IEEE Open Journal of the Computer Society

Received 18 March 2024; accepted 25 April 2024. Date of publication 30 April 2024;
date of current version 9 May 2024. The review of this article was arranged by Associate Editor Celimuge Wu.

Digital Object Identifier 10.1109/OJCS.2024.3394928

Multi-Rules Mining Algorithm for Combinatorially Exploded Decision Trees With Modified Aitchison-Aitken Function-Based Bayesian Optimization

YUTO OMAE, MASAYA MORI, AND YOHEI KAKIMOTO

College of Industrial Technology, Nihon University, Chiba 275-8575, Japan

CORRESPONDING AUTHOR: YUTO OMAE (e-mail: oomae.yuuto@nihon-u.ac.jp).

This work was supported by JSPS Grant-in-Aid for Scientific Research (C) under Grant 23K11310.
This article has supplementary downloadable material available at https://doi.org/10.1109/OJCS.2024.3394928, provided by the authors.

ABSTRACT Decision trees offer the benefit of easy interpretation because they allow the classification of input data based on if–then rules. However, as decision trees are constructed by an algorithm that achieves clear classification with minimum necessary rules, the trees possess the drawback of extracting only minimum rules, even when various latent rules exist in data. Approaches that construct multiple trees using randomly selected feature subsets do exist. However, the number of trees that can be constructed remains at the same scale because the number of feature subsets is a combinatorial explosion. Additionally, when multiple trees are constructed, numerous rules are generated, of which several are untrustworthy and/or highly similar. Therefore, we propose "MAABO-MT" and "GS-MRM" algorithms that strategically construct trees with high estimation performance among all possible trees with small computational complexity and extract only reliable and non-similar rules, respectively. Experiments are conducted using several open datasets to analyze the effectiveness of the proposed method. The results confirm that MAABO-MT can discover reliable rules at a lower computational cost than other methods that rely on randomness. Furthermore, the proposed method is confirmed to provide deeper insights than single decision trees commonly used in previous studies. Therefore, MAABO-MT and GS-MRM can efficiently extract rules from combinatorially exploded decision trees.

INDEX TERMS Data mining, decision tree, Bayesian optimization, Aitchison-Aitken kernel.

I. INTRODUCTION

A decision tree is one of the supervised learning methods for classifying input data [1]. It offers the advantage of easy interpretation by analysts as data classification is based on a simple if–then rule, making it effective for acquiring knowledge from data [2] and promoting application to diverse domains [3], [4], [5]. Decision trees are constructed by optimizing the Gini index [2], [6] or information gain [1] and acquiring the minimum number of necessary rules for data classification, thereby proving consistency with the principle of Occam's Razor [7]. However, only a small fraction of the large number of latent rules in data can be extracted. For example, the multiple decision trees constructed in previous studies [8], [9], [10] to predict the survival of passengers on the Titanic differ in their tree structure, even though the same data set was used. This is probably due to differences in data preprocessing and/or the algorithms used to construct the trees. This indicates that although there are several rules latent in the data, a single decision tree construction algorithm can only extract a portion of them. A method to overcome the aforementioned problem includes multiple decision tree construction. For instance, multiple decision trees can be constructed using randomly selected features such as random forest (RF) [11]. Several studies have analyzed the internal structure of RF [12], [13], [14] and extracted multiple rules through its application. However, as the number of combinations of randomly selected features is enormous, the constructed decision trees can be inappropriate. Additionally, decision trees with appropriate

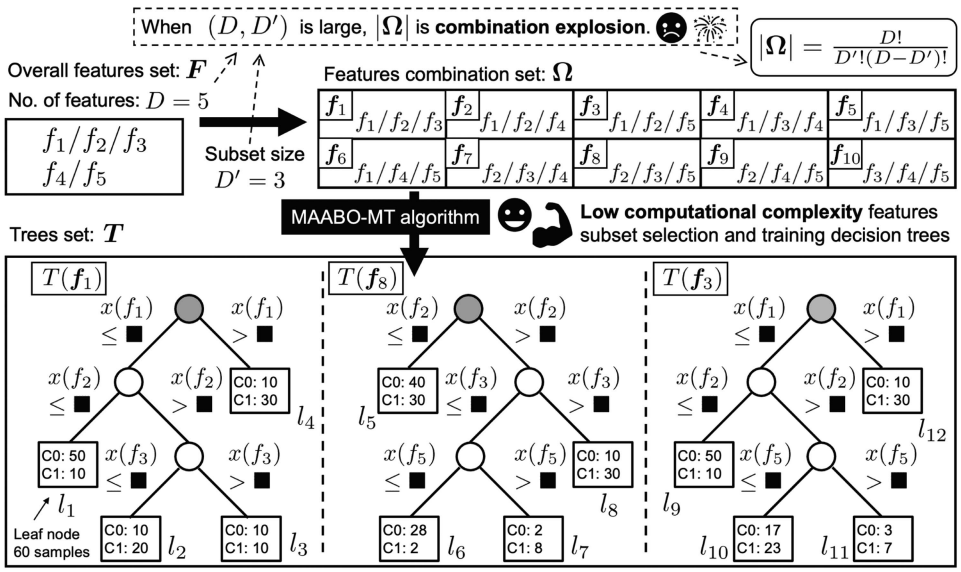

FIGURE 1. Input–output relationship of MAABO-MT. Here, the FCS $\Omega = \{f_1, \ldots, f_{10}\}$ is constructed by considering three out of five features $F = \{f_1, \ldots, f_5\}$, where $f_1 = \{f_1, f_2, f_3\}, \ldots, f_{10} = \{f_3, f_4, f_5\}$. Thereafter, only the three decision trees $T = \{T(f_1), T(f_3), T(f_8)\}$ that are expected to have high estimation performance are constructed. Notably, if the number of features D is large, the size of the FCS $|\Omega|$ is assumed to explode and the construction of all trees will not be possible (see (3)). Therefore, MAABO-MT is used to solve this problem.

rules can be overlooked before construction. Therefore, the strategical construction of multiple decision trees with good performance using methods with low computational costs is necessary, and random selection should be avoided. Additionally, decision trees can adopt meaningless noise features owing to parameter optimization (Appendix 1 in Supplemental text, available online). Therefore, extracting rules from decision trees using randomly selected features is perilous.

This article proposes an algorithm to solve the aforementioned existing issues. The input–output relationship is shown in Fig. 1. Three out of five features $f_1, \ldots, f_5$ are considered to construct decision trees. Since three out of five give ten combinations, we define $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{10}$. Ten decision trees can be constructed, but some trees may not perform satisfactorily. Therefore, the example in Fig. 1 uses only three constructed decision trees that are expected to perform well. On a small scale, all ten decision trees can be constructed without searching. However, doing so is impossible when the number of features exceeds a certain threshold. A set comprising D features can be defined as

$$\boldsymbol{F} = \{f_1, \ldots, f_D\}, \tag{1}$$

where $f_i \in \boldsymbol{F}$ represents a feature identifier such as "age." Vectors and matrices did not appear in this study, but they do exist in several other sets; therefore, bold font is used to denote sets. Let $\boldsymbol{f}$ be the feature subset obtained by extracting D' features from an overall feature set $\boldsymbol{F}$. Then, the features combination set (FCS) is defined by

$$\boldsymbol{\Omega} = \{\boldsymbol{f} \mid \boldsymbol{f} \subset \boldsymbol{F} \wedge |\boldsymbol{f}| = D'\},\ D > D', \tag{2}$$

where the element $\boldsymbol{f}$ of $\boldsymbol{\Omega}$ is an unordered set. Particularly, when $D' = 3$, $\boldsymbol{f} \in \boldsymbol{\Omega}$ is a feature combination such that $\boldsymbol{f}_1 = \{f_1, f_2, f_3\}$ and $\boldsymbol{f}_2 = \{f_1, f_2, f_4\}$. Additionally, $\{f_1, f_2, f_3\}$ and $\{f_3, f_2, f_1\}$ are the same elements and are not treated separately because $\boldsymbol{f}$ is an unordered set. Therefore, when $(D, D') = (5, 3)$, FCS becomes $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{10}\} \in \boldsymbol{\Omega}$, as shown in the upper right section of Fig. 1. Since $\boldsymbol{\Omega}$ comprises subsets obtained by extracting D' features from D features, its size is given by

$$|\boldsymbol{\Omega}| = {}_D C_{D'} = \frac{D!}{D'!(D-D')!}. \tag{3}$$

In Fig. 1, the size is $|\boldsymbol{\Omega}| = 5!/(5-3)! = 10$ as $(D, D') = (5, 3)$. However, when D is large, the construction of all decision trees become challenging. For example, when $(D, D') = (100, 5)$, the size is $|\boldsymbol{\Omega}| \simeq 7 \times 10^7$. Therefore, an efficient search for feature subsets that yield high-performance decision trees from $\boldsymbol{\Omega}$ is necessary.

The proposed algorithm uses Bayesian optimization [15] for searching high-performance solutions using nonparametric probability distribution. Particularly, Gaussian and Aitchison-Aitken (AA) kernels are used for solutions comprising real numbers and categorical values, respectively [16], [17]. Nevertheless, since this study focuses feature subset search, Bayesian optimization cannot be applied using the aforementioned kernel functions. Therefore, we propose a framework for direction application of Bayesian optimization to subset search, by using a new modified function of the AA kernel "modified AA (MAA)." Thus, the proposed algorithm is called "MAA function-based Bayesian optimization for making trees (MAABO-MT)."

