Fuzzy Classifiers with a Two-Stage Reject Option

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

Koyo Kawano
Graduate School of Engineering
Osaka Prefecture University
Sakai, Japan
sbb01056@st.osakafu-u.ac.jp

Hajime Shimahara
Collage of Engineering
Osaka Prefecture University
Sakai, Japan
seb01064@st.osakafu-u.ac.jp

Eric Vernon
Graduate School of Informatics
Osaka Metropolitan University
Sakai, Japan
sn22864k@st.omu.ac.jp

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

Hisao Ishibuchi
Southern University of
Science and Technology
Shenzhen, China
hisao@sustech.edu.cn

***Abstract*—In general, fuzzy classifiers have high interpretability. They can linguistically explain the reason why an input pattern is classified as a particular class through the linguistic interpretation of each antecedent fuzzy set. A reject option that rejects patterns near the boundaries between different classes is an approach to increase the reliability of classifiers. However, the conventional threshold-based reject option may reject more patterns than necessary to achieve high reliability. In this paper, we propose a two-stage reject option where a machine learning model is used after the threshold-based decision by a fuzzy classifier. If the class labels predicted by the machine learning model and the fuzzy classifier are the same, the fuzzy classifier outputs the predicted class label without rejection. Through computational experiments, we discuss the trade-off relation between the accuracy and rejection rate by the proposed two-stage reject option using various machine learning models.**

***Index Terms*—Fuzzy classifiers, genetics-based machine learning, reject options**

I. Introduction

Over recent years, AI technology has found increasing applications in various fields such as medicine and finance, where misclassification can significantly harm users. Therefore, AI systems need to exhibit not only high classification accuracy but also high explainability and reliability of classification results [1]. Fuzzy classifiers, which comprise multiple fuzzy rules, represent an example of AI that offers high explainability [2], [3]. To efficiently obtain fuzzy classifiers with high explainability, researchers have developed Fuzzy Genetics-based Machine Learning (FGBML) [4], which utilizes evolutionary optimization techniques. One of the main challenges of AI is the reliability enhancement of classification results. This highlights the need to improve AI systems to ensure that they can be trusted to accurately and reliably classify unseen patterns in real-world applications.

To increase the reliability of classifiers, researchers have proposed a reject option that rejects the classification of patterns estimated to have unreliable classification results near the classification boundaries between different classes [5]. However, conventional threshold-based reject options may reject more patterns than necessary, reducing the classifier's efficiency [6], [7]. Therefore, novel techniques for using a reject option under a good balance between the classification accuracy and rejection rate are required. By adopting such techniques, we can improve the efficiency and reliability of classifiers in real-world applications.

Fuzzy classifiers with reject options have been studied [8]–[11] but are not well established yet. In this study, we focus on the above-mentioned issue and propose a two-stage reject option to improve the reliability of fuzzy classifiers in real-world applications. The first stage is a threshold-based rejection where a fuzzy rule-based classifier is used. If the input pattern is rejected, a machine learning (ML) model also predicts a class label in the second stage as a second opinion. If the class labels predicted by the fuzzy classifier and the ML model are the same, its rejection is withdrawn and the fuzzy classifier outputs the predicted class label (which is the same as the classification result by the ML model). We evaluate the proposed two-stage reject option and compare its accuracy-rejection trade-off curves to the conventional threshold-based rejection. To validate our approach, we use eight real-world datasets. The results demonstrate that our two-stage reject option achieves a better trade-off between classification accuracy and rejection rate compared to the conventional threshold-based reject options.

This paper is organized as follows. Section II briefly explains fuzzy classifiers and FGBML. Section III describes the conventional threshold-based reject options and the proposed two-stage reject option introduced in the fuzzy classifier. Section IV shows computational experiments. Finally, Section V concludes this paper.

II. Michigan-Style Fuzzy Classifier Design

A. *Fuzzy Classifiers*

Assume n-dimensional patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn}), p = 1, 2, ..., m$. A fuzzy classifier S is composed of a number of

This work was partially supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 19K12159.

