H. Ishibuchi and Y. Nojima, "Handling a training dataset as a black-box model for privacy preserving in fuzzy GBML algorithms," Proc. of 2015 IEEE International Conference on Fuzzy Systems, 8 pages, Istanbul, Turkey, August 2-5, 2015.

Handling a Training Dataset as a Black-Box Model for Privacy Preserving in Fuzzy GBML Algorithms

Hisao Ishibuchi and Yusuke Nojima
Department of Computer Science and Intelligent Systems
Graduate School of Engineering, Osaka Prefecture University
Sakai, Osaka 599-8531, Japan
{hisaoi, nojima}@cs.osakafu-u.ac.jp

***Abstract*—In this paper, we assume that we have two types of datasets for classifier design. One is an in-house dataset which is fully available for classifier design as training data. The other is an external dataset which is kept under a very severe privacy preserving policy. We assume that the available information on the external dataset is only the error rate of a presented classifier. No other information is available such as the number of patterns, attribute values of each pattern, and its class label. Thus, the external dataset can be viewed as a black-box model where the error rate is calculated as an output for an input classifier. In this paper, we discuss how such a black-box type dataset can be utilized in fuzzy genetics-based machine leaning (GBML). We use a hybrid fuzzy GBML algorithm where its Michigan-style part is applied to each individual of a Pittsburgh-style part. Since a fuzzy rule-based classifier is an individual in the Pittsburgh-style part, a black-box type dataset can be utilized for fitness evaluation. Through computational experiments, we examine the effect of using a black-box type dataset in comparison with fuzzy rule-based classifiers design only from a fully available dataset.**

Keywords*—*Fuzzy rules, fuzzy classifiers, fuzzy genetics-based machine learning, privacy preserving, black-box type datasets

I. Introduction

In the current big data era [1]-[3], a huge amount of data are collected and stored. However, in some application fields, the size of the available data is much smaller than the total size of the stored data. This is because those data are often collected separately in a large number of different organizations (e.g., a large number of hospitals in many countries). It is often very difficult for an organization to use the stored data in other organizations. One reason for the difficulty is the heterogeneity of data. Each organization may have a different set of testing equipments, and each country may have different regulations for their use. Another reason is privacy preservation. Many organizations may have a severe privacy preserving policy.

Based on these discussions, we proposed an idea of parallel distributed fuzzy genetics-based machine learning (GBML) from multiple datasets with different missing attributes and a severe privacy preserving policy [4]. It was assumed that each organization had an incomplete dataset with missing attributes together with a severe privacy preserving policy. Each dataset was handled as a black-box model with a classifier as an input and its error rate as an output. An incomplete black-box type dataset used in [4] is illustrated as a black-box in Fig. 1.

This work was partially supported by Grand-in-Aid for Scientific Research (C): KAKENHI (25330292).

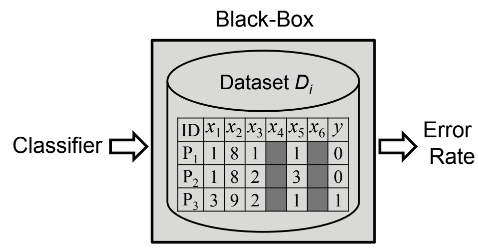

Fig. 1. An incomplete black-box type dataset used in [4].

In Fig. 1, the available information from the dataset was only the error rate of the presented classifier. There was no limit on the number of examined classifiers. All the other information was assumed to be hidden in the black-box. In addition to this black-box property, it was assumed in [4] that some attributes were missing (e.g., x_4 and x_6 in Fig. 1). Then, a parallel distributed fuzzy GBML algorithm [5] was applied to black-box type datasets with missing attributes in [4]. It was shown that the usefulness of incomplete black-box type datasets heavily depended on the number of missing attributes. When many attributes were missing, the use of incomplete black-box type datasets had no positive effects on the performance of designed fuzzy rule-based classifiers.

In this paper, we focus our attention on the utilization of black-box type datasets in fuzzy GBML. This is because the above-mentioned two issues (i.e., the utilization of black-box type datasets and the handling of missing attributes) are too difficult to analyze simultaneously. We assume that we have multiple black-box type datasets and a single fully available dataset. No missing attributes are assumed in those datasets.

This paper is organized as follows. In Section II, we briefly discuss how such a black-box type dataset can be utilized in classifier design. In Section III, we explain the use of black-box type datasets in our hybrid fuzzy GBML algorithm [6]. In Section IV, we explain its parallel distributed implementation [5]. In Section V, we report experimental results to examine the effect of using black-box type datasets in addition to a fully available dataset. Experimental results under the following three assumptions about training data are compared: (i) only a fully available in-house dataset is available, (ii) both a fully available in-house dataset and multiple black-box type external datasets are available, and (iii) all datasets are fully available. In the third setting, both the in-house dataset and the external datasets are assumed to be fully available for comparison. This paper is concluded in Section VI.

II. USE OF BLACK-BOX TYPE DATASETS

Privacy preserving data mining [7]-[11] has been an active research field. Our assumption about black-box type datasets can be viewed as the most severe privacy preserving policy except for a total ban on any case of datasets. To the best of our knowledge, classifier design from black-box type datasets has not been discussed in the literature yet.