At a low computational cost, high-performance decision trees are constructed using the proposed method; however, only appropriate rules should be extracted because some leaf

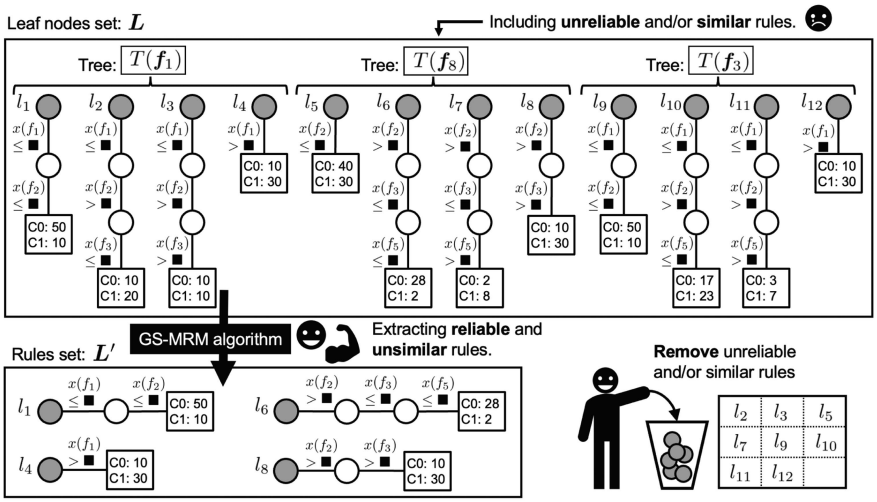

FIGURE 2. Input–output relationship of GS-MRM. Among the total 12 leaf nodes obtained from the three decision trees shown in Fig. 1, only reliable and dissimilar rules are extracted.

nodes have small sample sizes, whereas others do not have clearly separated classes. The number of reliable leaf nodes is not numerous, some of which are similar, thereby making proper academic discussion from all leaf nodes difficult. Therefore, we propose an algorithm to extract leaf nodes such that the following conditions are satisfied. 1) The sample size is sufficient, 2) classes are clearly divided, and 3) leaf nodes are not similar to the previously extracted ones. The input–output relationship of this algorithm is shown in Fig. 2. Herein, four leaf nodes are extracted from a total of 12 leaf nodes $l_1, \ldots, l_{12}$. C0 and C1 represent the class labels and the numbers in the square boxes represent the sample sizes. Initially, l_7 and l_{11} are automatically removed owing to their small sample size. Thereafter, l_5 and l_{10} are automatically removed as no clearly separated classes exist. Additionally, for similar rules such as l_1 and l_9, only one rule can be adopted. Therefore, the removals result in an output of only trusted leaf nodes (l_1, l_4, l_6, l_8). Since the method combines Gini index [18] and Simpson coefficient [19], we termed it as "Gini and Simpson coefficients-based multi-rules mining algorithm (GS-MRM algorithm)."

The key contributions of this study are listed below.

- From a set of decision trees that can be constructed in large numbers, a new algorithm is proposed to construct only decision trees that possess good rules at a small computational cost.
- A new algorithm is proposed to extract only reliable and non-similar leaf nodes from the large number of leaf nodes of the constructed decision trees.
- These algorithms are shown to be useful in extracting a large number of rules from a small number of decision trees.

II. RELATED WORKS ON DECISION TREE

Decision trees are one of the most commonly used data-mining methods [20], with several proposed algorithms such as ID3 [1], C4.5 [21], CHAID [22], and CART [23], [24], which use all features to construct a single decision tree. Decision trees can improve the estimation performance through ensemble learning. For example, the approach of constructing multiple decision trees using bootstrapping called RF [11] has been applied in various domains [25], [26], [27]. Additionally, some studies have analyzed the internal structure of RF [12], [13], [14].

A gradient boosting decision tree [28], such as XG-boost [29] and LightGBM [30], has been proposed using ensemble learning. Compared to random approaches, the aforementioned methods are more strategic in terms of error reduction and expected to perform better than RF. Huang et al. [31] and Joharestani et al. [32] reported a better performance of XG-boost relative to RF. Additionally, some studies reported a superior estimation performance of LightGBM than that of RF [33], [34].

As aforementioned, decision trees are evolving and the development of ensemble learning is particularly remarkable. Since previous studies mainly focused on improving estimation performance, our study focuses on rule mining and not on performance estimation. There are several methods for rule mining, but since this study focuses on tree structures, we will introduce rule mining with trees. To extract a large number of tree structured rules that are latent in the data, it is necessary to construct a large number of decision trees by limiting the number of features to be used, as described in Section I. Hence, a method is needed to measure the importance of the features in order to select them. The aforementioned method of constructing multiple trees can be used to measure feature importance (see RF [35], [36], XG-boost [37], and LightGBM [38]). Prior studies have used tree-based feature importance to obtain novel insights in a variety of domains. For examples, Venkateswarlu et al. [39] analyzed the relationship between land use factors and water quality

Algorithm 1: MAABO-MT Algorithm.

Input: Overall features set $\boldsymbol{F}$, features subset size D', initial solution size N_{I}, split coefficient α, iteration of Bayesian optimization N_{B}, maximum tree's depth $p_{\max}$, distribution degree to mismatches h, damping coefficient b, extracting single feature size N_{U}, sampling size N_{E}

Output: Trees set $\boldsymbol{T}$

1: Creating FCS $\boldsymbol{\Omega}$ based on D' and $\boldsymbol{F}$
2: Creating U-FCS: $\boldsymbol{F}' \leftarrow \boldsymbol{\Omega}$
3: Initializing V-FCS: $\overline{\boldsymbol{F}'} \leftarrow \varnothing$
4: Initializing trees set: $\boldsymbol{T} \leftarrow \varnothing$
5: **for** $n = 1$ **to** N_{I} **do**
6: Randomly selecting a features subset: $\boldsymbol{f} \in \boldsymbol{F}'$
7: Optimal depth: $p^* \leftarrow \mathrm{argmax}_{p\in\{1,\ldots,p_{\max}\}} S(\boldsymbol{f}, p)$
8: Updating trees set: $\boldsymbol{T} \leftarrow \boldsymbol{T} \cup \{T(\boldsymbol{f}, p^*)\}$
9: Updating V-FCS: $\overline{\boldsymbol{F}'} \leftarrow \overline{\boldsymbol{F}'} \cup \{\boldsymbol{f}\}$
10: Updating U-FCS: $\boldsymbol{F}' \leftarrow \boldsymbol{F}' \setminus \{\boldsymbol{f}\}$
11: **end for**
12: **for** $n = 1$ **to** N_{B} **do**
13: Descended sorting based on validation score $\overline{\boldsymbol{F}'}$
14: Split threshold: $N_{\mathrm{th}} \leftarrow \lfloor \alpha |\overline{\boldsymbol{F}'}| \rfloor$
15: Creating H-FCS $\boldsymbol{U}^+$ and L-FCS $\boldsymbol{U}^-$ with $(N_{\mathrm{th}}, \overline{\boldsymbol{F}'})$
16: Sampling $\boldsymbol{F}'' \leftarrow H(\boldsymbol{F}', \boldsymbol{U}^+, N_{\mathrm{U}}, N_{\mathrm{E}})$
17: Probability: $p(\boldsymbol{f}|\boldsymbol{U}^i) \leftarrow K(\boldsymbol{f}, \boldsymbol{U}^i, h, b),\ i \in \{+, -\}$
18: Next subset: $\boldsymbol{f}^* \leftarrow \mathrm{argmax}_{\boldsymbol{f}\in\boldsymbol{F}''} p(\boldsymbol{f}|\boldsymbol{U}^+)/p(\boldsymbol{f}|\boldsymbol{U}^-)$
19: Optimal depth: $p^* = \mathrm{argmax}_{p\in\{1,\ldots,p_{\max}\}} S(\boldsymbol{f}^*, p)$
20: Updating trees set: $\boldsymbol{T} \leftarrow \boldsymbol{T} \cup \{T(\boldsymbol{f}^*, p^*)\}$
21: Updating V-FCS: $\overline{\boldsymbol{F}'} \leftarrow \overline{\boldsymbol{F}'} \cup \{\boldsymbol{f}^*\}$
22: Updating U-FCS: $\boldsymbol{F}' \leftarrow \boldsymbol{F}' \setminus \{\boldsymbol{f}^*\}$
23: **end for**
24: **return** Trees set $\boldsymbol{T}$

using RF feature importance. Jabeur et al. [40] conducted a factor analysis of corporate bankruptcy using XG-boost feature importance. Li et al. [41] conducted a factor analysis of pregnancy outcomes after in vitro fertilization using LightGBM feature importance.

While these approaches are important, they only measure the importance of each individual feature and may overlook features that are more effective in combination. Therefore, even if decision trees were constructed using the features selected by the methods described above, it is likely that only a fraction of the rules would be extracted. To solve this problem, it is necessary to consider the effects of feature combinations, but this leads to a significant increase in computational cost. Therefore, this study proposes an algorithm that can strategically discover rules by considering feature combinations with a small computational cost.

