification and regression tasks. For classification tasks, it uses the Gini index (Equation 1) as a measure to create splits [44], [45]. Meanwhile, it employs variance as the splitting criterion in regression tasks [46], [47]. The variance reduction for a set S when split on attribute A is calculated as:

$$VR = V(S) - \left(\frac{|S_{left}|}{|S|} V(S_{left}) + \frac{|S_{right}|}{|S|} V(S_{right}) \right) \tag{6}$$

where $V(S)$ is the variance of the target variable in set S, and S_{left} and S_{right} are the subsets of S after the split on attribute A. In both cases, the goal is to choose the split that maximizes the respective measure (Gini impurity reduction for classification and variance reduction for regression), leading to the most homogenous subsets possible. The CART algorithm is described in Algorithm 2.

Algorithm 2 CART Algorithm

Require: $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$.
Ensure: Decision tree T.

function CART(D)
  if D is empty **then return** a terminal node with default value or class $c_{default}$
  end if
  if all instances in D have the same class label y **then return** a terminal node with class y
  end if
  if the feature set F is empty **then return** a leaf node with the average value of y in D
  end if
  Select the best feature f and split point s that minimize the cost function.
  Create a decision node for f and s.
  Partition the data set D into two subsets D_1 and D_2 based on the split.
  Recursively build the subtree for D_1 and D_2.
  Attach the subtrees to the decision node.
  return the decision node.
end function

D. CHI-SQUARED AUTOMATIC INTERACTION DETECTION

The CHAID algorithm, developed by Kass [48], performs multi-level splits when computing classification trees. It is particularly robust in the detection of interaction between variables. CHAID can handle more than two categories for each variable, and it uses the Chi-Square (χ^2) test of independence as its splitting criterion [49], [50]. This statistical test is applied to assess the relationship between categorical variables. For a given attribute A with different categories and a target class C, the χ^2 statistic is computed as:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{7}$$

where r is the number of categories of the attribute A, k is the number of different classes in the target variable C, O_{ij} is the observed frequency in the i^{th} category of attribute A and the j^{th} class of C, and E_{ij} is the expected frequency in the same cell under the null hypothesis of independence, calculated as $E_{ij} = \frac{(row\_total_i \times column\_total_j)}{total\_samples}$. The attribute with the highest χ^2 statistic is selected for splitting at each node. A higher χ^2 value indicates a stronger association between the attribute and the target variable, suggesting that the attribute is a good predictor for splitting the dataset. Algorithm 3 details the working process of the CHAID algorithm.

Algorithm 3 CHAID Algorithm

Require: $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$.
Ensure: Decision tree T.

function CHAID(D)
  if D is empty **then return** a terminal node with default class $c_{default}$
  end if
  if all instances in D have the same class label y **then return** a terminal node with class y
  end if
  if the feature set F is empty **then return** a terminal node with the most prevalent class in D
  end if
  Calculate the chi-squared statistic for each feature and its possible values.
  Select the feature and value with the highest chi-squared value.
  Create a decision node for the selected feature and value.
  Partition the data set D based on the selected feature and value.
  for each subset D_i of D **do**
    Recursively build the subtree for D_i.
    Attach the subtree to the decision node.
  end for
  return the decision node.
end function

E. CONDITIONAL INFERENCE TREES

The conditional inference trees, developed by Hothorn et al. [51], is a non-parametric class of decision trees that use statistical tests to determine splits, reducing bias and variance and providing a more statistically sound approach. It is mostly useful when solving complex, non-linear relationships that exist between the predictor variables and the response variable [52], [53]. Assuming S is a node in the tree, with m examples and d features. Let X_s be the subset of d features at node S, and Y_s be the corresponding response values. Let X_j

be the j-th feature in X_s. Then, the algorithm can be defined as:

1) For each feature X_j in X_s, calculate the p-value of a statistical test for the null hypothesis that there is no relationship between X_j and Y_s.
2) Choose the feature X_k and split point t_k that maximize the statistical significance, based on the p-values of the tests.
3) Split the node into two child nodes S_1 and S_2, where S_1 contains examples with $X_k \leq t_k$ and S_2 contains examples with $X_k > t_k$.
4) Recursively repeat steps 1-3 to every child node until a stopping criterion is reached.

F. RANDOM FOREST

the random forest, described in Algorithm 4, is an ensemble of decision trees [54], [55]. It improves upon the basic decision tree algorithm by reducing overfitting. Each tree in the forest is built from a sample drawn with replacement (i.e., bootstrap sample) from the input data [56]. The basic idea behind this algorithm is to generate a set of trees using different subsets of the input samples and features and then combine their outputs to obtain a final prediction. The Random Forest algorithm uses two main techniques to reduce overfitting and improve accuracy:

- Bootstrap Sampling: By sampling the data with replacement, the algorithm generates multiple training sets that are slightly different from each other. This type of sampling ensures reduced variance and prevents overfitting.
- Feature Randomization: Randomly selecting a subset of features for each tree ensures the algorithm decorrelates the trees and reduces the chance of selecting the same "best" feature for every tree. This improves the diversity and accuracy of the trees.