Our classifier design problem is shown in Fig. 2 where we have a single fully available dataset (i.e., D_1) and other black-box type datasets (i.e., $D_2, D_3, ..., D_N$). Since no information on attribute values of each pattern in the black-box type datasets is available, almost all supervised and non-supervised learning algorithms are unable to handle them. The simplest approach to our problem in Fig. 2 is to design a classifier using only the fully available dataset D_1. In this approach, our hybrid fuzzy GBML algorithm [6] is executed on D_1 to design a fuzzy rule-based classifier. This approach is referred to as E(D_1)-S(D_1) in Table I where E(D_1) and S(D_1) mean that only D_1 is used for the fitness evaluation at each generation and for the final classifier selection at the final generation, respectively.

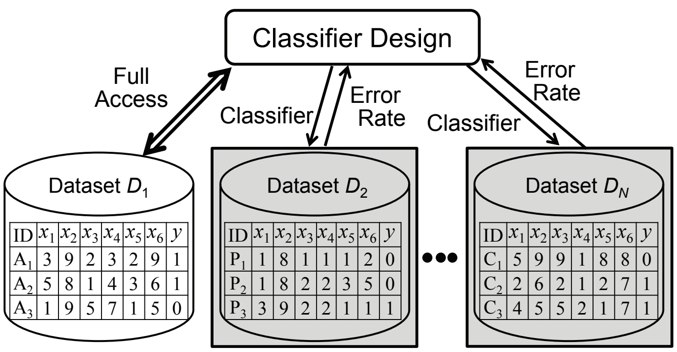

Fig. 2. Our classifier design problem in this paper.

If multiple classifiers are generated from D_1 as candidates for a final classifier, the black-box type datasets can be used for choosing a single classifier from the candidates. The reliability of classifier selection may be improved by using all datasets D_1, $D_2, ..., D_N$ instead of only the fully available dataset D_1. When our hybrid fuzzy GBML algorithm is applied to D_1, a number of fuzzy rule-based classifiers will be included in the final population. In the simplest setting E(D_1)-S(D_1) in Table I, the best classifier with respect to D_1 is selected from the final population as a single final classifier. Another setting is to choose the best classifier from the final population using all datasets $D_1, D_2, ..., D_N$. This setting is referred to as E(D_1)-S(All) in Table I. E(D_1)-S(All) means the use of D_1 for fitness evaluation and the use of all the N datasets for classifier selection. Except for the final classifier selection, E(D_1)-S(D_1) and E(D_1)-S(All) are exactly the same.

The basic structure of our hybrid fuzzy GBML algorithm is Pittsburgh-style GBML where each individual is a fuzzy rule-based classifier. Good individuals are selected to generate new fuzzy rule-based classifiers by crossover and mutation. In the Pittsburgh-style part, the black-box type datasets can be used together with the fully available dataset for evaluating each individual at each generation in the same manner as the above-mentioned classifier selection at the final generation.

TABLE I. USE OF THE FULLY AVAILABLE DATASET D_1 AND THE BLACK-BOX TYPE DATASETS $D_2, D_3, ..., D_N$ IN EACH VARIANT.

Algorithm Variant	Fitness Evaluation	Classifier Selection
E(D_1)-S(D_1)	D_1	D_1
E(D_1)-S(All)	D_1	$D_1, D_2, ..., D_N$
E(All)-S(D_1)	$D_1, D_2, ..., D_N$	D_1
E(All)-S(All)	$D_1, D_2, ..., D_N$	$D_1, D_2, ..., D_N$

We denote the fitness evaluation using all the N datasets by E(All) in Table I. Two variants of our hybrid fuzzy GBML algorithm with E(All) are shown in the last two rows of Table I depending on the setting of final solution selection: E(All)-S(D_1) and E(All)-S(All). Only D_1 is used for final solution selection in E(All)-S(D_1) while all the N datasets are used for final classifier selection in E(All)-S(All). Whereas all the N datasets are used for fitness evaluation at each generation in these two variants, only D_1 is used in almost all the other parts of our hybrid GBML algorithm as explained in the next section. For example, initial fuzzy rule-based classifiers are generated using D_1. When a new fuzzy rule is generated by crossover and mutation, its consequent class and rule weight are specified by D_1. In the Michigan-style part of all the four variants in Table I, only D_1 is always used. Since an individual is a single fuzzy rule in the Michigan-style part, each fuzzy rule instead of a fuzzy rule-based classifier is to be evaluated. However, the black-box type external datasets $D_2, D_3, ..., D_N$ can be used only for classifier evaluation. Thus they cannot be used in the Michigan-style part in all the four variants in Table I.