III. PROPOSED METHOD

A. MAABO-MT ALGORITHM

The MAABO-MT algorithm shown in Fig. 1 is a partial modification of Bayesian optimization [15], [16], an algorithm for extracting feature subsets from FCS $\boldsymbol{\Omega}$ that leads to high performance decision trees. The detailed procedure is presented in Algorithm 1. In addition, Algorithm 1 and the equations in this section are fully corresponding.

First, the initialization process shown in lines 1–4 of Algorithm 1 is explained. Initially, the following sets are defined.

- Unverified features combination set (U-FCS): $\boldsymbol{F}' = \boldsymbol{\Omega}$
- Verified features combination set (V-FCS): $\overline{\boldsymbol{F}'} = \varnothing$
- Trees set: $\boldsymbol{T} = \varnothing$

Herein, $\boldsymbol{F}'$ is a set comprising feature subsets for which estimation performance has not been verified and is initialized by $\boldsymbol{\Omega}$, while $\overline{\boldsymbol{F}'}$ represents a set comprising feature subsets where the estimation performance has been verified and initialized with the empty set $\varnothing$. Since MAABO-MT is an algorithm that constructs decision trees, the decision tree set is initialized with an empty set $\boldsymbol{T} = \varnothing$.

Next, the initial solution generation process shown in lines 5–11 of Algorithm 1 is described. Initially, we randomly construct N_{I} trees. Based on these validation performances, the next step involves feature exploration for tree construction. Therefore, N_{I} denotes the initial solution size. Training data are used to construct decision trees, validation data are used to measure and tune the validation performance and hyperparameter (maximum depth of trees), respectively, and no overlap exists between the two datasets.

First, the generation and evaluation of initial solutions are described. A feature subset $\boldsymbol{f}$ is randomly selected from U-FCS $\boldsymbol{F}'$, using which the maximum tree depth that leads to maximum verification performance is determined as

$$p^* = \underset{p\in\{1,\ldots,p_{\max}\}}{\mathrm{argmax}} S(\boldsymbol{f}, p), \tag{4}$$

where $S(\boldsymbol{f}, p)$ is the validation performance of the decision tree constructed by maximum depth p, feature subset $\boldsymbol{f}$, and the CART algorithm [24]. $p_{\max}$ denotes the optimal maximum depth. The performance index is the macro-ave. F1 score. The aforementioned process is performed to prevent overfitting. Using the optimal maximum depth p^* and a feature subset $\boldsymbol{f}$, we construct a decision tree $T(\boldsymbol{f}, p^*)$ that is added to the decision tree set as

$$\boldsymbol{T} = \boldsymbol{T} \cup \{T(\boldsymbol{f}, p^*)\}. \tag{5}$$

Since the aforementioned procedure validates the feature subset $\boldsymbol{f}$, V-FCS and U-FCS are updated as

$$\overline{\boldsymbol{F}'} = \overline{\boldsymbol{F}'} \cup \{\boldsymbol{f}\},\quad \boldsymbol{F}' = \boldsymbol{F}' \setminus \{\boldsymbol{f}\}. \tag{6}$$

By implementing the procedure N_{I} times, a set $\boldsymbol{T}$ comprising N_{I} decision trees is constructed. Therefore, the initial solution is generated following the aforementioned procedure.

Next, the process by which to construct the high-performance decision trees, as shown in lines 12–23 of Algorithm 1, is described. In V-FCS $\overline{\boldsymbol{F}'}$, some feature subsets have a high validation performance, while others have a low performance. For clarification, V-FCS $\overline{\boldsymbol{F}'}$ is sorted in descending order of verification performance, making it an ordered

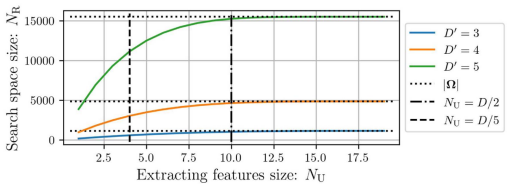

FIGURE 3. Relationships between extracting features N_U and search space size N_R.

set, such that

$$\overline{\boldsymbol{F}'} = \{\overline{\boldsymbol{f}_1}, \overline{\boldsymbol{f}_2}, \cdots\}^*, \quad \overline{\boldsymbol{f}_1} \succcurlyeq \overline{\boldsymbol{f}_2} \succcurlyeq \cdots, \tag{7}$$

where "$\succcurlyeq$" represents an ordered relationship based on verification performance and $\{\cdot\}^*$ represents an ordered set. Then, let $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$ denote the sets comprising the upper and lower elements of $\overline{\boldsymbol{F}'}$, respectively, such that

$$\begin{aligned} \boldsymbol{U}^+ &= \{\overline{\boldsymbol{f}_n} \mid n = 1, \ldots, N_{\text{th}}\}, \\ \boldsymbol{U}^- &= \{\overline{\boldsymbol{f}_n} \mid n = N_{\text{th}} + 1, \ldots, |\overline{\boldsymbol{F}'}|\}, \\ \overline{\boldsymbol{F}'} &= \boldsymbol{U}^+ \cup \boldsymbol{U}^-, \quad N_{\text{th}} = \lfloor \alpha |\overline{\boldsymbol{F}'}| \rfloor, \quad \alpha \in (0, 1) \end{aligned} \tag{8}$$

where α denotes the split coefficient. $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$ are known as high and low-score feature combination sets (H-FCS and L-FCS), respectively. Using $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$, the next feature subset to be validated is searched from U-FCS $\boldsymbol{F}'$. However, the size of U-FCS $|\boldsymbol{F}'|$ explodes when D and D' are large, as shown in (3). Therefore, the search space should be reduced.

The individual features used in each subset within $\boldsymbol{U}^+$ can contribute to higher performance. Therefore, the number of individual features $f_1, \ldots, f_D \in \boldsymbol{F}$ present in $\boldsymbol{U}^+$ are counted, and the top N_U features that are most frequently used are extracted. Subsequently, the subsets in U-FCS $\boldsymbol{F}'$ using at least one of the extracted features are chosen as a search space. Herein, the number of unselected features is denoted as $D - N_U$. Therefore, the search space is reduced by the number of feature subsets obtained by extracting D' features from $D - N_U$ features. Therefore, the reduction in the search space size is given by ${}_{D-N_U}C_{D'}$, and the reduced search space size is expressed as

$$N_R = |\boldsymbol{\Omega}| - {}_{D-N_U}C_{D'}. \tag{9}$$

Since ${}_xC_y = 0$ $(x < y)$, $N_U \leq D - D'$ is required to reduce the search space. The relationship between N_U and N_R is shown in Fig. 3. The obtained result $N_R < |\boldsymbol{\Omega}|$ indicates the reduction in search space. Particularly, when $N_U = D/5$, the search space is reduced by approximately half.

Smaller values of N_U reduce the search space; however, extremely small values can result in a localized search. Therefore, setting N_U to an extremely small value is undesirable. To perform search at a low computational cost when N_U is not too small, we randomly select N_E feature subsets from N_R candidates. Then, the next feature subset to be verified is selected from among the N_E subsets. In other words, N_U and N_E reduce the search space in this algorithm. Thus, search space obtained using the aforementioned procedure can be defined as

$$\boldsymbol{F}'' = H(\boldsymbol{F}', \boldsymbol{U}^+, N_U, N_E). \tag{10}$$

where H is a function that takes the top N_U single features that are frequently present in $\boldsymbol{U}^+$ and randomly selects N_E subsets containing the top features from $\boldsymbol{F}'$. The smaller the size of N_E, the lower is the computational cost. However, if N_E is too small, the subset to be adopted depends on random luck.

Next, we consider the feature subset selection from $\boldsymbol{F}''$ to construct a high-performance decision tree. Assuming two distributions to estimate the probabilities that $\boldsymbol{f} \in \boldsymbol{F}''$ belongs to $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$, we have

$$p(\boldsymbol{f}|\boldsymbol{U}^+), \quad p(\boldsymbol{f}|\boldsymbol{U}^-), \quad \boldsymbol{f} \in \boldsymbol{F}''. \tag{11}$$

Here, the features subset $\boldsymbol{f} \in \boldsymbol{F}''$ with a larger $p(\boldsymbol{f}|\boldsymbol{U}^+)$ and smaller $p(\boldsymbol{f}|\boldsymbol{U}^-)$ should be selected as the next features subset $\boldsymbol{f}^* \in \boldsymbol{F}''$, such that

$$\boldsymbol{f}^* = \underset{\boldsymbol{f} \in \boldsymbol{F}''}{\text{argmax}} \frac{p(\boldsymbol{f}|\boldsymbol{U}^+)}{p(\boldsymbol{f}|\boldsymbol{U}^-)}. \tag{12}$$

Then, using the feature subset $\boldsymbol{f}^*$, 4, 5, and 6 are processed, while the sets $\boldsymbol{T}, \overline{\boldsymbol{F}'}, \boldsymbol{F}'$ are updated. When the aforementioned process is repeated iteratively, high-performance trees are strategically constructed. If the number of iterations is N_B, the final number of trees constructed is $N_B + N_I$, using N_I number of initial solutions. The procedures discussed thus far form MAABO-MT.

B. PROBABILITY DISTRIBUTION WITH MODIFIED AITCHISON-AITKEN FUNCTION

This subsection describes the method for constructing probability distributions $p(\boldsymbol{f}|\boldsymbol{U}^i), i \in \{+, -\}$. When the search target is a categorical vector, the AA kernel [42] is used to construct the probability distribution. However, the AA kernel cannot be used in our case because $\boldsymbol{f}$ is a set of features. Therefore, this article proposes a modified AA function (MAA).

