Improving Generalization Performance of Multi-objective Fuzzy Genetics-Based Machine Learning using Synthetic Training Data Generated from High-Performance Black-Box Models

Yuji Shuto
Undergraduate School of Informatics
Osaka Metropolitan University
Sakai, Japan
sm22341z@st.omu.ac.jp

Naoki Masuyama
Graduate School of Informatics
Osaka Metropolitan University
Sakai, Japan
masuyama@omu.ac.jp

Yusuke Nojima
Graduate School of Informatics
Osaka Metropolitan University
Sakai, Japan
nojima@omu.ac.jp

***Abstract*—Recently, the interpretability of classifiers has become increasingly important, and fuzzy classifiers based on linguistically interpretable fuzzy IF-THEN rules have attracted attention. Multi-objective Fuzzy Genetics-Based Machine Learning (MoFGBML) is a method capable of efficiently generating a set of fuzzy classifiers considering both classification performance and interpretability, using an evolutionary multi-objective optimization algorithm. However, MoFGBML generally exhibits inferior classification performance compared to black-box models. To enhance the classification performance of MoFGBML, in this study, we propose transferring the classification knowledge of LightGBM, which possesses strong generalization ability. Specifically, we augment the training data by adding new patterns with the class label predicted by LightGBM. Computational experiments on real-world datasets demonstrate that MoFGBML trained with the proposed method can achieve higher classification accuracy while maintaining interpretability, compared to the conventional one trained solely on the original dataset.**

***Index Terms*—Fuzzy Classifier, Multi-objective Fuzzy Genetics-Based Machine Learning, LightGBM**

I. Introduction

In many applications (e.g., finance and medical diagnosis), stakeholders require explanations of the basis and internal process of Artificial Intelligence (AI) decision-making. Therefore, the importance of highly transparent AI is increasing [1], [2]. A fuzzy classifier is a highly transparent AI that performs a decision process (i.e., fuzzy inference) based on fuzzy rules whose antecedent parts are fuzzy sets associated with linguistic labels. A fuzzy classifier can make decisions considering real-world uncertainties [3]–[5].

Evolutionary computation has been actively used in fuzzy classifier design under the name of Evolutionary Fuzzy Systems [6]. In particular, Evolutionary Multi-objective Optimization Algorithm (EMOA) can offer diverse solutions (i.e., fuzzy systems) with different trade-offs among objectives to users, since they can handle multiple objective functions simultaneously in the design process [7]. While fuzzy classifiers have high interpretability, they generally exhibit inferior classification performance compared to black-box models in which their decision-making basis and internal processes are unclear. Black-box models such as LightGBM, exhibit extremely high classification and generalization performance in complex feature spaces, although their interpretability is low [8]. The main reasons why fuzzy classifiers are inferior to black-box models are that they cannot sufficiently capture complex decision boundaries using only training data, or there is insufficient learning in regions with few data patterns. If fuzzy classifiers can acquire the classification knowledge of black-box models, there is a possibility to significantly improve their classification performance while maintaining high interpretability.

To improve classification performance while maintaining high interpretability, in this study, we propose a framework to transfer the classification knowledge of LightGBM to MoFGBML. Specifically, we augment the training data by adding new patterns with the class label predicted by LightGBM. By training on such augmented data, MoFGBML is expected to appropriately approximate the complex decision boundaries of LightGBM. That is, the proposed method can be regarded as a data-level knowledge distillation framework, where a high-performance black-box model transfers its decision boundary to an interpretable fuzzy classifier.

In this paper, we introduce four data generation strategies; two perturbation-based pattern generation strategies and two Synthetic Minority Over-sampling Technique (SMOTE)-based strategies [9]. SMOTE is an over-sampling method that performs linear interpolation between samples. The four strategies aim to generate new patterns in different target regions respectively. We examine the effect of these strategies on the classification accuracy and complexity of MoFGBML, and evaluate the improvement in generalization performance of the fuzzy classifier through knowledge transfer from the

AI language models (Google Gemini, 3 pro and OpenAI ChatGPT, GPT-5.2) were used to support improvements in English expression and to clarify selected explanations in this manuscript. The technical content, experimental design, and conclusions were conceived, produced, and verified entirely by the authors.