In principle, evolutionary computation is applicable to any learning and optimization problems if each individual can be evaluated. Since each individual in the Pittsburgh-style part is a fuzzy rule-based classifier, we can apply it to a black-box type dataset D_k ($2 \leq k \leq N$). However, when we cannot use any information on each pattern for initial classifier generation or fuzzy rule specification at each generation, the search ability of our hybrid fuzzy GBML algorithm is severely deteriorated. Thus we modify our hybrid fuzzy GBML algorithm as follows for its application to D_k ($2 \leq k \leq N$): The Michigan-style part is not used, and D_1 is used for initial classifier generation and fuzzy rule specification. If we use the same notation as in Table I, this variant can be denoted as "E(D_k)-S(D_k)". However, we do not use this variant as a separate GBML algorithm because E(D_1)-S(D_1) is likely to outperform E(D_k)-S(D_k). As shown in Section IV, we examine the use of E(D_k)-S(D_k) as a part of a parallel distributed hybrid fuzzy GBML with N islands where E(D_k)-S(D_k) is executed at the kth island (k=1, 2, ..., N).

III. HYBRID FUZZY GBML ALGORITHM [6]

A. Fuzzy Rules for Classification Problems

We use the following fuzzy rule for a pattern classification problem with n attributes [12]:

$$\text{Rule } R_q\text{: If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{1}$$

where R_q is the label of the qth fuzzy rule, $\boldsymbol{x}=(x_1, ..., x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set on the ith attribute x_i (i=1, 2, ..., n), C_q is a consequent class, and

CF_q is a rule weight. A single training pattern $\mathbf{x}_p=(x_{p1}, x_{p2}, ..., x_{pn})$ is used to specify the n antecedent fuzzy sets $A_{q1}, A_{q2}, ..., A_{qn}$ by choosing a fuzzy set with a high compatibility grade for each attribute value x_{pi} [13]. Then each condition is replaced with *don't care* using a pre-specified *don't care* probability. The corresponding consequent class C_q and rule weight CF_q are specified using the compatible training patterns [14].

In this paper, we use fuzzy rules in (1) in order to realize a good compromise between accuracy and interpretability of fuzzy rule-based classifiers. If the accuracy maximization is a dominant objective, it may be a good idea to use fuzzy rules with certainty grades for all classes as in [15], [16]. For further discussions on fuzzy classification rules, see Cordón et al. [17].

B. Fuzzy Rule-Based Classifiers

A fuzzy rule-based classifier is a set of fuzzy rules in (1). Let S be a fuzzy rule-based classifier. When a new pattern is presented to S, the compatibility grade of each fuzzy rule with the pattern is calculated using the product operator. Then a single winner rule is selected using the weighted compatibility (i.e., the product of the compatibility and the rule weight). The new pattern is classified by the single winner rule. If multiple fuzzy rules with different consequent classes are selected as the winner rules (i.e., if those fuzzy rules have the same maximum weighted compatibility), the classification of the new pattern is rejected. The classification is also rejected if no fuzzy rule in S is compatible with the new pattern.

The use of the single winner-based fuzzy reasoning method makes the fitness assignment very easy in the Michigan-style part. This is because we can identify a single responsible fuzzy rule for the classification of each pattern. See Cordón et al. [17] for other types of fuzzy rules and fuzzy reasoning mechanisms for pattern classification problems.

C. Pittsburgh-Style Framework of Our Hybrid Algorithm

The overall structure of our hybrid fuzzy GBML algorithm [6] is shown in Fig. 3. Our algorithm has a Pittsburgh-style basic framework where an individual is a fuzzy rule-based classifier. Offspring (i.e., new fuzzy rule-based classifiers) are generated from the current population by selection, crossover and mutation in the Pittsburgh-style part. A Michigan-style GBML algorithm [13] is probabilistically applied to each of the newly generated fuzzy rule-based classifiers.

When the Michigan-style part is applied to a fuzzy rule-based classifier, all fuzzy rules included in the classifier are used as an initial population. Each step in Fig. 3 is explained in the following.

Initialization: First, an initial population of fuzzy rule-based classifiers is generated from the given training dataset. As we have already mentioned, the antecedent part of an initial fuzzy rule is specified from a randomly selected training pattern. Its consequent part is specified by compatible training patterns with the antecedent part. In our computational experiments, a fuzzy rule-based classifier with 30 fuzzy rules is generated by randomly selected 30 training patterns. This procedure is iterated to generate an initial population of fuzzy rule-based classifiers. In the initialization phase, we need the fully available in-house dataset D_1.

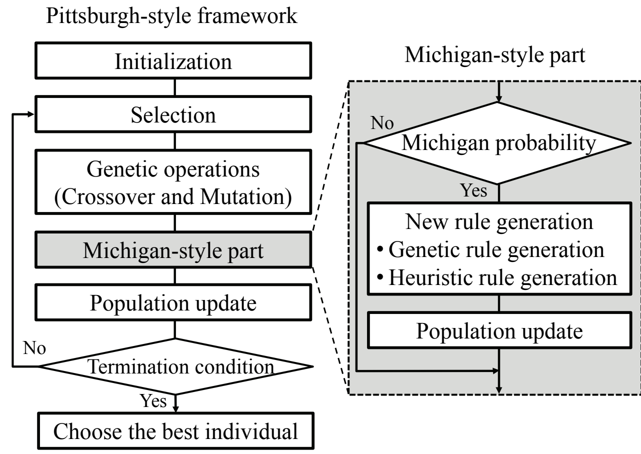

Fig. 3. Structure of our hybrid fuzzy GBML algorithm [5].