1) MODIFIED AITCHISON-AITKEN FUNCTION

As indicated in (12), the next $\boldsymbol{f}^*$ to be selected should be similar to H-FCS $\boldsymbol{U}^+$ and dissimilar to L-FCS $\boldsymbol{U}^-$. Therefore, a function $k(\boldsymbol{f}, \boldsymbol{u})$ that measures the similarity between the two feature subsets $\boldsymbol{f}$ and $\boldsymbol{u} \in \boldsymbol{U}^{i \in \{+,-\}}$ should be created.

Since $k(\boldsymbol{f}, \boldsymbol{u})$ is used to construct the probability distributions, we aim to satisfy the following constraints:

- (Constraint 1) $\sum_{\boldsymbol{f} \in \boldsymbol{\Omega}} k(\boldsymbol{f}, \boldsymbol{u}) = 1, \ \forall \boldsymbol{u} \in \boldsymbol{U}^{i \in \{+,-\}}$
- (Constraint 2) $k(\boldsymbol{f}, \boldsymbol{u}) \in [0, 1]$

Moreover, as $k(\boldsymbol{f}, \boldsymbol{u})$ is a function for measuring similarity, we aim to satisfy the following constraint:

- (Constraint 3) $D' - |\boldsymbol{f}_v \cap \boldsymbol{u}| > D' - |\boldsymbol{f}_w \cap \boldsymbol{u}| \Leftrightarrow k(\boldsymbol{f}_v, \boldsymbol{u}) < k(\boldsymbol{f}_w, \boldsymbol{u}), \quad \boldsymbol{f}_v, \boldsymbol{f}_w \in \boldsymbol{\Omega}$

Because the feature subsets $\boldsymbol{f}$ and $\boldsymbol{u}$ are sets comprising D' features, the number of mismatches is given by

$$m = D' - |\boldsymbol{f} \cap \boldsymbol{u}|, \quad m = 0, \ldots, D'. \tag{13}$$

Therefore, $k_0, \ldots, k_{D'}$ are defined as the values of $k(\boldsymbol{f}, \boldsymbol{u})$ when the number of mismatches is $m = 0, \ldots, D'$. Next, the values that should be assigned to $k_0, \ldots, k_{D'}$ to satisfy Constraints 1, 2, and 3 are considered.

First, $k(\boldsymbol{f}, \boldsymbol{u})$ based on the number of mismatches $m = 0$ is defined as

$$k_0 = 1 - h, \ h \in [0, 1]. \tag{14}$$

As a similarity measure between $\boldsymbol{f}$ and $\boldsymbol{u}$, k_0 must be large when the number of mismatches $m = 0$. Thus, $k_0 = 1$ is not possible from Constraint 1 because values must also be assigned for cases where the number of mismatches $m = 1, \ldots, D'$. Thus, the distribution amount for $m \geq 1$ is h.

Next, the usage of the distributed h is considered depending on the number of feature subsets with one or more mismatches in $\boldsymbol{\Omega}$. Therefore, the number of feature subsets that include m mismatches is defined as a_m. From Constraint 1, k_m must satisfy

$$a_0 k_0 + a_1 k_1 + \cdots + a_{D'} k_{D'} = 1. \tag{15}$$

Here, $a_0 = 1$ because m is zero only when $\boldsymbol{f}$ and $\boldsymbol{u}$ are the same. Additionally, by substituting (14) into (15), we obtain

$$a_1 k_1 + \cdots + a_{D'} k_{D'} = h. \tag{16}$$

Next, to satisfy Constraint 3,

$$k_m = b^{m-1} k_1, \ b \in (0, 1), \ m = 1, \ldots, D' \tag{17}$$

is set up, where b is the damping coefficient. Since k_m is the value of k_1 damped by b^{m-1}, when $b \in (0, 1)$, $k_1 > k_2 > \cdots > k_{D'}$ is satisfied. Additionally, if k_1 is clarified, the $k_2, \ldots, k_{D'}$ can be determined. Therefore, to clarify k_1, (17) is substituted into (16) to obtain

$$k_1 = \frac{h}{a_1 b^0 + \cdots + a_{D'} b^{D'-1}}. \tag{18}$$

By substituting the results into (17), we have

$$k_m = \frac{b^{m-1} h}{\sum_{i=1}^{D'} a_i b^{i-1}}, \ m = 1, \ldots, D'. \tag{19}$$

As h and b are hyper-parameters, the unknown variable is only a_i. Since a_i is the total number of feature subsets with i mismatches between $\boldsymbol{u}$ and $\boldsymbol{f} \in \boldsymbol{\Omega}$, its values is obtained by multiplying the following two terms,

- (1) the number of combinations obtained by selecting $D' - i$ features from among the features contained in $\boldsymbol{u}$
- (2) the number of combinations obtained by selecting i features from the features not included in $\boldsymbol{u}$

As $\boldsymbol{u}$ comprises D' features, term (1) is ${}_{D'}C_{D'-i}$. Since the number of features not included in $\boldsymbol{u}$ is $D - D'$, term (2) is denoted as ${}_{D-D'}C_i$. Therefore, a_i is obtained as

$$a_i = ({}_{D'}C_{D'-i})({}_{D-D'}C_i), \tag{20}$$

where ${}_xC_0 = 1$ and ${}_xC_y = 0$, $(x < y)$. Upon substituting into (19), we obtain

$$k_m = \frac{b^{m-1} h}{\sum_{i=1}^{D'} ({}_{D'}C_{D'-i})({}_{D-D'}C_i) b^{i-1}}, m = 1, \ldots, D'. \tag{21}$$

Thus, the specific values of $k_0, \ldots, k_{D'}$ are determined.

Based on (13), (14), and (21), we propose

$$k(\boldsymbol{f}, \boldsymbol{u}, h, b) = \begin{cases} 1 - h, & \text{if } D' - |\boldsymbol{f} \cap \boldsymbol{u}| = 0 \\ \dfrac{b^{D'-|\boldsymbol{f}\cap\boldsymbol{u}|-1}}{\sum_{i=1}^{D'} ({}_{D'}C_{D'-i})({}_{D-D'}C_i) b^{i-1}} h, & \text{if } D' - |\boldsymbol{f} \cap \boldsymbol{u}| > 0 \end{cases}, \quad h \in [0, 1], \ \ b \in (0, 1), \tag{22}$$

as the similarity function between feature subsets $\boldsymbol{f}$ and $\boldsymbol{u}$. The function is designed satisfy the Constraints 1, 2, and 3 and is therefore applicable as a discrete probability distribution. $k(\boldsymbol{f}, \boldsymbol{u}, h, b)$ is a proposed function called MAA. The design for distributing $1 - h$ in the case of a match, and h in the case of a mismatch also appears in the AA kernel [42]. The relationships between $k(\boldsymbol{f}, \boldsymbol{u}, h, b)$, m, a_m are discussed in Appendix 2 (Supplemental text, available online).

2) PROBABILITY DISTRIBUTION

This subsection describes the process of formulating the probability $p(\boldsymbol{f}|\boldsymbol{U}^i)$, where a feature subset $\boldsymbol{f} \in \boldsymbol{\Omega}$ belongs to $\boldsymbol{U}^i$, using MAA function $k(\boldsymbol{f}, \boldsymbol{u}, h, b)$.

As the MAA function is calculated for a single $\boldsymbol{u} \in \boldsymbol{U}^i$, $k(\boldsymbol{f}, \boldsymbol{u})$ is calculated for all $\boldsymbol{u} \in \boldsymbol{U}^i$ and averaged as

$$K(\boldsymbol{f}, \boldsymbol{U}^i, h, b) = \frac{1}{|\boldsymbol{U}^i|} \sum_{\boldsymbol{u} \in \boldsymbol{U}^i} k(\boldsymbol{f}, \boldsymbol{u}, h, b), \tag{23}$$

where, $i \in \{+, -\}$. The MAA function satisfies the definition of probability distribution for the input $\boldsymbol{f} \in \boldsymbol{\Omega}$ because the probability of occurrence of all events is 1, such that

$$\begin{aligned} P(\boldsymbol{\Omega}) &= \sum_{\boldsymbol{f} \in \boldsymbol{\Omega}} K(\boldsymbol{f}, \boldsymbol{U}^i) \\ &= \frac{1}{|\boldsymbol{U}^i|} \sum_{\boldsymbol{u} \in \boldsymbol{U}^i} \sum_{\boldsymbol{f} \in \boldsymbol{\Omega}} k(\boldsymbol{f}, \boldsymbol{u}) \\ &= 1, \ \ \because \sum_{\boldsymbol{f} \in \boldsymbol{\Omega}} k(\boldsymbol{f}, \boldsymbol{u}) = 1. \end{aligned} \tag{24}$$

Additionally,

$$K(\boldsymbol{f}, \boldsymbol{U}^i) \in [0, 1], \ \ \forall \boldsymbol{f}, \ \ \because k(\boldsymbol{f}, \boldsymbol{u}) \in [0, 1] \tag{25}$$

is satisfied. Therefore, $K(\boldsymbol{f}, \boldsymbol{U}^i)$ denotes a discrete probability distribution. Based on the aforementioned discussion, we adopt $K(\boldsymbol{f}, \boldsymbol{U}^i, h, b)$ as $p(\boldsymbol{f}|\boldsymbol{U}^i)$, such that

$$p(\boldsymbol{f}|\boldsymbol{U}^i) = K(\boldsymbol{f}, \boldsymbol{U}^i, h, b), \ \ i \in \{+, -\}. \tag{26}$$

Consequently, (12) can be calculated, and MAABO-MT can be performed.