Algorithm 4 Random Forest Algorithm

for $t = 1$ to T **do** ▷ Generate T trees
  Randomly sample n instances from D with replacement
  Randomly select m attributes from the total p attributes (where $m \ll p$)
  Build a decision tree h_t based on the sampled instances and attributes
end for
end for
To make predictions for a new instance x:
if classification task **then**
  $f(x) = \text{argmax}_c \frac{1}{T} \sum_{t=1}^{T} I\{h_t(x) = c\}$ ▷ Majority vote across trees
else if regression task **then**
  $f(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$ ▷ Average of tree predictions
end if
end if

G. GRADIENT BOOSTED DECISION TREES

Gradient Boosted Decision Trees (GBDT) is an ensemble learning method that combines multiple decision trees to create a powerful predictive model [57]. Unlike Random Forest, which builds independent trees in parallel, GBDT uses a sequential approach to build trees that correct the errors of the previous trees [58], [59]. It uses gradient descent to minimize errors. Assuming T is the number of trees, $h_t(x)$ is the prediction of the t-th tree, $F_{t-1}(x)$ is the current model's predictions for x, and $L(y, F_{t-1}(x))$ is the loss function, the GBDT algorithm works as follows:

1) Initialize the model with a constant value (e.g., the mean of the target variable).
2) For $t = 1$ to T:
   a) Compute the negative gradient of the loss function with respect to the current model's predictions for each instance in the training data.
   b) Fit a decision tree to the negative gradient values, using the input data as features and the negative gradient values as target variables.
   c) Update the model by adding the new tree, weighted by a learning rate η, to the current model.
3) Make a prediction for a new instance by summing the predictions from the various trees:
   a) For a regression task, the final prediction is the sum of the predictions of all the trees, i.e., $f(x)$ is given by:

$$f(x) = \sum_{t=1}^{T} \eta h_t(x) \tag{8}$$

   where η is the learning rate.
   b) For a classification task, the final prediction is the probability of the positive class, computed by applying a sigmoid function to the sum of the predictions of all the trees.

$$f(x) = \frac{1}{1 + e^{-\sum_{t=1}^{T} \eta h_t(x)}} \tag{9}$$

   where η is the learning rate and e is the Euler's number.

H. ROTATION FOREST

Rotation forest is a type of decision tree ensemble where each tree is trained on the principal components of a randomly selected subset of features [60], [61]. The core idea behind this algorithm is to train each classifier in the ensemble on a version of the training data that has been transformed to maintain the correlation between the features and introduce diversity among the classifiers. This is achieved through the following steps:

1) For each classifier to be trained, partition the set of features F into k subsets. The partitioning can be random but is done in such a way that each subset contains a different part of the features.

TABLE 1. Summary of decision tree algorithms.

Algorithm	Overview	Advantages	Disadvantages
ID3	A simple, greedy algorithm using Information Gain as the rule. Ideal for categorical data.	Easy to understand and implement.	Prone to overfitting; not for continuous variables.
C4.5	Extends ID3, improves handling of continuous data, and introduces tree pruning.	Handles both continuous and categorical data; uses pruning to reduce overfitting.	More complex than ID3; slower with large data.
C5.0	Evolution of C4.5, focuses on efficiency and scalability, uses Information Gain Ratio.	Efficient with large datasets; generates smaller trees.	Can overfit without proper pruning; less interpretable.
CART	Uses Gini index for classification and variance reduction for regression, makes binary splits.	Versatile for both classification and regression; simplifies the model.	Can overfit; needs careful tuning and pruning.
CHAID	Employs Chi-square test for multi-level splits, identifies variable interactions.	Good for categorical data and detecting interactions.	Can produce large trees; sensitive to data changes.
Conditional Inference Trees	Uses statistical tests for splitting to minimize bias, suitable for categorical variables.	Provides a statistically rigorous approach; less biased towards variables with many categories.	Computationally intensive; less intuitive than simpler models.
Random Forest	An ensemble of trees using bagging to reduce overfitting and improve accuracy.	Reduces overfitting; handles high dimensionality well.	Less interpretable; higher computational costs.
Gradient Boosted Decision Trees	Builds trees sequentially to correct errors, using variance reduction criteria.	High accuracy; effective with various data types.	Prone to overfitting; requires careful tuning.
Rotation Forest	Enhances model diversity by rotating the feature space using PCA before building each tree.	Improves accuracy by promoting diversity.	Increased computational complexity; challenging tuning.

2) For each subset of features, apply PCA to obtain the principal components. This step transforms the original feature space into a new space that captures the variance in the data more effectively.
3) Combine the principal components from all subsets to form a new set of features for training the classifier. This effectively rotates the axis of the feature space, hence the name *Rotation Forest*.
4) Train each base classifier on the transformed dataset. Different classifiers can be used, but decision trees are commonly applied.

Given a dataset D with n features, the algorithm partitions the feature set F into k non-overlapping subsets $F_1, F_2, \ldots, F_k$. For each subset F_i, PCA is applied to derive a set of principal components PC_i, capturing the main variance directions of the features in F_i. The transformation for a subset F_i can be represented as:

$$T_i = PCA(F_i) \tag{10}$$

where T_i is the transformation matrix obtained from PCA on subset F_i. The new feature set for training the j^{th} classifier, D_j, is obtained by applying the transformation T_i to each subset F_i and concatenating the results:

$$D_j = \bigoplus_{i=1}^{k} T_i(F_i) \tag{11}$$

where $\bigoplus$ denotes the concatenation of the transformed feature subsets. The ensemble's final output is typically the majority vote (for classification tasks) of the predictions from all base classifiers.

A summary of the different tree-based algorithms is tabulated in Table 1, including their advantages and disadvantages.

IV. DECISION TREE APPLICATIONS IN RECENT LITERATURE

Decision trees have gained significant attention in recent literature. This section discusses some popular applications of decision trees in fields such as healthcare and finance.