Selection: Binary tournament selection is used to choose a pair of parents from the current population. Each fuzzy rule-based classifier is evaluated by the following fitness function:

$$fitness(S) = w_1 f_1(S) + w_2 f_2(S) + w_3 f_3(S), \quad (2)$$

where w_1, w_2 and w_3 are pre-specified non-negative weights, and $f_1(S)$, $f_2(S)$ and $f_3(S)$ are defined as follows:

$f_1(S)$: Training data error rate of S in percentage,

$f_2(S)$: The number of fuzzy rules in S (i.e., $f_2(S) = |S|$),

$f_3(S)$: The total rule length of S.

The rule length of each fuzzy rule is the number of its antecedent conditions excluding *don't care* conditions. The total rule length of S is the same as the total number of antecedent conditions (excluding *don't care* conditions) of all fuzzy rules in S. As in our former studies [4], the weight values are specified as $w_1 = 100$, $w_2 = 1$ and $w_3 = 1$.

In the selection phase, we need only the error rate on the given training dataset. So, we can utilize the black-box type external datasets $D_2, ..., D_N$ as well as the fully available in-house dataset D_1. As we have already explained in Section II, we examine the two options: One is based on only the fully available dataset D_1 (i.e., E(D_1)), and the other is based on all the datasets $D_1, D_2, ..., D_N$ (i.e., E(All)).

Crossover and Mutation: An offspring classifier is constructed by choosing a randomly specified number of fuzzy rules from each parent. The size of each parent can be different. The size of the generated offspring can also be different from the size of each parent. For example, an offspring classifier with 20 rules can be generated from two parents with 30 and 40 rules. This crossover is applied to a pair of parents with a pre-specified crossover probability. Mutation is used to randomly replace an antecedent fuzzy set with another one (including *don't care*). When the antecedent part of a fuzzy rule is changed by the mutation, the consequent class and the rule weight are updated using the compatible training patterns. Thus we need the fully available in-house dataset D_1 when we generate new fuzzy rule-based classifiers.

Michigan-Style Part: The Michigan-style part is explained in the next subsection.

Population Update: Let N_{pop} be the population size. We generate N_{pop} offspring classifiers. Then we choose the best N_{pop} classifiers from the current population of size N_{pop} and the offspring population of size N_{pop}. That is, we use the $(\mu+\lambda)$-ES mechanism with $\mu=\lambda$. Each fuzzy rule-based classifier is evaluated by the fitness function in (2). As in the selection phase, we can utilize the black-box type datasets as well as the fully available dataset in the population update phase.

Termination Condition: We use a pre-specified number of generations as the termination condition in this paper.

Final Classifier Selection: The best classifier with respect to the fitness function in (2) is selected as the final result from the final population. We examine the following two options: One is based on only D_1 (i.e., S(D_1)), and the other is based on all datasets $D_1, D_2, ..., D_N$ (i.e., S(All)).

D. *Michigan-Style Part of Our Hybrid Algorithm*

A single iteration of a Michigan-style algorithm is applied to each offspring with a pre-specified probability in our hybrid algorithm [6]. An offspring in the Pittsburgh-style part is used as an initial population in the Michigan-style part. Each rule is evaluated as follows. First, each pattern in the given training dataset is classified using the population of fuzzy rules. A single winner rule is identified for the classification of each pattern. When a pattern is correctly classified, one point is added to the winner rule. After the classification of all training patterns, the fitness of each rule is defined by the number of correctly classified training patterns by the rule.

Since the classification result of each pattern is needed for fitness evaluation, we cannot use any black-box type datasets in the Michigan part. Only the fully available dataset D_1 is usable in the Michigan part. Actually, D_1 is needed almost everywhere in the Michigan part. For example, the antecedent part of a new rule is generated by uniform crossover of the two parent rules and mutation. D_1 is needed to specify the consequent class and the rule weight of the generated rule. A new rule is also generated from a misclassified or rejected training pattern. The attribute values of the pattern are needed to specify the antecedent part of the new rule. Then D_1 is needed to specify the consequent class and the rule weight.

IV. Parallel Distributed Implementation

A parallel distributed implementation of our hybrid fuzzy GBML algorithm [6] was proposed in [5] to significantly decrease its computation time. As shown in Fig. 4, the population of fuzzy rule-based classifiers is divided into N sub-populations ($N=7$ in Fig. 4). The given training dataset is also divided into N subsets. As a result, the computation load at each CPU becomes $1/N^2$ of the standard implementation.

The assigned training data subsets are periodically rotated over the CPUs to avoid the overfitting of each sub-population to a particular subset. Periodical migration of a copy of the best individual in each sub-population is also performed. An interesting trick is to perform the migration and the training data rotation in the opposite directions as shown in Fig. 4. It was reported in [5] that the computation time was decreased by seven CPUs to 1/44 of the standard non-parallel non-distributed implementation on average over nine benchmark problems without clear deterioration in the generalization ability of the designed fuzzy rule-based classifiers.