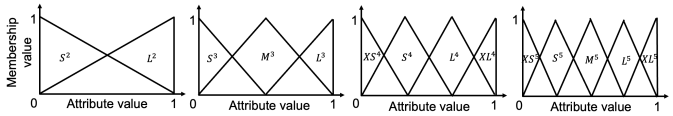

Fig. 1. Triangular membership functions.

black-box model.

The rest of this paper is organized as follows. First, Section II briefly describes fuzzy classifiers, MoFGBML, and LightGBM. Section III describes the four data augmentation strategies. Section IV presents experimental results using real-world datasets. Finally, Section V concludes this paper.

II. Preliminaries

A. Fuzzy Classifier

Consider a pattern classification problem given M-class n-dimensional data with m patterns $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$ where x_{pi} denotes the i-th attribute value $(i = 1, 2, \ldots, n)$ in the p-th pattern $(p = 1, 2, \ldots, m)$. A fuzzy classifier S consists of a set of fuzzy if-then rules of the following form as in [3].

$$\text{Rule } R_q : \text{If } x_{p1} \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_{pn} \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q,$$

where R_q is the label for the q-th fuzzy rule and $\mathbf{A}_q = (A_{q1}, \ldots, A_{qn})$ is the antecedent fuzzy set. We use 14 triangular-type membership functions from 2 to 5 partitions shown in Fig. 1 and an additional "*don't care*" condition which sets the membership value to 1 regardless of the attribute value. C_q is the consequent class and CF_q is the rule weight. These are uniquely determined from the antecedent part of the fuzzy rule and the training data. For more details, see [3] and [10].

For classification of unknown patterns, we use a single-winner rule strategy [3]. In this strategy, first, the product of the compatibility grade of each rule to the unknown pattern and its rule weight is calculated. Then, the rule with the largest product value is chosen as a single-winner rule. The consequent class of the winner rule is used as the classification result of the unknown pattern.

B. MoFGBML

Multi-objective Fuzzy Genetics-Based Machine Learning (MoFGBML) is an evolutionary machine learning technique that simultaneously optimizes the accuracy and interpretability of fuzzy classifiers [10]. MoFGBML adopts a hybrid framework combining the Pittsburgh-style and Michigan-style approaches. The optimization is primarily based on the Pittsburgh-style approach, where each individual in the population represents a complete set of fuzzy rules. In the Michigan-style approach, where each individual corresponds to a single rule, it is employed as a heuristic local search mechanism to refine specific rules.

The design of fuzzy classifiers is formulated as a multi-objective optimization problem with the following two objectives to be minimized:

1) Error rate ($f_1(S)$): The percentage of misclassified patterns on the training data. This objective evaluates the classification performance of the fuzzy classifier S.
2) Number of rules ($f_2(S)$): The total number of fuzzy rules $|S|$ in the classifier. Minimizing this objective improves the interpretability.

To efficiently search for a set of non-dominated solutions that trade off these objectives, MoFGBML utilizes Non-dominated Sorting Genetic Algorithm II (NSGA-II) [11]. We use the open source codes of MoFGBML[1].

C. LightGBM

Light Gradient Boosting Machine (LightGBM) is a highly efficient and scalable implementation of Gradient Boosting Decision Trees (GBDT) [8]. We use LightGBM thanks to its strong generalization and ability to model complex nonlinear boundaries.

III. Proposed Method

In this section, we describe the data augmentation framework for transferring classification knowledge of LightGBM to MoFGBML. The proposed method consists of three main stages: (1) learning of LightGBM as a teacher model, (2) expansion of the training data to construct the augmented data, and (3) training of MoFGBML using the augmented data.