C. GS-MRM ALGORITHM

The MAABO-MT algorithm can be used to construct $N_I + N_B$ decision trees that are expected to exhibit a high verification performance. However, when considering a rule-mining algorithm, returning all leaf nodes as output to the user is inappropriate because only a limited number of leaf nodes belonging to the decision tree are reliable and some of the multiple leaf nodes are similar to each other. Therefore, we propose the GS-MRM algorithm as a method for extracting reliable and non-similar leaf nodes from a large number of leaf nodes of $N_I + N_B$ decision trees.

Initially, the set $\boldsymbol{L}$ comprising N_L leaf nodes in $N_I + N_B$ decision trees and the set $\boldsymbol{L}'$ comprising the leaf nodes that are provided as outputs to the users are defined as

$$\boldsymbol{L} = \{l_n \mid n = 1, \ldots, N_L\},\ \boldsymbol{L}' = \varnothing. \tag{27}$$

Herein, $\boldsymbol{L}'$ is initialized with an empty set because the leaf nodes to be returned have not yet been determined. The method for moving leaf nodes in $\boldsymbol{L}$ to $\boldsymbol{L}'$ is defined as GS-MRM. In this study, the moving target leaf node l^* is determined as

$$\begin{aligned} l^* = &\operatorname*{argmin}_{l \in \boldsymbol{L}} \left(g(l) + \max_{l' \in \boldsymbol{L}'} v(l, l') \right), \\ &\text{s.t.},\ g(l) < \gamma g_{\max},\ \ n(l) \geq \beta,\ \ \max_{l' \in \boldsymbol{L}'} v(l, l') < \delta, \\ &\gamma \in (0, 1],\ \ \delta \in (0, 1], \end{aligned} \tag{28}$$

where $g(l)$ is the Gini index of leaf node l and $g_{\max}$ is the maximum Gini index when the classes are evenly mixed. The rule with a lower Gini index is adopted in preference to other rules because a leaf node with a lower Gini index is a more reliable rule. Furthermore, only the leaf nodes below the threshold are adopted following the constraint $g(l) < \gamma g_{\max}$. If the total sample size of a leaf node is N, the sample size of class c_i is N_i, and the number of class labels is C, then, the Gini index is defined as

$$g = 1 - \sum_{i=1}^{C} \left(\frac{N_i}{N} \right)^2. \tag{29}$$

The same expression has been expressed in (1) of [18]. When the Gini index is close to zero, data are clearly classified; if the index is large, the classes are mixed. Since $N_i = N/C, \forall i$ is the most mixed state, by substituting its value into (29), we obtain

$$g_{\max} = 1 - \frac{1}{C}. \tag{30}$$

In the constraint, γ is a parameter used to adjust the threshold of the Gini index. Thus, by setting γ close to zero, only leaf nodes with clearly classified data are extracted. The constraint does not work properly with unbalanced class data; therefore, a weighted Gini index must be used, which is calculated by giving "balanced" option to the class weight argument of DecisionTreeClassifier [43] in scikit-learn. Alternatively, the data can be converted into balanced data through over or under sampling [44].

Algorithm 2: GS-MRM Algorithm.

Input: Trees set $\boldsymbol{T}$, threshold of leaf node samples β, threshold of Gini coefficient γ, threshold of Simpson coefficient δ, class label size C
Output: Rules set $\boldsymbol{L}'$

```
Creating leaf nodes set L based on the trees set T
Initializing rules set: L' ← ∅
Initial leaf node size: N_L ← |L|
for i = 1 to N_L do
    if g(l_i) ≥ γ (1 − 1/C) or n(l_i) < β then
        Removing leaf node: L ← L \ {l_i}
    end if
end for
while L ≠ ∅ do
    if L' = ∅ then
        Opt. leaf node: l* ← argmin_{l∈L} g(l)
    else
        Opt. leaf node:
        l* ← argmin_{l∈L}(g(l) + max_{l'∈L'} v(l, l'))
    end if
    if max_{l'∈L'} v(l*, l') < δ then
        Updating rules set: L' ← L' ∪ {l*}
    end if
    Updating leaf nodes set: L ← L \ {l*}
end while
return L'
```

The sample size of the leaf node l denoted as $n(l)$. Since leaf nodes comprising small samples are not reliable, we set the constraint that if the sample size is not greater than or equal to the threshold β, it cannot be adopted as a rule. $v(l, l')$ represents the similarity between the leaf node $l \in \boldsymbol{L}$ and the previously extracted leaf node $l' \in \boldsymbol{L}'$. Owing to the presence of multiple extracted leaf nodes, the maximum similarity is calculated, and leaf nodes with smaller similarities are prioritized. Additionally, the rules with high similarity to previously extracted ones should not be adopted. Therefore, the constraint $\max_{l' \in \boldsymbol{L}'} v(l, l') < \delta$, where δ is a parameter, is considered. The details of $v(l, l')$ are presented in Section III-D.

The sets of $\boldsymbol{L}'$ and $\boldsymbol{L}$ are updated at the leaf node selected using (28), such that

$$\boldsymbol{L}' = \boldsymbol{L}' \cup \{l^*\},\ \ \boldsymbol{L} = \boldsymbol{L} \setminus \{l^*\}. \tag{31}$$

Only reliable and dissimilar rules are extracted by repeating (28) and (31). The details of GS-MRM are presented as Algorithm 2.

TABLE 1. Feature Labels of Titanic Dataset

ID	Feature	Details
f_1	Pclass	Ticket class. 1st (best), 2nd, 3rd (worst).
f_2	Sex	Male or female.
f_3	Age	passenger's age.
f_4	SibSp	The number of siblings and spouses.
f_5	ParCh	The number of parents and child.
f_6	Fare	The fee of getting on the Titanic.
f_7	Embarked C	Departure port: Cherbourg. True/False.
f_8	Embarked Q	Departure port: Queenstown. True/False.
f_9	Embarked S	Departure port: Southampton. True/False.
f_{10}	Noise 1	Uniform random numbers from 0 to 1.
⋮	⋮	⋮
$f_{9+N_{\text{noise}}}$	Noise N_{noise}	Uniform random numbers from 0 to 1.

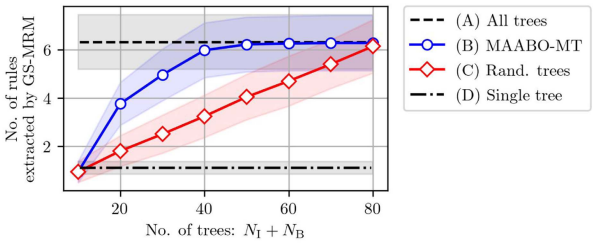

FIGURE 4. Number of extracted rules from decision trees constructed using (A)–(D) approaches. Each value is an average, and error bar represents 0.5 std. calculated from the results of 50 random seeds.

D. SIMILARITY BETWEEN LEAF NODES

To solve (28), we must design a function $v(l, l')$ that measures the similarity between two leaf nodes l and l'. The representation of leaf node l in the decision tree is given by

$$l : x(f_{i1}) > \blacksquare \wedge x(f_{i2}) \leq \blacksquare \wedge x(f_{i3}) > \blacksquare \wedge \cdots, \tag{32}$$

where the value of feature f_i is $x(f_i)$. In this study, a set comprising the smallest units of logic is defined as

$$\boldsymbol{R}_l = \{f_{i1} : \text{large}, f_{i2} : \text{small}, f_{i3} : \text{large}, \cdots\}. \tag{33}$$

Cases where the same logic appears in a leaf node do exist. In such cases, the logics are combined into a single logic for simplification. First, we construct $\boldsymbol{R}_{l'}$ for the leaf node l'. Thereafter, the similarity of the two sets $\boldsymbol{R}_l$ and $\boldsymbol{R}_{l'}$ is measured by Simpson coefficient [19], such that

$$v(l, l') = \frac{|\boldsymbol{R}_l \cap \boldsymbol{R}_{l'}|}{\min\{|\boldsymbol{R}_l|, |\boldsymbol{R}_{l'}|\}}. \tag{34}$$

The output of aforementioned function is 1 and 0 for the most and least similarity between the sets. Thus, solutions to the optimization problem expressed in (28) are possible.

IV. EXPERIMENT 1: EFFECTIVENESS OF MAABO-MT ON EXTRACTING RULES

A. OBJECTIVE AND OUTLINE

To verify the effectiveness of the proposed method, GS-MRM was applied to decision trees constructed using MAABO-MT for rule extraction. The Titanic dataset often used to evaluate machine-learning performance was adopted [45], [46], [47]. The data consisted of 1309 passenger details aboard Titanic, with class labels of dead (809 samples) and alive (500 samples). The features included in this dataset were $f_1, \ldots, f_9$, as listed in Table 1. Notably, noise features are also included in Table 1, but were not used in Experiment 1. They were used in Experiment 2 or later. In Experiment 1, the overall feature set was $\boldsymbol{F} = \{f_1, \ldots, f_9\}$. A feature subset size of $D' = 3$ was adopted to perform MAABO-MT, implying the selection of three features from the overall feature set comprising nine features to construct the FCS $\boldsymbol{\Omega}$. Herein, the number of decision trees that could be constructed was equal to the size of $\boldsymbol{\Omega}$; thus, ${}_9C_3 = 84$ from (3).