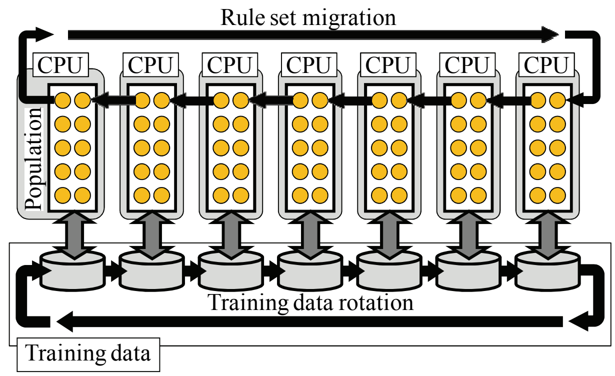

Fig. 4. Parallel distributed implementation [5].

Let us consider the parallel distributed implementation of our hybrid fuzzy GBML algorithm for our classifier design problem in Fig. 2 where D_1 is a fully available in-house dataset and $D_2, ..., D_N$ are black-box type external datasets. We assume that the number of CPUs is the same as the number of datasets (i.e., N). First, we focus on the implementation of the fitness evaluation of each individual using all datasets (i.e., the implementation of the E(All) variants in Section II) using the parallel distributed model in Fig. 4.

This can be easily implemented as follows. An initial population of fuzzy rule-based classifiers is generated using the fully available dataset D_1. The generated population is divided into N sub-populations of the same size and assigned to N CPUs. In the first iteration, D_N is assigned to the first CPU. Each individual at the first CPU is evaluated using D_N. In the second iteration, D_N is rotated to the second CPU, and D_{N-1} is assigned to the first CPU. Each individual at the first and second CPUs is evaluated using the assigned training data subset to each CPU. In the third iteration, each of D_N and D_{N-1} is rotated to the next CPU, and D_{N-2} is assigned to the first CPU. Each individual at each CPU is evaluated using the assigned training data subset. In this manner, we iterate the assignment of a training data subset, the rotation of the assigned training data subsets, and the evaluation of each individual at each CPU.

In the Nth iteration, the kth CPU has the kth training data subset D_k. After the evaluation of each individual, a single generation update step of our hybrid fuzzy GBML algorithm is performed on the first CPU with D_1. The fitness of each individual is defined by the evaluation results over all training data subsets. After the generation update is completed at the first CPU with D_1, all the assigned data subsets are rotated in the (N+1)th iteration. After the evaluation of each individual, the generation update is performed at the second CPU with D_1. In this manner, the generation update at each CPU is performed only when D_1 is assigned. At each CPU with one of the other (N–1) training data subsets, the CPU is used only for the evaluation of each individual using the assigned subset. That is, the generation update at each CPU is performed every N iterations. The computation load at each CPU is $1/N$ in comparison with a standard non-parallel non-distributed case.

This is because the number of individuals evaluated at each CPU is $1/N$ of the entire population while each individual is evaluated using all the training data sets. Actually this implementation is the same as an island model with N islands where all training data subsets are assigned to each island. No migration is performed in this implementation. In this case, it is clear that the computation load at each CPU is $1/N$.

The E(All)-S(D_1) and E(All)-S(All) variants in Section II are executed in the above-mentioned manner. The other variants (i.e., E(D_1)-S(D_1) and E(D_1)-S(All)) in Section II are performed as a standard non-parallel non-distributed algorithm since only D_1 is used for the fitness evaluation (i.e., since the computation load is not so heavy).

Next we discuss a more straightforward parallel distributed implementation for our classifier design problem in Fig. 2. An initial population of fuzzy rule-based classifiers is generated using D_1. The generated population is divided into N sub-populations. Those sub-populations are assigned to N CPUs. The training data subsets are also assigned to the N CPUs. At the CPU with D_1, our hybrid fuzzy GBML algorithm is performed with no modification. At the other CPUs with a black-box type dataset D_k ($2 \leq k \leq N$), we do not use the Michigan-style part. At each CPU, the assigned training data subset D_k is used for fitness evaluation while D_1 is always used for fuzzy rule specification. All the training data subsets are periodically rotated over the N CPUs.

V. Computational Experiments

We use ten classification problems in Table II from the UC Irvine Machine Learning Repository. The ten-fold cross-validation is used to calculate the average classification rate on test data. The ten-fold cross-validation is iterated five times (i.e., 5×10CV) in almost all experiments. The training data are randomly divided into seven subsets of the same size to specify the seven datasets (i.e., D_1, D_2, ..., D_7). In the last column, we show the size of D_1 for each problem.

TABLE II. Ten Classification Problems in This Paper.

Problem	Number of Patterns	Number of Attributes	Number of Classes	Average Size of D_1
Glass	214	9	6	28
Heart	270	13	2	35
Pima	768	8	2	99
Newthyroid	215	5	3	28
Wine	178	13	3	23
Wisconsin	683	9	2	88
Segment	2,310	19	7	297
Phoneme	5,404	5	2	695
Page-blocks	5,472	10	5	704
Satimage	6,435	36	6	827

Computational experiments are performed in the same settings as in our former study [5]:

Population size: 210 (Subpopulation size: 30),
Number of sub-populations (N): 7,
Termination condition: 50,000 generations,
Weight vector: $(w_1, w_2, w_3) = (100, 1, 1)$,
Training data rotation interval: 1, 10, 100, 1000,
Migration interval: 100,
Don't care probability: $(n-5)/n$ for n-dimensional problems,
The number of fuzzy rules in each initial classifier S: 30,
Upper limit on the number of fuzzy rules in S: 60,
In Pittsburgh-style part: Crossover probability: 0.9,
Mutation probability: $1/(n|S|)$,
In Michigan-style part: Crossover probability: 0.9,
Mutation probability: $1/n$.