A. Overall Framework

The purpose of this study is to enable MoFGBML to learn the complex decision boundaries, which are difficult to represent with the original training data alone, through generated patterns with the labels predicted by LightGBM.

First, LightGBM is trained using the original training dataset D_{train} to serve as a teacher model. The hyperparameters of LightGBM are optimized based on cross-validation using Optuna [12]. We define the trained LightGBM as $F_T(\cdot)$.

Next, new patterns $\mathbf{x}_{new}$ are generated in the feature space using the data augmentation strategies described in the next subsection. The class label for $\mathbf{x}_{new}$ is assigned with the predicted label by the trained LightGBM, i.e., $y_{new} = F_T(\mathbf{x}_{new})$. These generated patterns with predicted labels construct the dataset D_{gen}. Furthermore, to ensure consistent knowledge transfer of the teacher model, the class labels of the original training data D_{train} are also replaced with the predicted labels given by F_T.

Finally, the augmented data $D_{aug} = D_{train} \cup D_{gen}$, comprising the original data and the generated data, is used as the training data for MoFGBML. Consequently, the training data possesses the knowledge of the decision boundaries held by the teacher model, enabling MoFGBML to efficiently learn the classification knowledge of the teacher model.

B. Data Augmentation Strategies

To transfer classification knowledge of the teacher model, we introduce the following four data augmentation strategies.

[1]https://github.com/ci-labo-omu/MoFGBML_Basic

1) Strategy I: Perturbation on Misclassified Patterns: This strategy focuses on enhancing the learning in regions where the classification is difficult for MoFGBML.

1. Using the MoFGBML trained on the original training dataset D_{train}, classify D_{train} to identify the set of misclassified patterns D_{miss}:
$$D_{miss} = \{\mathbf{x} \in D_{train} \mid y_{classify} \neq y_{true}\} \quad (1)$$
2. For a pattern $\mathbf{x} \in D_{miss}$, add a noise vector $\boldsymbol{\epsilon}$ following a normal distribution with mean 0 and variance σ^2 to generate a new pattern $\mathbf{x}_{new}$:
$$\mathbf{x}_{new} = \mathbf{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2) \quad (2)$$
3. Assign the label predicted by the trained teacher model, $F_T(\mathbf{x}_{new})$, to the generated $\mathbf{x}_{new}$ and add it to D_{gen}.
4. Repeat steps 2-3 m times to increase the local data density.

We examine $\sigma \in \{0.02, 0.04, \ldots, 0.2\}$ and $m \in \{1, 2, 5\}$.

2) Strategy II: Perturbation on All Patterns: This strategy aims to faithfully approximate the entire decision surface of the teacher model by increasing the sampling density across the feature space.

1. Target all training patterns $\mathbf{x} \in D_{train}$.
2. Add perturbation $\boldsymbol{\epsilon}$ using Eq. (2) to generate $\mathbf{x}_{new}$.
3. Assign the label predicted by the trained teacher model, $F_T(\mathbf{x}_{new})$, to the generated $\mathbf{x}_{new}$ and add it to D_{gen}.
4. Repeat steps 2-3 m times to uniformly expand the entire dataset.

The settings for σ and m are the same as those of Strategy I.

3) Strategy III: Decision Boundary Interpolation with SMOTE: To precisely capture the decision boundaries of the teacher model, patterns are densely generated in regions where the class predicted by the teacher model differs from that of neighbor patterns.

1. Classify all training patterns $\mathbf{x} \in D_{train}$ using the trained teacher model F_T.
2. For each pattern $\mathbf{x}_i \in D_{train}$, identify a set of k nearest neighbors $N_k(\mathbf{x}_i)$ based on the Euclidean distance.
3. Identify the neighbor $\mathbf{x}_{neigh} \in N_k(\mathbf{x}_i)$ whose predicted label differs from that of $\mathbf{x}_i$ (i.e., $F_T(\mathbf{x}_i) \neq F_T(\mathbf{x}_{neigh})$).
4. Generate a new pattern $\mathbf{x}_{new}$ by applying SMOTE to linearly interpolate between these two samples and assign the predicted label $F_T(\mathbf{x}_{new})$ to the generated $\mathbf{x}_{new}$ and add it to D_{gen}.:
$$\mathbf{x}_{new} = \mathbf{x}_i + \delta \cdot (\mathbf{x}_{neigh} - \mathbf{x}_i), \quad \delta \in [0, 1] \quad (3)$$

We examine $k \in \{1, 2, 5, 10, 15, 20\}$.