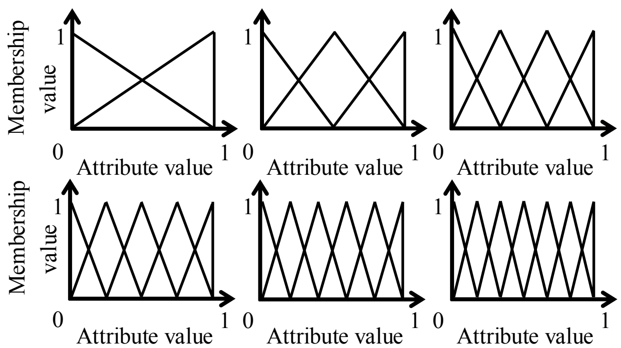


Fig. 1: Triangular membership functions used in this paper

fuzzy if-then rules defined by (1) [2]:

$$\text{Rule } R_q : \text{If } x_{p1} \text{ is } A_{q1} \text{ and ... and } x_{pn} \text{ is } A_{qn} \quad \text{then Class } C_q \text{ with } CF_q, \tag{1}$$

where R_q represents the q-th rule label in S. $\mathbf{A}_q = (A_{q1}, A_{q2}, ..., A_{qn})$, C_q, and CF_q represent the antecedent membership functions, the class label, and the rule weight of the q-th rule, respectively. We use 27 homogeneous triangular membership functions shown in Fig. 1 together with a "*don't care*" as antecedent conditions.

The consequent class label C_q and the rule weight CF_q are automatically determined by the confidence values for the training data. The confidence value $c(\mathbf{A}_q \Rightarrow \text{Class } h)$ for the class h $(h = 1, 2, ..., M)$ is calculated as follows:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{\sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}, \tag{2}$$

where $\mu_{\mathbf{A}_q}(\mathbf{x}_p)$ is the compatibility grade for the p-th pattern. We use the product operator as follows:

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \prod_{i=1}^{n} \mu_{A_{qi}}(x_{pi}), \tag{3}$$

$\mu_{A_{qi}}(x_{pi})$ is the membership degree of the i-th attribute. The consequent class label C_q can be defined by the class label with the maximum confidence value like:

$$C_q = \underset{h=1,2,..,M}{\arg\max}\ c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{4}$$

The consequent rule weight CF_q can be defined as follows:

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \tag{5}$$

The effects of CF_q and other specifications can be found in the literature [2].

For unseen patterns, we use a single-winner method where only a single rule is used for classifying each pattern. That is, we can easily explain why an input pattern is classified as a particular class by reading the antecedent conditions of the winner rule. To do that, we first calculate the product $\eta(R_q)$ of the compatibility grade and the rule weight by:

$$\eta(R_q) = \mu_{\mathbf{A}_q}(\mathbf{x}) \cdot CF_q, \tag{6}$$

Then, the winner rule R_{win} is selected as follows:

$$R_{\text{win}} = \underset{R_q \in S}{\arg\max}\ \{\eta(R_q)\}. \tag{7}$$

The class label of R_{win} is used for the classification of $\mathbf{x}$. If multiple rules with different class labels have the same maximum $\eta(R_q)$ value, the classification of $\mathbf{x}$ is regarded as a misclassification.

B. Michigan-style Fuzzy Genetics-Based Machine Learning

In this paper, we use the Michigan-style FGBML with heuristic rule generation from multiple patterns [12]. It is, of course, possible to use other optimization methods instead. A comparison of base fuzzy classifiers obtained by different optimization methods is not a focus of this paper. It can be future work.

The procedure of our Michigan-style FGBML is shown in Algorithm 1. Each individual in a population represents the antecedent conditions of a single rule. Since the consequent class and rule weight can be automatically determined based on the training data, a fuzzy if-then rule can be generated from each individual. Thus, the population can be regarded as a fuzzy classifier.

1) Initialization: We use heuristic rule generation from multiple patterns. Please refer [12] for the detailed algorithm. A pre-specified number of patterns are randomly selected from the training data for generating a single rule. The antecedent conditions are probabilistically chosen from 28 membership functions in order to well cover the selected patterns.

2) Evaluation: To evaluate each individual (i.e., each rule), all the training patterns are classified by the population P (i.e., fuzzy classifier). The fitness of each individual is calculated as follows:

$$fitness(R_q) = NCP(R_q) - w_{\text{pen}} \cdot MCP(R_q). \tag{8}$$

where $NCP(R_q)$ and $MCP(R_q)$ are the number of correctly classified patterns and the number of misclassified patterns, respectively. w_{pen} is a penalty factor for the misclassifications. The accuracy of the classifier is also recorded.