To verify the effectiveness of MAABO-MT, decision trees were constructed using the following four approaches: (A) "All trees": creating all 84 trees with all 84 features subsets in $\boldsymbol{\Omega}$. (B) "MAABO-MT": limited number of trees with features subset selected by MAABO-MT in $\boldsymbol{\Omega}$. (C) "Randomized trees": limited number of trees with features subset selected at random in $\boldsymbol{\Omega}$. (D) "Single tree": a single tree by using all 9 features. (A) involved the construction of all decision trees and ample computational resources were considered available. (B) involved the construction of a limited number of decision trees using MAABO-MT, an efficient search algorithm. (C) involved the construction of a limited number of decision trees using randomly selected feature subsets. (D) involved the construction of a single decision tree using all prepared features. Usually, (D) is used for the the academic discussion based on the rules obtained by a single decision tree. In this study, (D) was prepared because a single decision tree was assumed to be insufficient for extracting a sufficient number of rules.

The number of decision trees constructed using (B) MAABO-MT, as indicated in Algorithm 1 was $N_I + N_B$. We set $N_I = 10$ as the initial solution size and $N_B \in \{0, 10, \ldots, 70\}$ as the iterations of Bayesian optimization. To investigate the relationship between the number of decision trees and performance in MAABO-MT, we set various values for N_B. For comparison under equal conditions, the number of decision trees constructed using (C) was the same as in (B). For other parameters of MAABO-MT, we adopted split coefficient $\alpha = 0.25$, maximum tree depth $p_{\max} = 5$, distribution degree of mismatches $h = 0.5$, damping coefficient $b = 0.5$, extracting a single feature size $N_U = D$, and sampling size $N_E \leftarrow \infty$.

All decision trees constructed in this study used the CART algorithm [48]. To avoid overfitting, we adopted maximum depth yielding maximized the F-score macro-average of the validation dataset. Data from 1309 samples were randomly split in a 7:3 ratio and assigned as training and validation data. To eliminate the effects of randomness, data were split and each method was performed using 50 random seeds. GS-MRM was used for rules extraction of the decision trees obtained using (A)–(D). To detect reliable rules, $(\beta, \gamma, \delta) = (50, 0.3, 0.7)$ was adopted.

TABLE 2. Leaf Nodes and Extracted Rules Size (100 Runs Ave. ± Std., and the Bold Numbers Represent Highest Values.)

Dataset	Method	Leaf nodes size $\|L\|$	Extracted rules size $\|L'\|$
Titanic	MAABO-MT	435.20 ± 30.48	$\mathbf{9.50 \pm 0.75}$
	Single tree	16.00 ± 0.00	2.00 ± 0.00
	RF	340.68 ± 3.87	8.90 ± 1.30
	RF-FS	654.66 ± 60.19	6.96 ± 0.96
	XGB-FS	323.62 ± 52.93	6.58 ± 1.25
	LGBM-FS	$\mathbf{840.38 \pm 52.40}$	4.56 ± 1.33
Boston	MAABO-MT	$\mathbf{531.18 \pm 7.31}$	$\mathbf{16.78 \pm 1.14}$
	Single tree	4.00 ± 0.00	2.00 ± 0.00
	RF	373.90 ± 6.02	8.28 ± 2.20
	RF-FS	461.14 ± 19.38	13.90 ± 1.75
	XGB-FS	345.52 ± 18.53	9.92 ± 1.73
	LGBM-FS	462.74 ± 23.95	14.82 ± 1.58
Diabetes	MAABO-MT	$\mathbf{266.84 \pm 11.67}$	$\mathbf{6.92 \pm 0.66}$
	Single tree	8.00 ± 0.00	1.00 ± 0.00
	RF	172.64 ± 3.75	2.62 ± 1.00
	RF-FS	258.52 ± 23.80	5.72 ± 0.92
	XGB-FS	253.46 ± 19.71	4.86 ± 1.17
	LGBM-FS	236.34 ± 22.62	5.16 ± 1.46

B. RESULT AND DISCUSSION

The obtained results are presented in Fig. 4. Approximately six rules were extracted in the case of (A), while approximately one rule was extracted in the case of (D). Thus, we confirmed that the traditional approach using all features to construct a single decision tree could only extract a fraction of the trusted rules that were latent in the data. Contrarily, the results obtained on using (A) succeeded in extracting more rules, but ample computing resources were consumed. This study was aimed at discovering appropriate rules without constructing all trees, as was done in (A). Therefore, the algorithm that satisfied our requirements was (B) MAABO-MT.

In the case of (B) MAABO-MT, almost all rules were found by constructing only half of all the decision trees. Since this result was superior to that of (C) randomized trees, MAABO-MT could extract numerous reliable rules in a shorter computation time. The details of the extracted rules are presented in Section VI and Table 3. Additional information are presented in Appendix 3 (Supplemental text, available online).

V. EXPERIMENT 2: ROBUSTNESS AGAINST NOISE FEATURES

A. OBJECTIVE AND OUTLINE

When analyzing real-world data, the dataset often contains noisy features (meaningless numerical information) without the knowledge of an analyst. Therefore, evaluation of robustness against meaningless noise features is important. Approaches that used all features to construct a single tree tended to adopt noisy features (Appendix 1 in Supplemental text, available online). Contrarily, MAABO-MT was expected to avoid the adoption of noisy features because it constructed trees by exploring feature subsets. This section describes the results of the analysis.

Noise features generated by uniform random numbers in the Titanic dataset were added. After $f_{10}, \cdots$ the features listed in Table 1 are the noise features, of which the number of features adopted for the experiment included $N_{\text{noise}} \in \{1, \ldots, 20\}$. Adding noise features to the nine proper features $f_1, \ldots, f_9$ included in the Titanic dataset, the total number of features D ranged from 10 to 29. The size of the FCS $|\boldsymbol{\Omega}|$ obtained by extracting the three features ($D' = 3$) ranged between 120 and 3,654, from (3).

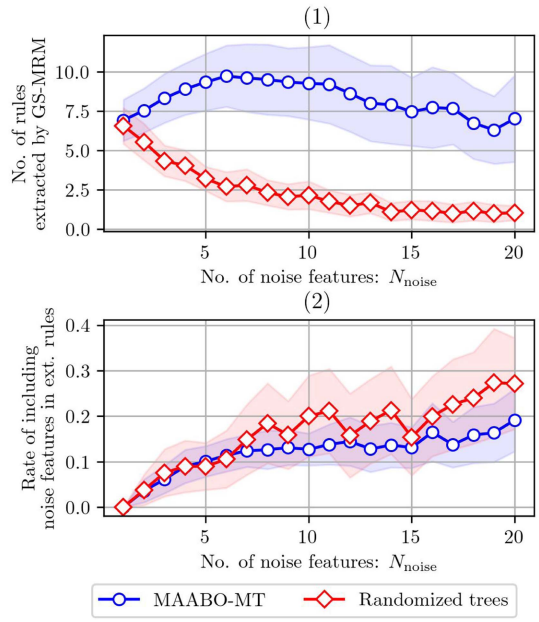

FIGURE 5. Effect on MAABO-MT search performance of noise features. Each value represents the average, and error bar represents 0.5 std. calculated from the results of 50 random seeds.

As the setting parameters of MAABO-MT, the initial solution size N_{I} was set to 10 and the number of iterations of Bayesian optimization was $N_{\text{B}} = 100$. Thus, 110 decision trees were constructed. The other parameters were the same as those used in Experiment 1. The randomized tree approach described in Experiment 1 was adopted for comparison. The number of decision trees constructed using the randomized trees was also 110. GS-MRM was used for rule extraction, and the adoption parameters were the same as those used in Experiment 1. Additionally, an analysis was conducted using 50 random seeds to eliminate the effects of randomness.