4) Strategy IV: Decision Boundary Interpolation with SMOTE-NC: Real-world datasets often contain a mixture of numerical and categorical attributes. Applying simple Euclidean distance or linear interpolation can cause scale discrepancies between attributes or semantic contradictions. Therefore, we perform data augmentation applying SMOTE-NC for datasets containing categorical attributes.

1. Classify all training patterns $\mathbf{x} \in D_{train}$ using the trained teacher model F_T.
2. Calculate the distance by integrating the Euclidean distance for numerical attributes and the Hamming distance for categorical attributes to identify a set of k nearest neighbors $N_k(\mathbf{x}_i)$. The distance $d(\mathbf{x}_i, \mathbf{x}_j)$ between two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as follows:
$$d(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{n \in Num} (x_{i,n} - x_{j,n})^2 + V \cdot \sum_{c \in Cat} \Gamma(x_{i,c} \neq x_{j,c}) \right]^{1/2} \quad (4)$$
Num and Cat are the sets of numerical and categorical attributes, respectively. $\Gamma(\cdot)$ is the indicator function representing the Hamming distance (returns 1 if attributes differ, 0 otherwise). V is the penalty coefficient for the categorical distance. In this study, V is defined as the median of the variances of all numerical attributes.
3. Identify the neighbor $\mathbf{x}_{neigh} \in N_k(\mathbf{x}_i)$ whose predicted label differs from that of $\mathbf{x}_i$ and generate $\mathbf{x}_{new}$ as follows:
   - *Numerical Attributes:* Perform linear interpolation using Eq. (3).
   - *Categorical Attributes:* Randomly select and inherit the attribute value from either $\mathbf{x}_i$ or $\mathbf{x}_{neigh}$.
4. Assign the predicted label $F_T(\mathbf{x}_{new})$ to the generated $\mathbf{x}_{new}$ and add it to D_{gen}.

The settings for k are the same as those of Strategy III.

C. MoFGBML with Transferred Knowledge

MoFGBML is trained using the augmented data generated in Section III-B to design fuzzy classifiers. MoFGBML simultaneously minimizes two objective functions: the error rate $f_1(S)$ and the number of rules $f_2(S)$. The resulting non-dominated solution set provides classifiers that inherit the high classification performance of LightGBM while maintaining interpretability based on IF-THEN rules. In our experiments, we evaluate the generalization performance of the obtained fuzzy classifiers using test data that were not used in the training process of LightGBM and MoFGBML.

IV. Computational Experiments

A. Datasets and Experimental Settings

Table I shows the fourteen real-world datasets used in the computational experiments. These datasets are available in the KEEL dataset repository [13]. In the table, N and C denote the number of numerical and categorical attributes, respectively. In our experiments, all attributes, including categorical ones, are normalized to the interval $[0, 1]$.

In the computational experiments, we optimized the hyperparameters of LightGBM using Optuna. Table II shows the optimized hyperparameters and training settings. Next, we design fuzzy classifiers using MoFGBML trained on the augmented

TABLE I
REAL-WORLD DATASETS

Dataset	# Patterns	# Attributes (N/C)	# Classes
bal	625	4 (0/4)	3
bupa	345	6 (6/0)	2
contraceptive	1473	9 (2/7)	3
heart	270	13 (5/8)	2
iris	150	4 (4/0)	3
mammographic	830	5 (1/4)	2
newthyroid	215	5 (5/0)	3
pima	768	8 (8/0)	2
sonar	208	60 (60/0)	2
tae	151	5 (1/4)	3
vehicle	846	18 (18/0)	4
wine	178	13 (13/0)	3
wisconsin	699	9 (9/0)	2
yeast	1484	8 (8/0)	10