The four settings of the training data rotation interval are examined for the parallel distributed implementation. For example, the data rotation interval 1000 means that the data rotation is performed at every 1000 generations. In other words, the hybrid fuzzy GBML algorithm is locally performed at each CPU using the same training data subset for 1000 generations between the training data rotations. The migration interval is always specified as 100 (migration at every 100 generations).

In Table III, we show the average classification rate on test data by each variant in Section II for each problem. The best result for each problem is highlighted by boldface. We performed the Friedman test to examine whether there exist significant differences among four variants or not. The obtained p-value was 2.328×10^{-6}. Then, we applied the Shaffer's post-hoc procedure to perform all pairwise comparisons of four variants. Tables IV and V show the Friedman rankings and the adjusted p-values by the Shaffer's procedure, respectively. We used the software available from http://sci2s.ugr.es/sicidm/ [18].

Table V shows that the difference between E(D_1)-S(D_1) and E(D_1)-S(All) was not statistically significant. That is, the use of all training datasets for the final classifier selection has almost no effect when D_1 is used for fitness evaluation. We can observe the same effect from the comparison between E(All)-S(D_1) and E(All)-S(All).

Table V also shows that the difference between E(D_1)-S(D_1) and E(All)-S(D_1) was statistically significant. The difference between E(D_1)-S(All) and E(All)-S(All) was also statistical significant. It is clear that the use of all training data subsets for fitness evaluation has a much larger positive effect on the generalization ability of the designed classifiers than the use of all training data subsets for final classifier selection. The best results in Table III are obtained by E(All)-S(All) where all training data subsets are used for fitness evaluation and classifier selection. These observations show that the use of the black-box type datasets can improve the performance of our hybrid fuzzy GBML algorithm.

As shown in the last column of Table II, the size of D_1 is very small in the first six classification problems. Thus, it is very difficult to design fuzzy rule-based classifiers with high generalization ability using only D_1 for those classification problems. A large increase in the average classification rate on test data by the use of the black-box type training data subsets is observed for each of the first six problems in Table III.

The size of D_1 is not so small in the last three classification problems in Table II (i.e., 695-827 patterns). Even for those large-size problems, we can observe an increase in the average classification rate on the test data by the use of the black-box type training data subsets in Table III (e.g., a 3% increase from 81.28% to 84.28% for Phoneme).

TABLE III. EFFECT OF THE USE OF THE BLACK-BOX TYPE DATASETS IN THE FITNESS EVALUATION E(ALL) AND THE FINAL SOLUTION SELECTION S(ALL).

Problem	E(D_1)-S(D_1)	E(D_1)-S(All)	E(All)-S(D_1)	E(All)-S(All)
Glass	54.50	55.08	62.02	**65.34**
Heart	70.81	70.89	78.22	**80.07**
Pima	69.59	69.46	74.02	**74.20**
Newthyroid	87.20	87.56	91.85	**94.61**
Wine	84.75	85.07	89.93	**93.80**
Wisconsin	93.53	93.86	95.96	**95.99**
Segment	89.77	89.79	92.78	**93.21**
Phoneme	81.28	81.31	83.76	**84.28**
Page-blocks	95.22	95.26	96.10	**96.20**
Satimage	83.50	83.54	84.91	**85.13**
Average	81.02	81.18	84.95	**86.28**

TABLE IV. FRIEDMAN RANKING.

Variant	Friedman ranking
E(D_1)-S(D_1)	3.90
E(D_1)-S(All)	3.10
E(All)-S(D_1)	2.00
E(All)-S(All)	1.00

TABLE V. ADJUSTED *P*-VALUES BY THE SHAFFER'S TEST.

Hypothesis	Adjusted p-value
E(D_1)-S(D_1) vs. E(All)-S(All)	3.053×10^{-6}
E(D_1)-S(All) vs. E(All)-S(All)	8.265×10^{-4}
E(D_1)-S(D_1) vs. E(All)-S(D_1)	0.003
E(D_1)-S(All) vs. E(All)-S(D_1)	0.170
E(All)-S(D_1) vs. E(All)-S(All)	0.170
E(D_1)-S(D_1) vs. E(D_1)-S(All)	0.170

In Table VI, we show experimental results by the parallel distributed implementation of our hybrid fuzzy GBML algorithm for the fully available dataset D_1 and the black-box type datasets D_2, ..., D_7. The four settings of the training data rotation interval are examined in Table VI. For example, (100) means that the training data rotation is performed every 100 generations. All training data subsets are used for final classifier selection in Table VI. The best results are obtained in Table VI when the data rotation is performed at every generation (i.e., (1) in Table VI) on average. In order to confirm the statistical difference among four setting, we performed the Friedman test for Table VI. The obtained p-value was 5.927×10^{-5}. Then, we applied the Holm's post-hoc procedure to compare the best setting (i.e., (1) in Table VI)) as the control algorithm with the remaining ones. Table VII shows the Friedman rankings and the adjusted p-values by the Holm's procedure. Table VII shows that the difference between the best setting (i.e., (1) in Table VI) and the second best one (i.e., (10) in Table VI) was not statistical significant, while the differences between the best setting and the worst two settings others (i.e., (100) and (1000) in Table VI) were statistical significant. That is, the frequent training data rotation improved the generalization ability.