3) Offspring generation: 50% of offspring individuals are generated by genetic operations while the remaining 50% are generated by heuristic rule generation from multiple patterns. In heuristic rule generation, H misclassified patterns are randomly selected. If the number of misclassified patterns is less than H, patterns are selected from the whole training data.

4) Generation update: Low-fitness individuals are replaced with offspring. Then, a new population (i.e., classifier) is temporarily generated and its accuracy is calculated by classifying the training data. If the accuracy of the newly-generated population is higher than that of the previous one, the population is updated by the new one. If not, the previous population is continuously used in the next generation.

Algorithm 1: Michigan-style FGBML

Input: Training data D, termination generation T
Output: A fuzzy classifier S

Initialize population P_0 from D by heuristic rule generation
Evaluate P_0.
for $t = 1$ *to* T **do**
  Generate offspring fuzzy rules R from P_{t-1} by genetic operations.
  Evaluate offspring rules.
  Make P_t from P_{t-1} and offspring rules.
  if *accuracy* of P_{t-1} is higher than that of P_t **then**
    $P_t \leftarrow P_{t-1}$
Output P_t as a fuzzy classifier S

Algorithm 2: Threshold Optimization

Input:
Fuzzy Classifier: S,
Parameters: k_{max}, t,
Constraint: RR_{max}
Output: Thresholds $\mathbf{T} = (T_1, ..., T_M)$

1 $\mathbf{T} \leftarrow$ Initialize all values are equal to zero
2 $f_{\text{acc}} \leftarrow Accuracy(S)$
3 $f_{\text{reject}} \leftarrow 0.0$
4 **while do**
5   $T_i \leftarrow$ update the threshold value which maximize $\Delta f_{\text{acc}}/\Delta f_{\text{reject}}$, subject to *RejectRate* is smaller than RR_{max}.
6   **if** f_{acc} == $Accuracy(S, \mathbf{T})$ **then**
7   break
8   $f_{\text{acc}} \leftarrow Accuracy(S, \mathbf{T})$
9   $f_{\text{reject}} \leftarrow RejectRate(S, \mathbf{T})$
10 **return** $\mathbf{T}$

III. Reject Options

A. Single Threshold-based Reject Option

In [6], Chow demonstrated that by rejecting classification results with a posterior probability below a certain threshold, an optimal trade-off curve between classification accuracy and rejection rate can be achieved. According to eq. (9), the posterior probability of the classification for $\mathbf{x}_p$ is more than or equal to the threshold T, the class label C is output. Otherwise, the classification is rejected.

$$y = \begin{cases} C & \text{if } P(C|\mathbf{x}_p) \geq T, \\ \text{reject} & \text{otherwise.} \end{cases} \tag{9}$$

This method is referred to as ST reject option in this paper.

B. Class-Wise Thresholds-based Reject Option

The Chow's rule is commonly used to obtain an optimal tradeoff between error and reject probabilities. However, the Chow's rule is not optimal when the a posteriori probabilities are subject to errors. Fumera et al. [7] proposed the use of multiple reject thresholds that are associated with the data classes. It is referred to as Class-Wise-Thresholds (CWT). Adopting CWT for the reject rule was demonstrated to obtain a better error-rejection trade-off compared to the Chow's rule. In CWT reject option, the threshold T_C is set for each class.

$$y = \begin{cases} C & \text{if } P(C|\mathbf{x}_p) \geq T_C, \\ \text{reject} & \text{otherwise.} \end{cases} \tag{10}$$

C. Reject Option for Fuzzy Classifiers

In this paper, we propose the certainty degree δ defined in eq. (12) which is used in threshold-based reject option methods instead of the a posteriori probabilities $P(C|\mathbf{x}_p)$.

$$\delta = \frac{\eta(R_{\text{win}}) - \eta(R_{\text{second}})}{\eta(R_{\text{win}})}, \tag{11}$$

$$\delta = \frac{\eta(R_{\text{win}}) - \eta(R_{\text{second}})}{\eta(R_{\text{win}})}, \tag{12}$$

where $\eta(R_{\text{win}})$ is the product value of the compatibility grade and the rule weight of the winner rule R_{win} in eq. (6). $\eta(R_{\text{second}})$ is the product value of the second winner rule which has a different class label from the winner rule. The certainty degree δ becomes small near the decision boundary.