TABLE II
HYPERPARAMETERS AND PARAMETER SETTINGS OF LIGHTGBM

Settings / Parameters	Descriptions / values
Tuning Strategy	Stepwise Tuning
Objective	binary / multiclass
Metric	binary_logloss / multi_logloss
Boosting Type	gbdt
Max Boosting Rounds	2,000
Early Stopping Rounds	200
Automatically Tuned Hyperparameters	**Search range**
Feature fraction	$[0.4, 1.0]$
Number of leaves	$[2, 256]$
Bagging fraction	$[0.4, 1.0]$
Bagging frequency	$[1, 7]$
L1 regularization	$[10^{-8}, 10.0]$
L2 regularization	$[10^{-8}, 10.0]$
Min child samples	$[5, 100]$

data. Table III shows the parameters of MoFGBML. We performed each strategy for 30 runs (i.e., 10-fold cross-validation repeated three times). To ensure statistical reliability, average error rates were calculated only for rule counts that appeared in at least 16 out of 30 runs.

B. Experimental Results

First, we select the bupa dataset as a representative example to visualize the effects of different settings of m, σ, and k for perturbation-based sampling and SMOTE-based interpolation on the trade-off between the error rate and the number of rules. Figs. 2, 3, and 4 illustrate the test error rates for all settings of m, σ, and k in Strategies I, II, and III, respectively.

Focusing on the results obtained using only the original training data in the conventional approach (i.e., red line in the plot), we can observe that the error rate is approximately 5% worse than that of LightGBM. We can also observe that the error rate remains almost unchanged when the number of rules exceeds six. This indicates that increasing the complexity (i.e., the number of rules) does not necessarily lead to an improvement in generalization performance.

We compare the results obtained using the augmented training data with the original data in Figs. 2, 3, and 4.

TABLE III
PARAMETER SETTINGS OF MOFGBML

Parameters	Values
Initial population size	60
Offspring population size	60
Number of generations	5,000
Number of evaluated solutions	300,000
Probability of Michigan operation	0.5
Crossover probability in Pittsburgh operation	0.9
Mutation probability in Pittsburgh operation	$1/N(S)$
Crossover probability in Michigan operation	0.9
Mutation probability in Michigan operation	$1/n$
Rule change rate in Michigan operation	0.2
Max number of rules	60

$N(S)$: Number of rules in S, n: Number of attributes

While the results vary according to the parameters m, σ, and k, the proposed strategies outperform the conventional method (i.e., "Original" in the plots) in most settings. In some settings, generalization performance comparable to that of LightGBM is also achieved. This demonstrates the robustness of the proposed data augmentation strategies and indicates the successful generation of fuzzy classifiers with higher generalization performance.

It is worth noting that, in Strategies II and III as shown in Figs. 3 and 4, using the augmented data tends to generate fuzzy classifiers with a larger number of rules than those trained with only the original data. In Strategy III, increasing the value of k tends to generate new patterns across the entire pattern distribution, making it more closer to Strategy II. We need to further investigate how augmenting patterns over the entire pattern distribution contributes to obtaining of fuzzy classifiers with a larger number of rules. This is left as future work.

Next, we compare the generalization performance and the complexity across all datasets. We choose the fuzzy classifier with the best training error rate among the non-dominated classifiers obtained by each MoFGBML. Table IV shows the test error rates at the number of rules that minimized the training error rate obtained by MoFGBML and the average test error rates obtained by LightGBM. "Original*" indicates the test error rates of the MoFGBML trained on the original dataset without data augmentation, where all class labels are replaced by LightGBM's predicted labels. The numbers in parentheses represent the number of trees for LightGBM and the number of rules in the selected fuzzy classifier. It should be noted that we applied Strategy IV only to the datasets which include the categorical attributes. Table V shows three parameters (m, σ, k) used to obtain the results in Table IV. In Table IV, the best result among MoFGBML (i.e., Original, Strategies I-IV) for each dataset is highlighted by boldface. Better results than the original are also highlighted by blue shading.