B. RESULT AND DISCUSSION

The results obtained from the aforementioned procedure are shown in Fig. 5 (1), which presents the number of rules extracted by the GS-MRM. In the case of randomized tree approach, we confirmed that the larger the number of noise features, the smaller the number of extracted reliable leaf nodes. However, MAABO-MT confirmed that a certain number of reliable leaf nodes could be extracted even when the noise features increased. Nonetheless, cases in which noisy features were included in the extracted rules could exist. Therefore, we calculated the noise content rate of the extracted rules, and the results are presented in Fig. 5 (2). Thus, the noise content in the extracted rules was confirmed to

TABLE 3. Extracted Rules by MAABO-MT and Single Tree (Titanic, Boston Housing, and Diabetes Datasets)

Dataset	Method	Rule ID	Class	Sample size	Gini	Extracted rule
Titanic	MAABO-MT	P1	Survival	59	0.00	Sex/female, Age/high, Fare/large
		P2	Survival	110	0.02	Sex/female, Fare/large, Pclass/good
		P3	Survival	72	0.04	Sex/female, ParCh/small, Pclass/good
		P4	Death	103	0.11	Sex/male, Age/high, Pclass/bad
		P5	Death	79	0.14	Sex/male, Fare/small
				### and other 5 rules ###		
	Single tree	S1	Survival	108	0.01	Sex/female, Age/high, Fare/large, Pclass/good
		S2	Death	103	0.11	Sex/male, Age/high, pclass/bad
Boston	MAABO-MT	P1	High price	100	0.03	LSTAT/small, RM/large
		P2	Low price	56	0.05	AGE/large, NOX/large
		P3	High price	75	0.04	NOX/small, RM/large
		P4	Low price	139	0.11	PTRATIO/large, LSTAT/large
		P5	High price	101	0.11	CRIM/small, RM/large
				### and other 11 rules ###		
	Single tree	S1	Low price	139	0.11	PTRATIO/large, LSTAT/large
		S2	High price	131	0.12	RM/large, LSTAT/small
Diabetes	MAABO-MT	P1	Low progression	81	0.03	HDL/large, BMI/small, LTG/small
		P2	Low progression	89	0.06	AGE/small, BMI/small, LTG/small
		P3	Low progression	58	0.08	TC/small, HDL/large, BMI/small
		P4	High progression	54	0.14	GLU/large, BMI/large, LTG/large
		P5	Low progression	81	0.14	AGE/small, HDL/large, BMI/small
				### and other 2 rules ###		
	Single tree	S1	Low progression	89	0.06	AGE/small, BMI/small, LTG/small

increase with an increase in noise features. Herein, MAABO-MT tended to have a slightly lower noise content than the randomized tree. Although the difference was slight, the randomized tree could only extract a small number of rules when the noise features were large (Fig. 5(1)). When there were more than 15 noise features, only one rule was extracted from the randomized trees.

Therefore, MAABO-MT could extract a certain number of reliable rules even when the number of noise features increased. The approach based on randomized trees was unable to extract rules when the number of noise features increased.

VI. EXPERIMENT 3: DETAILS OF RULES EXTRACTED BY MAABO-MT AND GS-MRM ALGORITHM

A. OBJECTIVE AND OUTLINE

This section presents the specific rules detected using MAABO-MT and GS-MRM. The datasets used included Titanic, Boston housing [49], [50], and diabetes [51]. Boston Housing is a dataset that is used to estimate home prices based on environmental factors and other attributes. Diabetes is a dataset used to estimate disease progression based on personality and blood components. Both datasets have been used to evaluate the performance of machine-learning algorithms [52], [53], [54], [55], [56]. The features and class labels included in each dataset are described in Appendix 4. The conditions for running MAABO-MT included the number of initial solutions $N_{\mathrm{I}} = 10$ and the number of trees to be constructed is $N_{\mathrm{I}} + N_{\mathrm{B}} = |\Omega|/2$, it means half of $|\Omega|$. The number of Bayesian optimization iteration is $N_{\mathrm{B}} = |\Omega|/2 - N_{\mathrm{I}}$. Other parameters were the same as those used in Experiment 1.

B. EXISTING METHODS

To verify the relative effectiveness of MAABO-MT, tree structure rule mining was also conducted using other methods. First, we adopted a single decision tree via CART, i.e., a traditional analysis method ("Single tree"). The second method is the random forest ("RF"), which constructs multiple decision trees via bootstrap sampling.

Note that the proposed MAABO-MT constructs multiple trees based on feature selection. Other existing methods of constructing feature selection-based decision trees were also explored. The first existing method based on feature selection is RF-based features selection ("RF-FS"). RF can measure feature importance via the impurities of the many branches in multiple trees [57]. This information is used to construct feature subsets and multiple decision trees. Specifically, first, the importance of the feature f_i is p_i. When this sum is normalized to $\sum p_i = 1$, a multinomial distribution emerges. According to this distribution, a specified number of features are sampled to construct multiple feature subsets (the same features are not adopted). Multiple decision trees are then constructed using the selected feature subsets. XG-boost and LightGBM, proposed after RF, can also measure feature importance [37], [38]. For this reason, we also adopted the methods of constructing multiple decision trees using feature importance by XG-boost and LightGBM ("XGB-FS" and "LGBM-FS"). Evaluating feature importance via RF, XG-boost, and LightGBM is popular, and these methods have been used in various studies [39], [40], [41].

In order to make comparisons under equal conditions, the numbers and maximum depths of the decision trees constructed via RF, RF-FS, XGB-FS, and LGBM-FS were identical to those constructed in MAABO-MT. The other hyperparameters were set to their default values [37], [38], [57]. Note that scikit-learn (ver. 1.4.1) [57] was used for RF and RF-FS, xgboost (ver. 2.0.3) [37] was used for XGB-FS, and LightGBM (ver. 4.3.0) [38] was used for LGBM-FS. Rules were extracted by applying GS-MRM to the multiple decision trees constructed using each method. To remove the effect of randomness, all methods were run 100 times with different random seeds.

C. RESULTS AND DISCUSSION

The total number of leaf nodes in the constructed decision trees is represented by $|\boldsymbol{L}|$, and $|\boldsymbol{L}'|$ denotes the total number of rules extracted by GS-MRM, as listed in Table 2. We can confirm that MAABO-MT succeeded in extracting the most rules. Note that although the LGBM-FS leaf node size $|\boldsymbol{L}|$ in the Titanic dataset is large, the extracted rule size $|\boldsymbol{L}'|$ is small. This means that many of the rules were similar. In the single tree case, which is the most classical method, the extracted rule size $|\boldsymbol{L}'|$ was small and insufficient for rule mining. In summary, MAABO-MT was the best approach.

Table 3 shows the rules obtained from the results of one seed, focusing on MAABO-MT, which succeeded in extracting the most rules, and single tree, which extracted only a few rules.

For the Titanic dataset and single tree, the rules for survival and death were limited. For example, S1 and S2 showed that elderly males and females were more likely to die and survive, respectively. However, the rules extracted by MAABO-MT confirmed that females were more likely to survive (P2 and P3) regardless of age. Thus, males were more likely to die regardless of age (P5). If the results of only a single decision tree were considered for discussion, misunderstandings such as the age being older could arise. Such issues were eliminated by using MAABO-MT. Additionally, MAABO-MT identified "low ParCh (P3)" as a survival condition, which was not identified in a single tree. As males were more likely to die, the entire family was not in a position to escape. Therefore, single individuals were considered more likely to escape and survive.

In the case of Boston, a single tree extracted the rule (S1) "lower housing prices when there are more students per teacher and more low-income families." Additionally, in the case of MAABO-MT, a different rule (P1) was extracted along with S1, such that "If the house is old and the air is dirty, the house price is low." In the case of a single tree, the rule (S2) that stated "areas with fewer low-income residents and more rooms have higher housing prices" was extracted. Conversely, in the case of MAABO-MT, rules such as "areas with low crime rates have higher housing prices" and "areas with clean air have higher housing prices" (P3 and P5) were extracted in addition to S2.

In Diabetes and single tree case, only the rule "low age, low BMI and low LTG, then, low progression of diabetes" was extracted (S1). Conversely, in the case of MAABO-MT, rules that could not be found in the single tree case, such as "if HDL is high, diabetes is low progression" (P1, P3, and P5) were extracted. The results were valid because patients with diabetes had low HDL [58]. Furthermore, MAABO-MT also extracted rules with high progression of diabetes that were not extracted in a single tree case (P4). Therefore, MAABO-MT provided deeper insights than a single tree for all the datasets tested.

VII. CONCLUSION

In this study, we highlight the disadvantages of rule extraction using a single decision tree, which is a traditional approach that can only discover a small fraction of the multiple rules latent in data. Therefore, we propose multi-rule mining algorithms MAABO-MT and GS-MRM to solve existing problems. We propose an MAA function that is required to solve the feature subset search problem with Bayesian optimization.

Several experimental results show that the proposed method has the following effects. Experiment 1: Of all the decision trees, MAABO-MT succeeded in discovering all the rules by constructing approximately half of them. Experiment 2: Robustness to noise features was observed. Experiment 3: MAABO-MT was able to extract a larger number of rules than were previously developed methods.

The proposed method includes several hyperparameters. Sensitivity analysis is performed on some of the parameters, while others are not fully analyzed. In the future, we plan to analyze all parameters to determine the recommended values for each hyperparameter.

REFERENCES

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

[2] S. S. Sundhari, "A knowledge discovery using decision tree by gini coefficient," in *Proc. Int. Conf. Bus., Eng. Ind. Appl.*, 2011, pp. 232–235.

[3] D. Y. Yeh, C. H. Cheng, and S. C. Hsiao, "Classification knowledge discovery in mold tooling test using decision tree algorithm," *J. Intell. Manuf.*, vol. 22, pp. 585–595, 2011.

[4] Z. Wen and Y. Tao, "Building a rule-based machine-vision system for defect inspection on apple sorting and packing lines," *Expert Syst. Appl.*, vol. 16, pp. 307–313, 1999.

[5] H. Hamsa, S. Indiradevi, and J. J. Kizhakkethottam, "Student academic performance prediction model using decision tree and fuzzy genetic algorithm," *Procedia Technol.*, vol. 25, pp. 326–332, 2016.

[6] S. Tangirala, "Evaluating the impact of Gini index and information gain on classification using decision tree classifier algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, pp. 612–619, 2020.