A. MEDICAL DIAGNOSIS

Healthcare is one of the prominent areas where decision trees have found extensive use. Researchers have utilized decision trees to predict disease diagnosis, treatment outcomes, and patient prognosis. Decision trees are effective in identifying patterns and relationships in medical data, leading to more accurate diagnoses and personalized treatment plans. For example, decision trees have been used to predict the likelihood of a patient developing a specific disease based on their medical history and lifestyle factors [11], [62], [63]. This information can then be used to implement preventive measures and interventions, ultimately improving patient outcomes and reducing healthcare costs.

Pathak and Arul Valan [64] proposed a heart disease prediction model using a decision tree. The model was built using a fuzzy rule-based technique combined with a decision tree, achieving an accuracy of 88% when trained on the Cleveland heart disease dataset obtained from the

University of California Irvine (UCI) machine learning repository. Similarly, Maji and Arora [65] conducted a study on heart disease prediction using a different dataset from the UCI machine learning repository. The study employed the C4.5 decision tree and a hybrid decision tree made of C4.5 and artificial neural network (ANN), where the former achieved an accuracy of 76.66% and the latter 78.14%. The study demonstrated the robustness of hybridising decision trees with neural networks.

Ahmad et al. [66] studied the performance of several algorithms using different heart disease datasets, including Cleveland, Switzerland, and Long Beach. The algorithms studied include random forest, decision tree, support vector machine (SVM), k-nearest neighbor (KNN), linear discriminant analysis, and gradient boosting classifier. The study employed sequential feature selection (SFS) to obtain the most significant features, which were then used to train the models. The study concluded that the random forest-SFS and decision tree-SFS achieved the best accuracy. For the Cleveland dataset, the random forest and decision tree obtained accuracies of 100.

In [67], the authors identified the C4.5 and random forest as potentially robust algorithms for detecting chronic kidney disease (CKD) stages. The study employed a CKD dataset from the UCI machine learning repository, comprising 25 features and 400 samples. The results indicated that the C4.5 achieved an accuracy of 85.5%, outperforming the random forest, which achieved an accuracy of 78.25%.

Decision tree-based methods have also been employed to diagnose COVID-19. Ahmad et al. [66] proposed a deep learning-based decision tree model to detect COVID-19 using chest X-ray images. The approach consists of three decision trees trained using deep learning architectures, including a convolutional neural network (CNN). One tree classifies the images as normal or abnormal, another tree detects tuberculosis indicators in the abnormal images, and the last detects COVID-19. The approach achieved an average accuracy of 95%. Ghiasi and Zendehboudi [68] proposed a decision tree-based ensemble classifier for detecting breast cancer. The study used the well-known Wisconsin Breast Cancer dataset and aimed to build a robust breast cancer detection framework using the random forest and extra trees classifier (ET). The approach resulted in an accuracy of 100%.

Mienye and Sun [69] studied the performance of ML algorithms for heart disease prediction. The study utilized the following algorithms: decision tree, XGBoost, random forest, logistic regression, and naive Bayes. Firstly, the authors employed the Synthetic Minority Oversampling Technique-Edited Nearest Neighbor (SMOTE-ENN) to resample the data and solve the imbalance class problem. Also, the recursive feature elimination technique was employed to identify the most significant attributes to further enhance the classification performance of the models. The results showed that the decision tree, random forest, and XGBoost achieved an accuracy of 87.7%, 93%, and 95.6%, respectively, with the XGBoost obtaining the highest accuracy.

Meanwhile, Adler et al. [70] developed a Glaucoma detection method using the random forest ensemble classifier. The study evaluated the performance of ensemble pruning on the imbalanced glaucoma dataset. The ensemble pruning techniques include pruning by prediction accuracy (using the Brier Score strategy), pruning by uncertainty-weighted accuracy (UWA), and pruning by diversity (using the Double-Fault measure). The experimental results indicated that the RF model reached an area under the receiver operating characteristic curve (AUC) of 0.98 for the Brier and double-fault pruning techniques.

Additionally, Mienye et al. [71] employed decision tree, SVM, and logistic regression for CKD detection. The selected algorithms were also used as the base learners in the AdaBoost ensemble. The study reported accuracies of 94% and 100% for the decision tree and AdaBoost classifier that used a decision tree as a based learner. The study demonstrated the robustness of using a decision tree in the AdaBoost over the SVM and logistic regression. Furthermore, Mienye and Sun [72] studied the impact of cost-sensitive ML in medical diagnosis using the following algorithms: decision tree, random forest, and XGBoost. Cost-sensitive learning involves modifying the algorithm to focus on the minority class samples, thereby enhancing the model's performance on the minority class, which in most applications is of higher importance than the majority class. When applied for detecting cervical cancer, the cost-sensitive random forest obtained the highest classification accuracy of 98.8%, outperforming the other cost-sensitive and standard algorithms.

Furthermore, Khan et al. [73] proposed an ensemble approach called optimal trees ensemble (OTE) and applied it to diverse classification problems, including hepatitis and Pakinson's disease detection, achieving error rates of 0.1230 and 0.0861, respectively. The error rates, which translate to 87.7% and 91.4% accuracy, imply the proposed OTE outperformed other baseline models, including KNN, LDA, and random forest. Table 2 summarizes the discussed studies on medical diagnosis, indicating how decision trees have been employed in the medical domain, achieving excellent classification performance.

B. FINANCE

Decision trees have also been widely employed in the field of finance. By analysing historical data and identifying relevant variables, decision trees can accurately predict the creditworthiness of individuals. This information is crucial for banks and lending institutions in determining the risk associated with granting loans [74], [75]. Furthermore, decision trees have been used to detect fraudulent activities in financial transactions by examining transactional data and identifying suspicious patterns, helping to prevent financial losses.

TABLE 2. Summary of the medical diagnosis studies.