From Table IV, we can observe the fuzzy classifiers obtained using the augmented data improved the generalization performance for most of the datasets. Strategies I and II seem to be more promising. The number of rules was not substantially larger than that obtained using the original data. We can also observe the error rates for some settings were

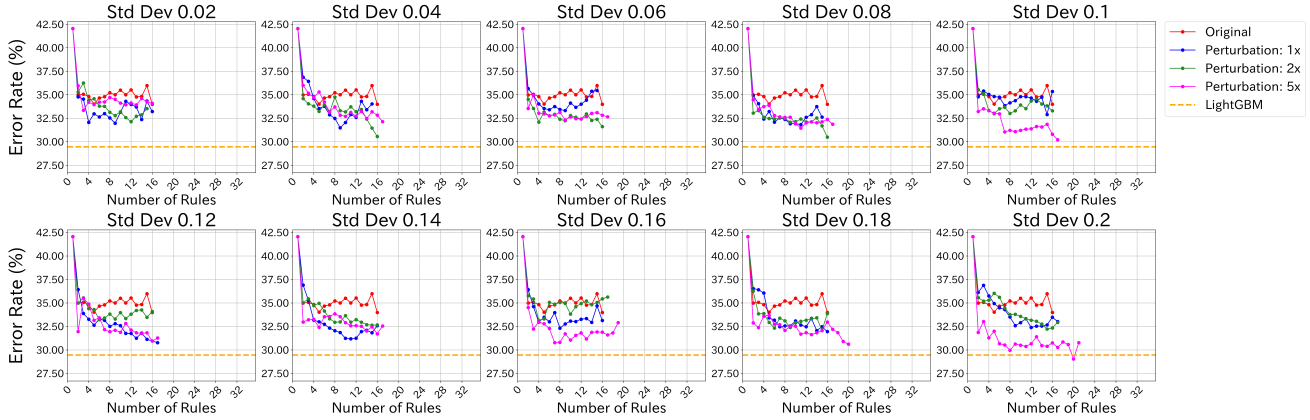

Fig. 2. Test error rates on Bupa dataset for Strategy I

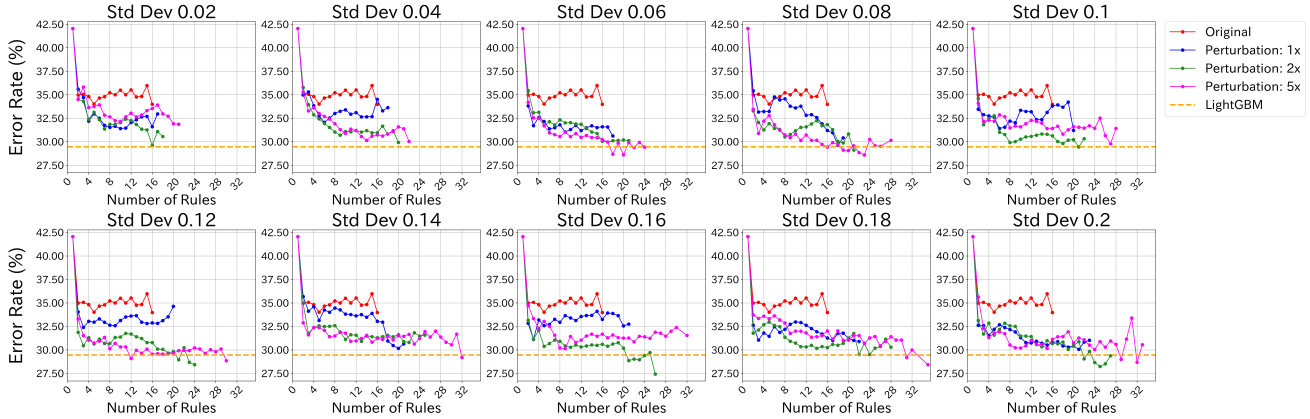

Fig. 3. Test error rates on Bupa dataset for Strategy II

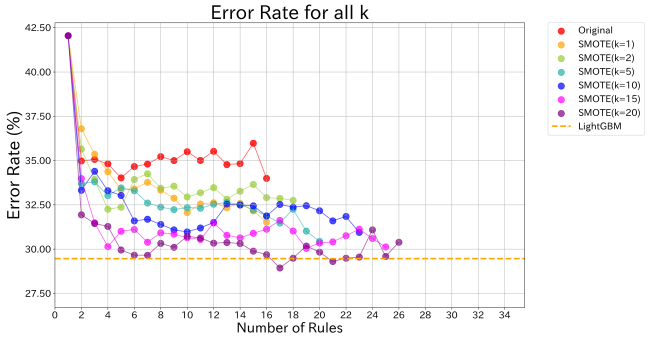

Fig. 4. Test error rates on Bupa dataset for Strategy III

better than those by LightGBM (e.g., Strategy I for iris and pima, Strategy II for newthyroid, Strategy IV for tae).

From Table V, we can observe the number of repetitions m was low in Strategies I and II. This indicates that a large number of new patterns are not always necessary. We can also observe that the number of neighbors k was small in Strategies III and IV for most of the datasets. When we use a small k for SMOTE, new patterns are generated only in the decision boundaries. The number of newly generated patterns would also be smaller than the case when we use a large k.

Notably, Strategy II appears to be the most promising in terms of generalization performance among the augmented data generation strategies, although it does not focus only on the decision boundaries. It is possible that patterns are also generated outside the original pattern distribution, and this may be reflected in the results. In particular, since this study uses triangular membership functions shown in Fig. 1, fuzzy classifiers obtained by MoFGBML do not necessarily cover the entire feature space. Further investigation is required to clarify this issue.

V. Conclusion