D. Optimization of Rejection Thresholds

To find the optimal threshold values $\mathbf{T} = (T_1, ..., T_M)$ under the maximum rejection rate RR_{max}, we assume the following constraint optimization problem [13]:

$$\begin{aligned} &\text{Maximize} && Accuracy(S, \mathbf{T}), \\ &\text{subject to} && RejectRate(S, \mathbf{T}) \leq RR_{\text{max}}, \end{aligned} \tag{13}$$

where $Accuracy(S, \mathbf{T})$ is the ratio of correctly classified patterns to the data without ones rejected by $\mathbf{T}$. $RejectRate(S, \mathbf{T})$ is the ratio of the rejected patterns to the data, respectively. In general, $Accuracy$ and $RejectRate$ increase as the thresholds become larger.

Under the specified RR_{max}, we gradually increase the thresholds until the accuracy improvement stops as shown in Algorithm 2. In line 5 of Algorithm 2, each of M thresholds is updated using eq. (14). Δ represents the amount of change.

$$T_i = T_i + k \times t, \tag{14}$$

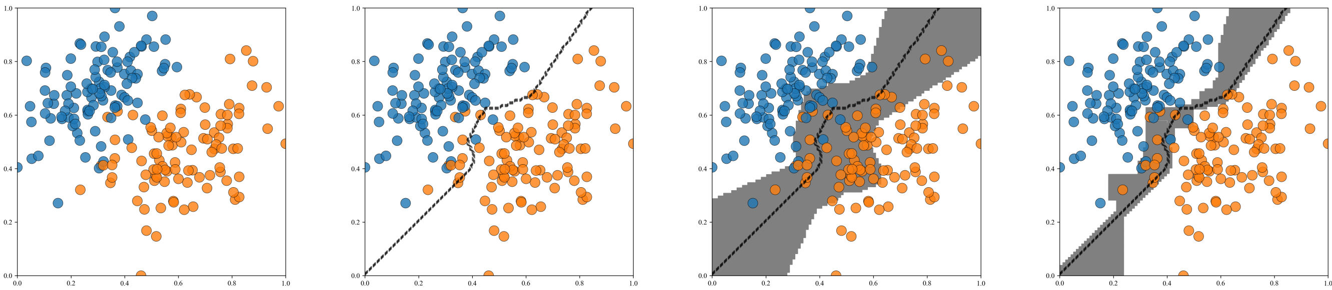
where $i = \{1, ..., M\}$ and $k = \{0, 1, ..., k_{\text{max}}\}$ are determined by grid search to maximize Δf_{acc} / Δf_{reject}. Here, k_{max} and t are hyperparameters to control the range of grid search and the step size for the update.

E. Two-Stage Reject Option

Threshold based reject options explained in III-A and III-B often reject many patterns more than necessary in order to maximize the accuracy. To avoid this unnecessary rejection, we propose a two-stage reject option.

As shown in eq. (15), the same threshold-based reject option (i.e., ST or CWT) is performed in the first stage. If the classification is rejected, an ML model is performed as a second opinion in the second stage. If the output class of the fuzzy classifier is the same as that of the ML model, the rejection is withdrawn and the class label of the winner rule in the fuzzy classifier is output.

$$y(\mathbf{x}_p) = \begin{cases} C & \text{if } \delta \geq T, \\ C & \text{if } \delta < T \text{ and } f_{\text{ML}}(\mathbf{x}_p) = C, \\ \text{reject} & \text{otherwise,} \end{cases} \tag{15}$$



(a) Pattern distribution (b) Decision boundary (c) Reject region by the 1st stage (d) Reject region by the 2nd stage

Fig. 2: Example of the proposed fuzzy classifier with a two-stage reject option on a two-dimensional synthetic dataset

where $f_{\text{ML}}(\mathbf{x}_p)$ is the output class from an ML model. Any ML model can be employed in the second stage.