Reference	Year	Algorithm	Application	Accuracy(%)
Adler et al. [70]	2016	Random forest ensemble pruning	Glaucoma	-
Maji and Arora [65]	2018	C4.5	Heart Disease	76.66
Maji and Arora [65]	2018	Hybrid DT of C4.5 and ANN	Heart Disease	78.14
Khan et al. [73]	2019	Optimal trees ensemble	Hepatitis	87.7
Khan et al. [73]	2019	Optimal trees ensemble	Pakinson's disease	91.4
Pathak et al. [64]	2020	C4.5 Decision Tree	Heart Disease	88.0
Yoo et al. [66]	2020	Deep learning-based Decision Tree	COVID-19	95.0
Mienye et al. [71]	2021	AdaBoost-DT	Chronic Kidney Disease	100
Ilyas et al. [67]	2021	C4.5	Chronic Kidney Disease	85.5
Ilyas et al. [67]	2021	Random forest	Chronic Kidney Disease	78.25
Ghiasi and Zendehboudi [68]	2021	Random forest and ET	Breast Cancer	100
Mienye and Sun [72]	2021	Cost-sensitive random forest	Cervical Cancer	98.8
Mienye and Sun [69]	2021	XGBoost	Heart Disease	95.6
Ahmad et al. [66]	2022	Random forest	Heart Disease	100
Ahmad et al. [66]	2022	CART	Heart Disease	100

Yao et al. [76] studied credit risk within an enterprise setting. The study proposed a decision tree-based ensemble classifier that uses the SMOTE and AdaBoost algorithms. The proposed model was aimed at identifying enterprise credit risk by incorporating supply chain information. Other benchmark models were built using KNN, logistic regression, SVM, and random forest. The study indicated that the proposed decision tree ensemble achieved the best and most stable performance, obtaining an AUC of 0.902.

Liu et al. [77] developed an approach for financial institutions to effectively predict credit risk and enhance profitability. The proposed approach uses the gradient-boosting decision tree. While the GBDT was efficient in predicting the credit risk, it lacked sufficient interpretability. Therefore, the study introduced an enhanced method called tree-based augmented GBDT, which uses a step-wise feature augmentation framework. The proposed approach achieved a classification accuracy of 93.78%, outperforming the standard GBDT and displaying robust interpretability.

Alam et al. [78] studied the imbalance class problem in credit risk prediction. The study employed different credit risk datasets, including the German credit approval dataset, the Taiwan dataset, and the European credit card clients dataset. The gradient-boosted decision tree model combined with the k-means SMOTE technique achieved accuracies of 84.6%, 89%, and 87.1% on the German, Taiwan, and European clients datasets, respectively.

Hancock and Khoshgoftaar [79] employed gradient-boosted decision tree-based algorithms for detecting health insurance fraud. This is an important ML application as healthcare fraud is capable of denying patients the needed medical attention. In this study, the authors employed claims data to train the various classifiers, including categorical boosting (CatBoost), achieving an AUC of 0.775, outperforming other ML algorithms. The study went further to demonstrate the model's performance after introducing a new variable called Healthcare provider state, leading to the CatBoost obtaining an AUC of 0.882.

Wong et al. [80] conducted a comparative study of ML algorithms for credit risk prediction. The study focused on decision tree, random forest, KNN, logistic regression, and naive Bayes classifiers. The aim of the study was to assess which classifier would achieve the highest performance in terms of accuracy and other metrics. The experimental results indicated that the decision tree and random forest achieved an accuracy of 92.11% and 94.57%, with the random forest outperforming the other classifiers, demonstrating the robustness of tree-based ensemble classifiers.

Seera et al. [81] employed a decision tree for credit card fraud detection, using credit card transaction records in Malaysia, obtaining a classification accuracy of 99.96%. Rawat et al. [82] studied the performance of four classifiers on credit credit card fraud detection. The classifiers include logistic regression, RF, KNN, and AdaBoost. The various models achieved classification accuracies of 99%. Similarly, Adhegaonkar et al. [83] employed decision tree, random forest, logistic regression, and SVM for credit card fraud detection. The experimental results showed that the decision tree obtained an accuracy of 84.9%. However, the random forest obtained the best performance with an accuracy of 85.2%. A summary of the reviewed papers is tabulated in Table 3.

V. DISCUSSIONS AND FUTURE RESEARCH DIRECTIONS

Decision trees have proven to be effective in various domains, including healthcare and finance. However, like any other algorithm, decision trees have their limitations and areas for improvement. In this section, we will explore some potential future research directions in decision trees that can enhance their performance and address their limitations.

Firstly, the handling of missing data is a crucial area of potential improvement for decision trees. Currently, decision trees either ignore instances with missing values or use surrogate splits to make predictions [86], [87]. However, these approaches may not always be optimal and can lead to biased or inaccurate results. Future research could focus on developing more sophisticated methods to handle missing data in decision trees, such as advanced imputation techniques or incorporating uncertainty estimation.

TABLE 3. Summary of the credit risk and fraud detection studies.