In this paper, we proposed four data augmentation strategies for transferring the classification knowledge of a black-box model to MoFGBML, with the aim of designing fuzzy classifiers that possess high generalization performance while maintaining interpretability. Experimental results showed that the proposed methods successfully learned the classification knowledge of the teacher model (i.e., LightGBM).

As future work, we further investigate the reason why classifiers with a large number of rules are obtained and the distribution of the newly generated patterns. Future work also

TABLE IV

TEST ERROR RATES AT THE NUMBER OF RULES THAT MINIMIZED THE TRAINING ERROR RATE OBTAINED BY MOFGBML AND AVERAGE TEST ERROR RATES OBTAINED BY LIGHTGBM. THE NUMBERS IN THE PARENTHESES REPRESENT THE MODEL COMPLEXITY.

Dataset	LightGBM	Original	Original*	Strategy I	Strategy II	Strategy III	Strategy IV
bal	2.19 (5909)	16.91 (34)	15.43 (34)	18.53 (42)	**15.13 (37)**	17.18 (37)	-
bupa	29.46 (265)	33.99 (16)	32.32 (15)	**30.61 (20)**	30.64 (21)	31.52 (16)	-
contraceptive	44.56 (219)	45.77 (38)	45.15 (32)	46.01 (42)	**44.74 (38)**	46.92 (38)	46.37 (35)
heart	15.93 (84)	17.34 (15)	18.48 (11)	16.88 (12)	16.90 (24)	16.94 (14)	**16.29 (12)**
iris	4.89 (344)	5.66 (4)	4.93 (3)	5.83 (4)	**4.88 (5)**	6.61 (4)	-
mammographic	15.73 (108)	20.47 (14)	17.49 (10)	**15.28 (8)**	16.14 (14)	17.42 (11)	17.98 (11)
newthyroid	4.00 (562)	6.48 (6)	6.31 (6)	5.93 (6)	**3.66 (8)**	5.20 (6)	-
pima	24.12 (76)	24.76 (20)	24.11 (15)	24.72 (16)	24.85 (23)	**23.34 (16)**	-
sonar	11.92 (108)	23.92 (12)	23.92 (12)	23.92 (12)	**22.55 (16)**	23.07 (14)	-
tae	47.44 (173)	49.99 (17)	47.41 (15)	47.45 (17)	52.06 (17)	47.52 (16)	**46.35 (15)**
vehicle	22.58 (1060)	30.32 (36)	29.14 (34)	**28.19 (38)**	29.88 (38)	30.53 (40)	-
wine	2.09 (425)	7.94 (4)	4.57 (8)	7.94 (4)	4.95 (5)	**3.37 (5)**	-
wisconsin	2.97 (63)	4.42 (9)	7.94 (4)	3.58 (8)	**3.50 (10)**	3.51 (8)	-
yeast	38.68 (667)	42.04 (28)	**40.70 (26)**	40.92 (33)	41.51 (39)	41.84 (27)	-