IV. Computational Experiments

A. Two-dimensional Synthetic Dataset

For the demonstration purpose, we apply the proposed fuzzy classifier with a two-stage reject option to a two-dimensional synthetic dataset which includes two classes in Fig. 2 (a). The dot-line in Fig. 2 (b) represents the decision boundary generated by a fuzzy classifier which is obtained by our Michigan-style FGBML. For this dataset, we use only two partitions (i.e., top-left membership functions in Fig. 1) and three partitions (i.e., top-center membership functions in Fig. 1). Twelve patterns are misclassified in this example.

By using the ST reject option with $T = 0.55$ explained in III-A, the patterns in the grey region are rejected for the classification in Fig. 2 (c). That is, all the patterns except for the rejected ones are correctly classified. This improves the reliability of the classifier. The rejected patterns are usually re-investigated or manually inspected. However, in this case, there are too many rejected patterns.

By applying Random Forest to the rejected patterns in the second stage, the rejections of many orange class patterns around the decision boundary are withdrawn. Then, those patterns are correctly classified as shown in Fig. 2 (d). The unnecessary reject region is well-shrunk while all the patterns are correctly classified.

Table I shows the fuzzy rules in the classifier obtained by our Michigan-style FGBML. Each row represents a single rule. S2 and L2 represent Small and Large of the top-left membership functions in Fig. 1, respectively. S3, M3, L3 represent Small, Medium, and Large of the top-center membership functions in Fig. 1, respectively. DC represents "*don't care*". "Base", "1st stage", and "2nd stage" mean the number of classified patterns in each stage (i.e., Fig. 2 (b), (c), and (d), respectively). We can see that R4 covers the central region around the decision boundary. The patterns classified by R4 are rejected a lot in the first stage and are well recovered in the second stage.

B. Real-World Datasets

To analyze the effects of two threshold settings (i.e., ST and CWT) and the chosen ML models on accuracy-rejection rate tradeoffs, we apply our proposed method to eight real-world datasets shown in Table II available from the Keel dataset repository [14]. Each attribute value is normalized into [0, 1] beforehand in this paper.

TABLE I: The fuzzy rules in the obtained classifier

Index	Antecedent		Class	CF	Number of classified patterns		
	x_1	x_2			Base	1st stage	2nd stage
R1	S3	DC	Blue	0.82	68	51	59
R2	L3	DC	Orange	0.96	42	36	42
R3	DC	L3	Blue	0.66	32	27	32
R4	L2	M3	Orange	0.49	27	4	25
R5	DC	S3	Orange	0.84	8	4	8
R6	M3	S3	Orange	0.92	7	5	7
R7	M3	L3	Blue	0.73	7	7	7
R8	M3	S2	Orange	0.39	6	0	6
R9	L3	M3	Orange	0.98	2	2	2
R10	S2	DC	Blue	0.30	1	0	0

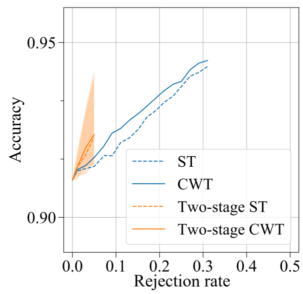
TABLE II: Real-world datasets

Dataset name	Number of patterns	Number of attributes	Number of classes
Australian	690	14	2
Pima	768	8	2
Vehicle	846	18	4
Vowel	990	13	11
Phoneme	5,404	5	2
Texture	5,500	40	11
Satimage	6,435	36	7
Penbased	10,992	16	10

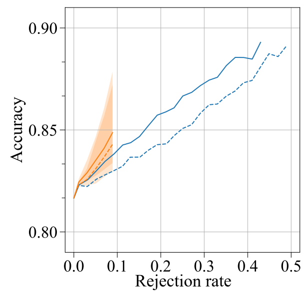
For threshold optimization, we optimize the threshold(s) of ST and CWT reject options using the training data and the hyperparameters in Table 2. We use eight ML models shown in Table IV which are implemented in Scikit-learn. Hyperparameters in Table IV are tuned by grid search using the training data. For the other hyperparameters, the default settings are used.