Reference	Year	Algorithm	Application	Accuracy(%)
Nadim et al. [84]	2019	Random forest	Credit card fraud detection	98.6
Makki et al. [85]	2019	C5.0	Credit Card Fraud Detection	96.0
Wong et al. [80]	2020	Random forest	Credit Risk Prediction	94.6
Wong et al. [80]	2020	Decision tree	Credit Risk Prediction	92.1
Alam et al. [78]	2020	GDBT and k-means SMOTE	Credit Risk Prediction (German dataset)	84.6
Alam et al. [78]	2020	GDBT and k-means SMOTE	Credit Risk Prediction (Taiwan dataset)	89.0
Alam et al. [78]	2020	GDBT and k-means SMOTE	Credit card fraud detection	87.1
Seera et al. [81]	2021	CART	Credit card fraud detection	99.9
Hancock et al. [79]	2021	CatBoost	Healthcare Insurance Fraud	-
Yao et al. [76]	2022	Decision tree ensemble with SMOTE	Enterprise credit risk	-
Liu et al. [77]	2022	Augmented GBDT	Credit risk prediction	93.8
Seera et al. [81]	2024	AdaBoost	Credit card fraud detection	99.0
Adhegaonkar et al. [83]	2024	Random forest	Credit card fraud detection	85.2

Another future research direction will be enhancing the ability of decision trees to handle high-dimensional data [88], [89], [90]. Decision trees can struggle when faced with datasets that have a large number of features, as the tree structure becomes complex and prone to overfitting. Future research could explore techniques to improve the scalability and efficiency of decision trees in high-dimensional settings, such as feature selection methods or dimensionality reduction techniques.

Furthermore, while decision trees are known for their interpretability compared to other machine learning algorithms, they can still be difficult to understand and explain, especially when they become large and complex. Future research could investigate methods to simplify decision trees and make them more understandable to non-experts, such as rule extraction algorithms or visualisation techniques. Additionally, decision trees are sensitive to outliers and can easily be influenced by noisy data, leading to inaccurate predictions [91]. It might be worth examining the robustness of decision trees to outliers and noisy data and exploring methods to make decision trees more robust to outliers and noise, such as outlier detection techniques or robust splitting criteria.

Lastly, the application of decision trees in emerging fields and domains is a potential future research direction. Decision trees have been extensively studied and applied in traditional domains such as healthcare, finance, and marketing. However, there are numerous emerging fields where decision trees can potentially make a significant impact. For example, decision trees could be applied in the field of autonomous vehicles to aid in decision-making processes or in the field of natural language processing to improve sentiment analysis and text classification tasks. Future research could explore the potential applications of decision trees in these emerging fields and investigate their effectiveness in solving complex problems.

VI. CONCLUSION

Decision trees have shown great potential and effectiveness in various fields. Their ability to analyse complex data and identify patterns and relationships makes them valuable in the field of machine learning. This paper presented an overview of the decision trees, including their early development to the recent high-performing tree-based ensemble methods. The article covers the main decision tree algorithms, such as CART, ID3, C4.5, C5.0, CHAID, and conditional inference trees. Their applications in medical diagnosis, credit risk, and fraud detection were reviewed. This study will be beneficial to ML practitioners and researchers trying to understand decision trees and the widely used tree-based algorithms.

REFERENCES

[1] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature Commun.*, vol. 11, no. 1, Aug. 2020, Art. no. 3923.

[2] G. Obaido, F. J. Agbo, C. Alvarado, and S. S. Oyelere, "Analysis of attrition studies within the computer sciences," *IEEE Access*, vol. 11, pp. 53736–53748, 2023.

[3] S. Ahmed, M. M. Alshater, A. E. Ammari, and H. Hammami, "Artificial intelligence and machine learning in finance: A bibliometric review," *Res. Int. Bus. Finance*, vol. 61, Oct. 2022, Art. no. 101646.

[4] G. Obaido, B. Ogbuokiri, C. W. Chukwu, F. J. Osaye, O. F. Egbelowo, M. I. Uzochukwu, I. D. Mienye, K. Aruleba, M. Primus, and O. Achilonu, "An improved ensemble method for predicting hyperchloremia in adults with diabetic ketoacidosis," *IEEE Access*, vol. 12, pp. 9536–9549, 2024.

[5] C. Wang, J. Xu, S. Tan, and L. Yin, "Secure decision tree classification with decentralized authorization and access control," *Comput. Standards Interfaces*, vol. 89, Apr. 2024, Art. no. 103818.

[6] M. M. Rahman and S. A. Nisher, "Predicting average localization error of underwater wireless sensors via decision tree regression and gradient boosted regression," in *Proc. Int. Conf. Inf. Commun. Technol. Develop.* Singapore: Springer, 2023, pp. 29–41.

[7] T. O'Halloran, G. Obaido, B. Otegbade, and I. D. Mienye, "A deep learning approach for maize lethal necrosis and maize streak virus disease detection," *Mach. Learn. Appl.*, vol. 16, Jun. 2024, Art. no. 100556.

[8] R. Rivera-Lopez, J. Canul-Reich, E. Mezura-Montes, and M. A. Cruz-Chávez, "Induction of decision trees as classification models through metaheuristics," *Swarm Evol. Comput.*, vol. 69, Mar. 2022, Art. no. 101006.

[9] O. Sagi and L. Rokach, "Explainable decision forest: Transforming a decision forest into an interpretable tree," *Inf. Fusion*, vol. 61, pp. 124–138, Sep. 2020.

[10] L.-A. Dong, X. Ye, and G. Yang, "Two-stage rule extraction method based on tree ensemble model for interpretable loan evaluation," *Inf. Sci.*, vol. 573, pp. 46–64, Sep. 2021.

[11] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision tree and ensemble learning algorithms with their applications in bioinformatics," in *Advances in Experimental Medicine and Biology*. New York, NY, USA: Springer, 2011, pp. 191–199.

[12] L. Cañete-Sifuentes, R. Monroy, and M. A. Medina-Pérez, "A review and experimental comparison of multivariate decision trees," *IEEE Access*, vol. 9, pp. 110451–110479, 2021.

[13] A. Dhull and G. Gupta, "A self explanatory review of decision tree classifiers," in *Proc. Int. Conf. Recent Adv. Innov. Eng. (ICRAIE)*, May 2014, pp. 1–7.