[7] C. Bessiere, E. Hebrard, and B. O'Sullivan, "Minimising decision tree size as combinatorial optimisation," *Lecture Notes Comput. Sci.*, vol. 5732 LNCS, pp. 173–187, 2009.

[8] N. Kokash and L. Makhnist, "Using decision trees for interpretable supervised clustering," *SN Comput. Sci.*, vol. 5, pp. 1–11, 2024.

[9] A. Singh, S. Saraswat, and N. Faujdar, "Analyzing Titanic disaster using machine learning algorithms," in *Proc. IEEE Int. Conf. Comput., Commun. Automat.*, 2017, pp. 406–411.

[10] J. Sherlock, M. Muniswamaiah, L. Clarke, and S. Cicoria, "Classification of Titanic passenger data and chances of surviving the disaster," 2018, *arXiv:1810.09851v1*.

[11] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[12] D. Petkovic, R. Altman, M. Wong, and A. Vigil, "Improving the explainability of random forest classifier – user centered approach," in *Proc. Pacific Symp. Biocomputing,* 2018, vol. 23, pp. 204–215.

[13] M. P. Neto and F. V. Paulovich, "Explainable matrix - visualization for global and local interpretability of random forest classification ensembles," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1427–1437, Feb. 2021.

[14] F. Gossen and B. A. Steffen, "Algebraic aggregation of random forests: Towards explainability and rapid evaluation," *Int. J. Softw. Tools Technol. Transfer*, vol. 25, pp. 267–285, 2023.

[15] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 24, pp. 2546–2554.

[16] S. Watanabe, "Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance," 2023, *arXiv:2304.11127v3*.

[17] A. Agnesina et al., "AutoDMP: Automated dreamplace-based macro placement," in *Proc. Int. Symp. Phys. Des.*, 2023, pp. 149–157.

[18] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the Gini index and information gain criteria," *Ann. Math. Artif. Intell.*, vol. 41, pp. 77–93, 2004.

[19] L. Antonioli et al., "Convolutional neural networks cascade for automatic pupil and iris detection in ocular proton therapy," *Sensors*, vol. 21, no. 13, 2021, Art. no. 4400.
[20] X. Wu et al., "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, pp. 1–37, 2008.
[21] A. Cherfi, K. Nouira, and A. Ferchichi, "Very fast c4.5 decision tree algorithm," *Appl. Artif. Intell.*, vol. 32, pp. 119–137, 2018.
[22] M. Milanovic and M. Stamenkovic, "Chaid decision tree: Methodological frame and application," *Econ. Themes*, vol. 54, pp. 563–586, 2016.
[23] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, "The cart decision tree for mining data streams," *Inf. Sci.*, vol. 266, pp. 1–15, 2014.
[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC, 2017.
[25] A. L. Boulesteix, S. Janitza, J. Kruppa, and I. R. Konig, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *Wiley Interdiscipl. Reviews: Data Mining Knowl. Discov.*, vol. 2, pp. 493–507, 2012.
[26] Y. Qi, "Random forest for bioinformatics," in *Ensemble Machine Learning*, C. Zhang, and Y.Q. Ma, Eds., USA: Springer, 2012, pp. 307–323, doi: 10.1007/978-1-4419-9326-7_11.
[27] M. Belgiu and L. Dragu, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogrammetry Remote Sens.*, vol. 114, pp. 24–31, 2016.
[28] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, pp. 1189–1232, 2001.
[29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[30] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 3149–3157.
[31] L. Huang et al., "Combining random forest and XGBoost methods in detecting early and mid-term winter wheat stripe rust using canopy level hyperspectral measurements," *Agriculture*, vol. 12, pp. 74, 2022.
[32] M. Z. Joharestani, C. Cao, X. Ni, B. Bashir, and S. Talebiesfandarani, "$Pm_{2.5}$ prediction based on random forest, XGboost, and deep learning using multisource remote sensing data," *Atmosphere*, vol. 10, 2019, Art. no. 373.
[33] P. Tao, H. Shen, Y. Zhang, P. Ren, J. Zhao, and Y. Jia, "Status forecast and fault classification of smart meters using LightGBM algorithm improved by random forest," *Wireless Commun. Mobile Comput.*, vol. 2022, 2022, Art. no. 3846637.
[34] D. N. Wang, L. Li, and D. Zhao, "Corporate finance risk prediction based on LightGBM," *Inf. Sci.*, vol. 602, pp. 259–268, 2022.
[35] M. Loecher, "Unbiased variable importance for random forests," *Commun. Statist. - Theory Methods*, vol. 51, pp. 1413–1425, 2020.
[36] R. Yao, J. Li, M. Hui, L. Bai, and Q. Wu, "Feature selection based on random forest for partial discharges characteristic set," *IEEE Access*, vol. 8, pp. 159151–159161, 2020.
[37] "Python API reference, XGBoost (ver. 2.0.3)." Accessed: Mar. 6, 2024. [Online]. Available: https://xgboost.readthedocs.io/en/stable/python/python_api.html
[38] "Parameters, LightGBM (ver. 4.3.0)." Accessed: Mar. 6, 2024. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/Parameters.html
[39] T. Venkateswarlu and J. Anmala, "Importance of land use factors in the prediction of water quality of the Upper Green River watershed, Kentucky, USA, using random forest," *Environ., Develop. Sustain.*, pp. 1–24, 2023.
[40] S. B. Jabeur, N. Stef, and P. Carmona, "Bankruptcy prediction using the XGBoost algorithm and variable importance feature engineering," *Comput. Econ.*, vol. 61, pp. 715–741, 2023.
[41] L. Li, X. Cui, J. Yang, X. Wu, and G. Zhao, "Using feature optimization and LightGBM algorithm to predict the clinical pregnancy outcomes after in vitro fertilization," *Front. Endocrinol.*, vol. 14, 2023, Art. no. 1305473.
[42] J. Aitchison and C. G. Aitken, "Multivariate binary discrimination by the kernel method," *Biometrika*, vol. 63, pp. 413–420, 1976.
[43] "Sklearn.tree.decisiontreeclassifier–scikit-learn 1.3.0 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/classes.html
[44] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: Overview study and experimental results," in *Proc. IEEE 11th Int. Conf. Inf. Commun. Syst.*, 2020, pp. 243–248.
[45] K. Singh, R. Nagpal, and R. Sehgal, "Exploratory data analysis and machine learning on Titanic disaster dataset," in *Proc. Confluence - 10th Int. Conf. Cloud Comput., Data Sci. Eng.*, 2020, pp. 320–326.
[46] A. Singh, S. Saraswat, and N. Faujdar, "Analyzing Titanic disaster using machine learning algorithms," in *Proc. IEEE Int. Conf. Comput., Commun. Automat.*, 2017, pp. 406–411.
[47] N. Farag and G. Hassan, "Predicting the survivors of the Titanic - Kaggle, machine learning from disaster," in *Proc. 7th Int. Conf. Softw. Inf. Eng.*, 2018, pp. 32–37.
[48] S. Singh and M. Giri, "Comparative study ID3, cart and C4.5 decision tree algorithm: A survey," *Int. J. Adv. Inf. Sci. Technol.*, vol. 3, pp. 47–52, 2014.
[49] "The boston house-price data." [Online]. Available: http://lib.stat.cmu.edu/datasets/boston
[50] D. Harrison Jr and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *J. Environ. Econ. Manage.*, vol. 5, no. 1, pp. 81–102, 1978.
[51] "Toy datasets (7.1.2. diabetes dataset) - scikit-learn 1.4.dev0 documentation," [Online]. Available: https://scikit-learn.org/dev/datasets/toy_dataset.html#diabetes-dataset
[52] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, pp. 407–499, 2004.
[53] A. Honda, M. Itabashi, and S. James, "A neural network based on the inclusion-exclusion integral and its application to data analysis," *Inf. Sci.*, vol. 648, 2023, Art. no. 119549.
[54] S. Oh, "Feature interaction in terms of prediction performance," *Appl. Sci.*, vol. 9, 2019, Art. no. 5191.
[55] Y. Chen and Y. Yang, "The one standard error rule for model selection: Does it work?," *Stats*, vol. 4, pp. 868–892, 2021.
[56] X. Peng, "TSVR: An efficient twin support vector machine for regression," *Neural Netw.*, vol. 23, pp. 365–372, 2010.
[57] "RandomForestClassifier, scikit-learn (ver. 1.4.1)." Accessed: Mar. 6, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[58] M. F. Lopes-Virella, P. G. Stone, and J. A. Colwell, "Serum high density lipoprotein in diabetic patients," *Diabetologia*, vol. 13, pp. 285–291, 1977.

YUTO OMAE received the Ph.D. degree in engineering from the Nagaoka University of Technology, Niigata, Japan, in 2016. He is currently a Lecturer with the Department of Industrial Engineering and Management and the vice Director with the Artificial Intelligence Research Center, College of Industrial Technology, Nihon University, Tokyo, Japan. His research interests include theories of machine learning and Bayesian optimization.

MASAYA MORI received the master's degree in engineering from the Nagaoka University of Technology, Niigata, Japan, in 2022. He is currently a Researcher with College of Industrial Technology, Nihon University. His research interests include intelligent informatics and machine learning.

YOHEI KAKIMOTO received the Ph.D. degree in engineering from Nihon University, Tokyo, Japan in 2023. He is currently an Assistant Professor with the College of Industrial Technology, Nihon University. His research interests include operations research and social simulation.