TABLE VI. EXPERIMENTAL RESULTS OF THE PARALLEL DISTRIBUTED IMPLEMENTATION FOR THE FULLY AVAILABLE DATASET D_1 AND THE BLACK-BOX TYPE DATASETS D_2, ..., D_7.

Problem	Parallel Distributed (Data Rotation Interval)			
	(1)	(10)	(100)	(1000)
Glass	**63.28**	61.56	57.19	56.53
Heart	**80.52**	77.56	73.85	76.00
Pima	**75.63**	74.43	73.36	73.23
Newthyroid	**94.33**	93.68	91.00	89.24
Wine	**93.26**	92.61	91.26	89.47
Wisconsin	**96.58**	96.34	95.58	95.32
Segment	91.85	**93.77**	91.56	91.39
Phoneme	**82.88**	82.76	81.41	81.94
Page-blocks	95.76	**95.91**	95.72	95.65
Satimage	83.86	**85.86**	83.19	84.39
Average	**85.79**	85.45	83.41	83.31

TABLE VII. FRIEDMAN RANKING AND ADJUSTED *P*-VALUES BY THE HOLM'S TEST WITH THE BEST SETTING ((1) IN TABLE VI) AS THE CONTROL ALGORITHM.

Setting	Friedman ranking	Adjusted p-value
Every generation	1.40	-
10 generations	1.70	0.603
100 generations	3.30	0.002
1000 generations	3.60	4.16×10^{-4}

From the comparison between Table III and Table VI, we can see that better results are obtained in almost all settings in Table VI than the first two variants with E(D_1) in Table III. This observation suggests the usefulness of using all training data subsets (including the black-box type training data subsets) for fitness evaluation in the Pittsburgh-style part.

For comparison, we report experimental results using all the training data as fully available data. That is, we assume that all of the seven training data subsets D_1, D_2, ..., D_7 are fully available (whereas we assumed that D_2, ..., D_7 are black-box type training data subsets in Table III and Table VI). Experimental results are summarized in Table VIII. The second column of Table VIII shows the experimental results by the standard non-parallel non-distributed implementation of our hybrid fuzzy GBML algorithm [6]. Due to its heavy computation load, the average classification rate on test data is calculated over three iterations of the ten-fold cross-validation (3×10CV) for the last four problems with more than 2000 patterns. 5×10CV is used for all the other problems. The last four columns show the results by the parallel distributed implementation with different settings of the training data rotation interval. As reported in [5], similar average classification rates on test data are obtained from the standard algorithm in the second column and the parallel distributed implementation in the other columns (especially when the data rotation interval is 1).

TABLE VIII. EXPERIMENTAL RESULTS UNDER THE ASSUMPTION THAT ALL TRAINING DATASETS $D_1, D_2, \ldots, D_7$ ARE FULLY AVAILABLE.

Problem	Standard Algorithm	Parallel Distributed (Data Rotation Interval)			
		(1)	(10)	(100)	(1000)
Glass	**67.88**	65.61	61.24	61.25	58.00
Heart	78.37	**81.78**	79.48	78.00	77.11
Pima	75.31	**75.34**	75.29	74.40	73.47
Newthyroid	94.42	**94.90**	93.58	91.65	91.94
Wine	**94.18**	93.82	93.61	90.02	89.31
Wisconsin	95.96	**96.31**	95.93	95.43	95.32
Segment	94.01	92.77	**94.16**	94.02	93.13
Phoneme	**84.57**	83.43	84.55	84.09	83.43
Page-blocks	96.19	95.94	**96.22**	96.31	96.11
Satimage	84.47	84.16	85.94	**86.89**	85.77
Average	**86.54**	86.41	86.00	85.21	84.36

TABLE IX. FRIEDMAN RANKING AND ADJUSTED *p*-VALUES BY THE HOLM'S TEST WITH THE STANDARD METHOD AS THE CONTROL ALGORITHM.

Algorithm	Friedman ranking	Adjusted *p*-value
Standard	2.20	-
PD (1)	2.75	0.873
PD (10)	2.50	0.873
PD (100)	3.10	0.609
PD (1000)	4.45	0.006

In order to confirm the statistical difference among five models, we performed the Friedman test for Table VIII. The obtained p-value was 0.016. The difference was statistically significant but not highly significant. Then, we applied the Holm's post-hoc procedure to compare the best model (i.e., "standard algorithm" in Table VIII)) as the control algorithm with the remaining ones. Table IX shows the Friedman rankings and the adjusted p-values by the Holm's procedure. The differences between the standard algorithm and the others were not statistically significant except for the parallel distributed model with infrequent training data rotation (i.e., (1000) in Table VIII).