[14] V. G. Costa and C. E. Pedreira, "Recent advances in decision trees: An updated survey," *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 4765–4800, May 2023.

[15] C. Gupta and A. Ramdas, "Distribution-free calibration guarantees for histogram binning without sample splitting," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3942–3952.

[16] F. Mazurek, A. Tschand, Y. Wang, M. Pajic, and D. Sorin, "Rigorous evaluation of computer processors with statistical model checking," in *Proc. 56th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2023, pp. 1242–1254.

[17] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[18] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[19] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Amsterdam, The Netherlands: Elsevier, 2014.

[20] I. D. Mienye, Y. Sun, and Z. Wang, "Prediction performance of improved decision tree-based algorithms: A review," *Proc. Manuf.*, vol. 35, pp. 698–703, Jan. 2019.

[21] S. Piramuthu, "Input data for decision trees," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1220–1226, Feb. 2008.

[22] S. Hwang, H. G. Yeo, and J.-S. Hong, "A new splitting criterion for better interpretable trees," *IEEE Access*, vol. 8, pp. 62762–62774, 2020.

[23] J.-S. Hong, J. Lee, and M. K. Sim, "Concise rule induction algorithm based on one-sided maximum decision tree approach," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121365.

[24] D. Bertsimas and J. Dunn, "Optimal classification trees," *Mach. Learn.*, vol. 106, no. 7, pp. 1039–1082, Jul. 2017.

[25] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "The CART decision tree for mining data streams," *Inf. Sci.*, vol. 266, pp. 1–15, May 2014.

[26] C. J. Mantas, J. Abellán, and J. G. Castellano, "Analysis of credal-C4.5 for classification in noisy domains," *Expert Syst. Appl.*, vol. 61, pp. 314–326, Nov. 2016.

[27] G. S. Reddy and S. Chittineni, "Entropy based C4.5-SHO algorithm with information gain optimization in data mining," *PeerJ Comput. Sci.*, vol. 7, p. e424, Apr. 2021.

[28] N. Peker and C. Kubat, "Application of chi-square discretization algorithms to ensemble classification methods," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115540.

[29] L. A. Badulescu, "A chi-square based splitting criterion better for the decision tree algorithms," in *Proc. 25th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2021, pp. 530–534.

[30] F. Mahan, M. Mohammadzad, S. M. Rozekhani, and W. Pedrycz, "Chi-MFlexDT: Chi-square-based multi flexible fuzzy decision tree for data stream classification," *Appl. Soft Comput.*, vol. 105, Jul. 2021, Art. no. 107301.

[31] F. M. J. M. Shamrat, S. Chakraborty, M. M. Billah, P. Das, J. N. Muna, and R. Ranjan, "A comprehensive study on pre-pruning and post-pruning methods of decision tree classification algorithm," in *Proc. 5th Int. Conf. Trends Electron. Informat. (ICOEI)*, Jun. 2021, pp. 1339–1345.

[32] Y. Manzali and Pr. M. E. Far, "A new decision tree pre-pruning method based on nodes probabilities," in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, May 2022, pp. 1–5.

[33] S. Trabelsi, Z. Elouedi, and K. Mellouli, "Pruning belief decision tree methods in averaging and conjunctive approaches," *Int. J. Approx. Reasoning*, vol. 46, no. 3, pp. 568–595, Dec. 2007.

[34] T. Lazebnik and S. Bunimovich-Mendrazitsky, "Decision tree post-pruning without loss of accuracy using the SAT-PP algorithm with an empirical evaluation on clinical data," *Data Knowl. Eng.*, vol. 145, May 2023, Art. no. 102173.

[35] E. Frantar and D. Alistarh, "SparseGPT: Massive language models can be accurately pruned in one-shot," in *Proc. 40th Int. Conf. Mach. Learn.*, vol. 202, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., Jul. 2023, pp. 10323–10337.

[36] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, pp. 1–11, Jan. 2021.

[37] S. J. Oh, B. Schiele, and M. Fritz, "Towards reverse-engineering black-box neural networks," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Lecture Notes in Computer Science), vol. 11700, W. Samek, G. Montavon, A. Vedaldi, L. Hansen, and K. R. Müller, Eds. Cham, Switzerland: Springer, 2019, pp. 121–144, doi: 10.1007/978-3-030-28954-6_7.

[38] E. Zihni, V. I. Madai, M. Livne, I. Galinovic, A. A. Khalil, J. B. Fiebach, and D. Frey, "Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0231166.

[39] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinf.*, vol. 9, no. 1, pp. 1–11, Dec. 2008.

[40] S. M. F. D. S. Mustapha, "Predictive analysis of students' learning performance using data mining techniques: A comparative study of feature selection methods," *Appl. Syst. Innov.*, vol. 6, no. 5, p. 86, Sep. 2023.

[41] S. Ben Jabeur, N. Stef, and P. Carmona, "Bankruptcy prediction using the XGBoost algorithm and variable importance feature engineering," *Comput. Econ.*, vol. 61, no. 2, pp. 715–741, Feb. 2023.

[42] J. R. Quinlan. (2004). *Data Mining Tools See5 and C5.0*. [Online]. Available: http://www.rulequest.com/see5-info.html

[43] L. Breiman, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.

[44] M.-M. Chen and M.-C. Chen, "Modeling road accident severity with comparisons of logistic regression, decision tree and random forest," *Information*, vol. 11, no. 5, p. 270, May 2020.