"-" indicates that the strategy is not applicable due to attribute types.

TABLE V

THE PARAMETERS (m, σ, k) USED TO OBTAIN THE RESULTS IN TABLE IV

Dataset	Strategy I (m, σ)	Strategy II (m, σ)	Strategy III (k)	Strategy IV (k)
bal	$m = 1, \sigma = 0.02$	$m = 1, \sigma = 0.02$	$k = 1$	-
bupa	$m = 5, \sigma = 0.18$	$m = 1, \sigma = 0.14$	$k = 1$	-
contraceptive	$m = 1, \sigma = 0.16$	$m = 1, \sigma = 0.2$	$k = 1$	$k = 1$
heart	$m = 1, \sigma = 0.12$	$m = 1, \sigma = 0.2$	$k = 1$	$k = 1$
iris	$m = 5, \sigma = 0.16$	$m = 2, \sigma = 0.02$	$k = 2$	-
mammographic	$m = 2, \sigma = 0.16$	$m = 1, \sigma = 0.18$	$k = 1$	$k = 1$
newthyroid	$m = 1, \sigma = 0.02$	$m = 1, \sigma = 0.02$	$k = 1$	-
pima	$m = 1, \sigma = 0.14$	$m = 1, \sigma = 0.2$	$k = 1$	-
sonar	$m = 1, \sigma = 0.02$	$m = 1, \sigma = 0.04$	$k = 1$	-
tae	$m = 1, \sigma = 0.18$	$m = 1, \sigma = 0.02$	$k = 1$	$k = 1$
vehicle	$m = 1, \sigma = 0.02$	$m = 1, \sigma = 0.02$	$k = 1$	-
wine	$m = 1, \sigma = 0.02$	$m = 2, \sigma = 0.02$	$k = 2$	-
wisconsin	$m = 1, \sigma = 0.14$	$m = 1, \sigma = 0.1$	$k = 1$	-
yeast	$m = 1, \sigma = 0.16$	$m = 1, \sigma = 0.2$	$k = 1$	-

"-" indicates that the strategy is not applicable due to attribute types.

includes evaluating performance on more datasets especially large-scale datasets, and verifying the effectiveness of the proposed framework when using black-box models other than LightGBM as a teacher model.

REFERENCES

[1] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[2] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, and R. Ranjan, "Explainable AI (XAI): Core ideas, techniques, and solutions," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–33, 2023.

[3] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer Science & Business Media, 2004.

[4] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 3–15, Springer, 2018.

[5] J. M. Alonso, C. Castiello, L. Magdalena, and C. Mencar, *Explainable Fuzzy Systems - Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems*, vol. 970. Springer, 2021.

[6] A. Fernandez, F. Herrera, O. Cordon, M. J. Del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?," *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, 2019.

[7] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 45–65, 2013.

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems* (I. Guyon and et al., eds.), vol. 30, Curran Associates, Inc., 2017.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[10] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *International Journal of Approximate Reasoning*, vol. 44, no. 1, pp. 4–31, 2007.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[13] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, pp. 255–287, 2011.