The comparison between Table VI and Table VIII shows that the difference in the average classification rates between the two tables is small. For example, the best overall average classification rates in Table VI and Table VIII are 85.79% and 86.54%, respectively. To statistically confirm this difference, we performed the Wilcoxon signed-rank test. The obtained p-value was 0.202. Thus, this difference is not statistically significant. The best overall average classification rate 86.28% in Table III is also similar to the best result 86.54% in Table VI. We also examined this difference by the Wilcoxon signed-rank test. The obtained p-value was 0.508. However, if no black-box type datasets are available (i.e., only D_1 is available), the average classification rate 81.02% by E(D_1)-S(D_1) in Table III is much smaller than the above-mentioned results obtained using the black-box type training data subsets $D_2, \ldots, D_7$ in addition to the fully available training data subset D_1 (i.e., see Table IV).

In our computational experiments, 85.7% of the training data (i.e., six out of the seven training data subsets) are assumed to be black-box type datasets. It is very interesting to observe that similar results are obtained from the following two settings: 100% of the training data are fully available in Table VII and 85.7% of the training data are black-box type training data subsets in Table III and Table VII. It is also interesting to observe that good results are not obtained when we use only 14.3% of the training data. These observations show that the use of the black-box type datasets has a positive effect on the generalization ability of designed classifiers.

VI. CONCLUSIONS

In this paper, we examined the effect of using black-box type datasets on the generalization ability of fuzzy rule-based classifiers designed by our hybrid fuzzy GBML algorithm. We assumed that the available information on a black-box type dataset was only the error rate for a presented classifier. We explained how such a black-box type dataset can be utilized in our hybrid fuzzy GBML algorithm and its parallel distributed implementation. In our computational experiments, we compared the following three settings about the available training data: (i) 100% of the training data were fully available, (ii) 14.3% of the training data were fully available and the other 85.7% were totally unavailable, and (iii) 14.3% of the training data were fully available and the other 85.7% were available as black-box type training data subsets. We obtained the following observations:

1. Good results were not obtained from the use of only 14.3% fully available training data,
2. Similar results were obtained from the following two cases: 100% fully available training data, and 85.7% black-box type and 14.3% fully available training data.

These observations suggest that the increase in the amount of available training data may improve the generalization ability of designed classifiers even when only the amount of available training data in black-box type datasets is increased. Moreover, our experimental results do not suggest any large difference in the importance between fully available datasets and black-box type datasets with respect to the generalization ability of designed fuzzy rule-based classifiers.

One interesting future research topic is classifier design only from black-box type datasets (whereas this task may be very difficult). Another interesting research topic is the performance evaluation of designed classifiers using various combinations of the amount of fully available training datasets and the amount of black-box type datasets. It may also be interesting to discuss the implementation of multiobjective classifier design algorithms for black-box type datasets.

ACKNOWLEDGMENT

The authors would like to thank Mr. Masakazu Yamane and Mr. Yuji Takahashi for their help with computational experiments.

REFERENCES

[1] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging "Big dimensionality"," *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 14-26, August 2014.

[2] P. Huijse, P. A. Estevez, P. Protopapas, J. C. Principe, and P. Zegers, "Computational intelligence challenges and applications on large-scale astronomical time series databases," *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 27-39, August 2014.

[3] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62-74, November 2014.

[4] H. Ishibuchi, M. Yamane, and Y. Nojima, "Learning from multiple data sets with different missing attributes and privacy policies: Parallel distributed fuzzy genetics-based machine learning approach," *Proc. of IEEE Big Data 2013 Workshop on Scalable Machine Learning: Theory and Applications*, pp. 63-70, Santa Clara, October 6-9, 2013.

[5] H. Ishibuchi, S. Mihara, and Y. Nojima, "Parallel distributed hybrid fuzzy GBML models with rule set migration and training data rotation," *IEEE Trans. on Fuzzy Systems*, vol. 21, no. 2, pp. 355-368, April 2013.

[6] H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Hybridization of fuzzy GBML approaches for pattern classification problems," *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 35, no. 2, pp. 359-365, April 2005.

[7] R. Agrawal and R. Srikant, "Privacy-preserving data mining," *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data,* pp. 439-450, Dallas, May 15-18, 2000.

[8] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Proc. of the 20th Annual International Cryptology Conference (LNCS 1880: Advances in Cryptology - CRYPTO 2000)*, pp. 36-54, Santa Barbara, August 20-24, 2000.

[9] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177-206, June 2002.

[10] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *Sigmod Record*, vol. 33, no. 1, pp. 50-57, March 2004.

[11] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1026-1037, September 2004.

[12] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, no. 1, pp. 21-32, November 1992.

[13] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 29, no. 5, pp. 601-618, October 1999.

[14] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, pp. 428-435, August 2005.

[15] S. K. Pal and D. P. Mandal, "Linguistic recognition system based on approximate reasoning," *Information Sciences*, vol. 61, no. 1-2, pp. 135–161, April 1992.

[16] D. P. Mandal, C. A. Murthy, and S. K. Pal, "Formulation of a multivalued recognition system," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 607-620, July/August 1992.

[17] O. Cordón, M. J. del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, vol. 20, no. 1, pp. 21-45, January 1999.

[18] S. García and F. Herrera, "An Extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparison," *Journal of Machine Learning Research*, vol. 9, pp. 2677-2694, 2008.