[45] D.-H. Lee, S.-H. Kim, and K.-J. Kim, "Multistage MR-CART: Multiresponse optimization in a multistage process using a classification and regression tree method," *Comput. Ind. Eng.*, vol. 159, Sep. 2021, Art. no. 107513.

[46] E. Belli and S. Vantini, "Measure inducing classification and regression trees for functional data," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 15, no. 5, pp. 553–569, Oct. 2022.

[47] H. Ishwaran, "The effect of splitting on random forests," *Mach. Learn.*, vol. 99, no. 1, pp. 75–118, Apr. 2015.

[48] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 29, no. 2, pp. 119–127, 1980.

[49] S. Kushiro, S. Fukui, A. Inui, D. Kobayashi, M. Saita, and T. Naito, "Clinical prediction rule for bacterial arthritis: Chi-squared automatic interaction detector decision tree analysis model," *SAGE Open Med.*, vol. 11, Jan. 2023, Art. no. 205031212311609.

[50] H. Prasetyono, A. Abdillah, T. Anita, A. Nurfarkhana, and A. Sefudin, "Identification of the decline in learning outcomes in statistics courses using the chi-squared automatic interaction detection method," *J. Phys., Conf. Ser.*, vol. 1490, no. 1, Mar. 2020, Art. no. 012072.

[51] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased recursive partitioning: A conditional inference framework," *J. Comput. Graph. Statist.*, vol. 15, no. 3, pp. 651–674, Sep. 2006.

[52] N. Levshina, "Conditional inference trees and random forests," in *A Practical Handbook of Corpus Linguistics*. Cham, Switzerland: Springer, 2020, pp. 611–643.

[53] B. Schivinski, "Eliciting brand-related social media engagement: A conditional inference tree framework," *J. Bus. Res.*, vol. 130, pp. 594–602, Jun. 2021.

[54] N. Younas, A. Ali, H. Hina, M. Hamraz, Z. Khan, and S. Aldahmani, "Optimal causal decision trees ensemble for improved prediction and causal inference," *IEEE Access*, vol. 10, pp. 13000–13011, 2022.

[55] Z. Khan, A. Gul, O. Mahmoud, M. Miftahuddin, A. Perperoglou, W. Adler, and B. Lausen, "An ensemble of optimal trees for class membership probability estimation," in *Analysis of Large and Complex Data*. Cham, Switzerland: Springer, 2016, pp. 395–409.

[56] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022.

[57] Z. Zhang and C. Jung, "GBDT-MO: Gradient-boosted decision trees for multiple outputs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3156–3167, Jul. 2021.

[58] M.-J. Jun, "A comparison of a gradient boosting decision tree, random forests, and artificial neural networks to model urban land use changes: The case of the Seoul metropolitan area," *Int. J. Geographical Inf. Sci.*, vol. 35, no. 11, pp. 2149–2167, Nov. 2021.

[59] V. A. Dev and M. R. Eden, "Formation lithology classification using scalable gradient boosted decision trees," *Comput. Chem. Eng.*, vol. 128, pp. 392–404, Sep. 2019.
[60] S. Demir and E. K. Sahin, "Comparison of tree-based machine learning algorithms for predicting liquefaction potential using canonical correlation forest, rotation forest, and random forest based on CPT data," *Soil Dyn. Earthq. Eng.*, vol. 154, Mar. 2022, Art. no. 107130.
[61] E. K. Sahin, I. Colkesen, and T. Kavzoglu, "A comparative assessment of canonical correlation forest, random forest, rotation forest and logistic regression methods for landslide susceptibility mapping," *Geocarto Int.*, vol. 35, no. 4, pp. 341–363, Mar. 2020.
[62] F. L. Seixas, B. Zadrozny, J. Laks, A. Conci, and D. C. Muchaluat Saade, "A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer's disease and mild cognitive impairment," *Comput. Biol. Med.*, vol. 51, pp. 140–158, Aug. 2014.
[63] G. Obaido, B. Ogbuokiri, I. D. Mienye, and S. M. Kasongo, "A voting classifier for mortality prediction post-thoracic surgery," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2022, pp. 263–272.
[64] A. K. Pathak and J. A. Valan, "A predictive model for heart disease diagnosis using fuzzy logic and decision tree," in *Smart Computing Paradigms: New Progresses and Challenges* (Advances in Intelligent Systems and Computing). Singapore: Springer, 2019, pp. 131–140.
[65] S. Maji and S. Arora, "Decision tree algorithms for prediction of heart disease," in *Information and Communication Technology for Competitive Strategies*. Singapore: Springer, Aug. 2018, pp. 447–454.
[66] G. N. Ahmad, S. Ullah, A. Algethami, H. Fatima, and S. Md. H. Akhter, "Comparative study of optimum medical diagnosis of human heart disease using machine learning technique with and without sequential feature selection," *IEEE Access*, vol. 10, pp. 23808–23828, 2022.
[67] H. Ilyas, S. Ali, M. Ponum, O. Hasan, M. T. Mahmood, M. Iftikhar, and M. H. Malik, "Chronic kidney disease diagnosis using decision tree algorithms," *BMC Nephrology*, vol. 22, no. 1, Dec. 2021, Art. no. 273.
[68] M. M. Ghiasi and S. Zendehboudi, "Application of decision tree-based ensemble learning in the classification of breast cancer," *Comput. Biol. Med.*, vol. 128, Jan. 2021, Art. no. 104089.
[69] I. D. Mienye and Y. Sun, "Effective feature selection for improved prediction of heart disease," in *Pan-African Artificial Intelligence and Smart Systems*, T. M. N. Ngatched and I. Woungang, Eds. Cham, Switzerland: Springer, 2022, pp. 94–107.
[70] O. Gefeller, A. Gul, F. Horn, Z. Khan, B. Lausen, and W. Adler, "Ensemble pruning for glaucoma detection in an unbalanced data set," *Methods Inf. Med.*, vol. 55, no. 6, pp. 557–563, 2016.
[71] I. D. Mienye, G. Obaido, K. Aruleba, and O. A. Dada, "Enhanced prediction of chronic kidney disease using feature selection and boosted classifiers," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2021, pp. 527–537.
[72] I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data," *Informat. Med. Unlocked*, vol. 25, Jan. 2021, Art. no. 100690.
[73] Z. Khan, A. Gul, A. Perperoglou, M. Miftahuddin, O. Mahmoud, W. Adler, and B. Lausen, "Ensemble of optimal trees, random forest and random projection ensemble classification," *Adv. Data Anal. Classification*, vol. 14, no. 1, pp. 97–116, Mar. 2020.
[74] V. García, A. I. Marqués, and J. S. Sánchez, "Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction," *Inf. Fusion*, vol. 47, pp. 88–101, May 2019.
[75] N. Arora and P. D. Kaur, "A bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105936.
[76] G. Yao, X. Hu, T. Zhou, and Y. Zhang, "Enterprise credit risk prediction using supply chain information: A decision tree ensemble model based on the differential sampling rate, synthetic minority oversampling technique and AdaBoost," *Expert Syst.*, vol. 39, no. 6, Jul. 2022, Art. no. e12953.
[77] W. Liu, H. Fan, and M. Xia, "Credit scoring based on tree-enhanced gradient boosting decision trees," *Expert Syst. Appl.*, vol. 189, Mar. 2022, Art. no. 116034.
[78] T. M. Alam, K. Shaukat, I. A. Hameed, S. Luo, M. U. Sarwar, S. Shabbir, J. Li, and M. Khushi, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020.
[79] J. T. Hancock and T. M. Khoshgoftaar, "Gradient boosted decision tree algorithms for medicare fraud detection," *Social Netw. Comput. Sci.*, vol. 2, no. 4, Jul. 2021, Art. no. 268.
[80] Y. Wang, Y. Zhang, Y. Lu, and X. Yu, "A comparative assessment of credit risk model based on machine learning—A case study of bank loan data," *Proc. Comput. Sci.*, vol. 174, pp. 141–149, Jan. 2020.
[81] M. Seera, C. P. Lim, A. Kumar, L. Dhamotharan, and K. H. Tan, "An intelligent payment card fraud detection system," *Ann. Oper. Res.*, vol. 334, nos. 1–3, pp. 445–467, Mar. 2024.
[82] A. Rawat, S. S. Aswal, S. Gupta, A. P. Singh, S. P. Singh, and K. C. Purohit, "Performance analysis of algorithms for credit card fraud detection," in *Proc. 2nd Int. Conf. Disruptive Technol. (ICDT)*, Mar. 2024, pp. 567–570.
[83] V. R. Adhegaonkar, A. R. Thakur, and N. Varghese, "Advancing credit card fraud detection through explainable machine learning methods," in *Proc. 2nd Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT)*, Jan. 2024, pp. 792–796.
[84] A. H. Nadim, I. M. Sayem, A. Mutsuddy, and M. S. Chowdhury, "Analysis of machine learning techniques for credit card fraud detection," in *Proc. Int. Conf. Mach. Learn. Data Eng. (iCMLDE)*, Dec. 2019, pp. 42–47.
[85] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93010–93022, 2019.
[86] S. Nijman, A. Leeuwenberg, I. Beekers, I. Verkouter, J. Jacobs, M. Bots, F. Asselbergs, K. Moons, and T. Debray, "Missing data is poorly handled and reported in prediction model studies using machine learning: A literature review," *J. Clin. Epidemiol.*, vol. 142, pp. 218–229, Feb. 2022.
[87] R. V. McCarthy, M. M. McCarthy, W. Ceccucci, and L. Halawi, "Predictive models using decision trees," in *Applying Predictive Analytics*. Cham, Switzerland: Springer, 2019, pp. 123–144.
[88] A. Mhasawade, G. Rawal, P. Roje, R. Raut, and A. Devkar, "Comparative study of SVM, KNN and decision tree for diabetic retinopathy detection," in *Proc. Int. Conf. Comput. Intell. Sustain. Eng. Solutions (CISES)*, Apr. 2023, pp. 166–170.
[89] T. Wang, R. Gault, and D. Greer, "Cutting down high dimensional data with fuzzy weighted forests (FWF)," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2022, pp. 1–8.
[90] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," *IEEE Access*, vol. 11, pp. 80348–80391, 2023.
[91] Y. Xia, "A novel reject inference model using outlier detection and gradient boosting technique in peer-to-peer lending," *IEEE Access*, vol. 7, pp. 92893–92907, 2019.

IBOMOIYE DOMOR MIENYE (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering and the M.Sc. degree (cum laude) in computer systems engineering from the University of East London, in 2012 and 2014, respectively, and the Ph.D. degree in electrical and electronic engineering from the University of Johannesburg, South Africa. His research interests include machine learning and deep learning for finance and healthcare applications.

NOBERT JERE received the M.Sc. and Ph.D. degrees in computer science from the University of Fort Hare, South Africa, in 2009 and 2013, respectively. He is currently an Associate Professor with the Department of Information Technology, Walter Sisulu University, South Africa. He has authored or coauthored numerous peer-reviewed journal articles and conference proceedings. His main research interest include ICT for sustainable development. He serves as a reviewer for numerous reputable journals. He has chaired/co-chaired international conferences.