We perform 10-fold cross-validation three times (i.e., 30 trials in total). In each trial, we perform the proposed method with different $RR_{\max}$ from 0.0 to 1.0 in increments of 0.01 in order to obtain the accuracy-rejection rate curve. For the accuracy and rejection rate obtained over 30 trials, we calculate the averages of accuracy and rejection rate for each 0.02 interval of the rejection rate, only if it is obtained for more trials than half of all trials (i.e., more than 15 trials).

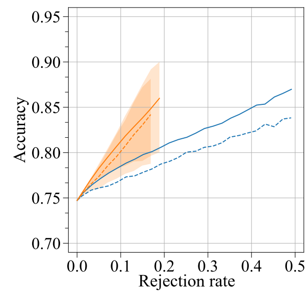
Figs. 3 and 4 show the average accuracy-rejection rate



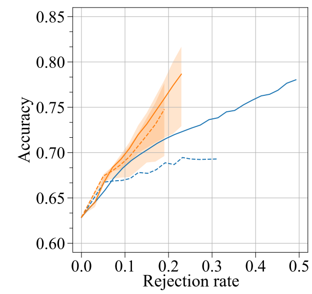
(a) Australian



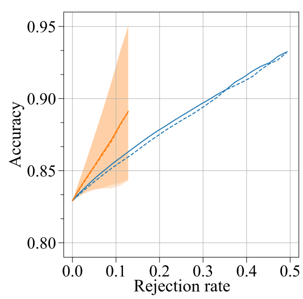
(b) Pima



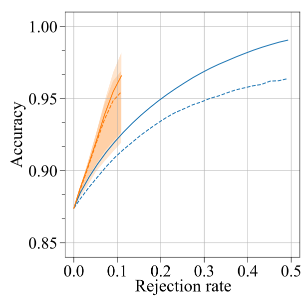
(c) Vehicle



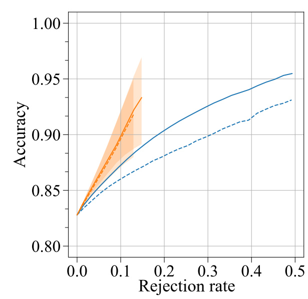
(d) Vowel



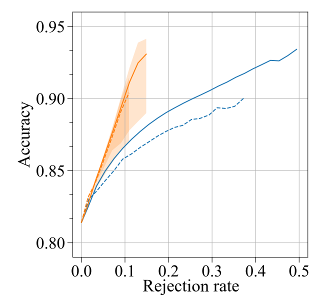
(e) Phoneme



(f) Texture



(g) Satimage



(h) Penbased

Fig. 3: Accuracy-rejection rate tradeoffs on the training data. Blue and orange lines represent the threshold-based reject options and the two-stage reject options, respectively. Dashed and solid lines represent ST and CWT reject options, respectively.

tradeoffs on the training data and test data, respectively. Blue and orange lines represent single-stage and two-stage reject options while dashed and solid lines represent ST and CWT reject options, respectively. The top and bottom edges of the orange zone mean the best and worst ML models, respectively.

For single-stage reject options (i.e., blue lines in each graph), we can say that CWT is better than ST for most of the datasets on both training and test data. However, for two-stage reject options (i.e., orange lines in each graph), a clear difference is not observed between CWT and ST. By comparing between single-stage and two-stage reject options, two-stage reject options have better accuracy-rejection rate tradeoffs than single-stage reject options on average for most datasets. However, the orange zone under the accuracy-reject rate tradeoff curve by the single-stage reject options indicates that some ML models do not improve the accuracy-rejection rate tradeoff.

Fig. 5 shows the detailed accuracy-rejection tradeoffs on the test data. Due to the page limitation, the results on four out of eight datasets by CWT reject options are shown. Although the best ML model depends on the datasets, decision tree and random forest seem to be better for many datasets. Decision tree can also be explainable for second opinions.

V. Conclusion

In this paper, we proposed a fuzzy classifier with a two-stage reject option where an ML model is used after the threshold-based decision in order to avoid unnecessary rejections. Computational experiments showed better accuracy-rejection rate tradeoffs by two-stage reject options with different machine learning models than the conventional threshold-based reject options. A further tradeoff analysis together with an explainability issue is our future work.

TABLE III: Parameter specification for threshold optimization

Parameter	Values
t	0.001
k_{max}	500
RR_{max}	{0, 0.01, ..., 0.5}

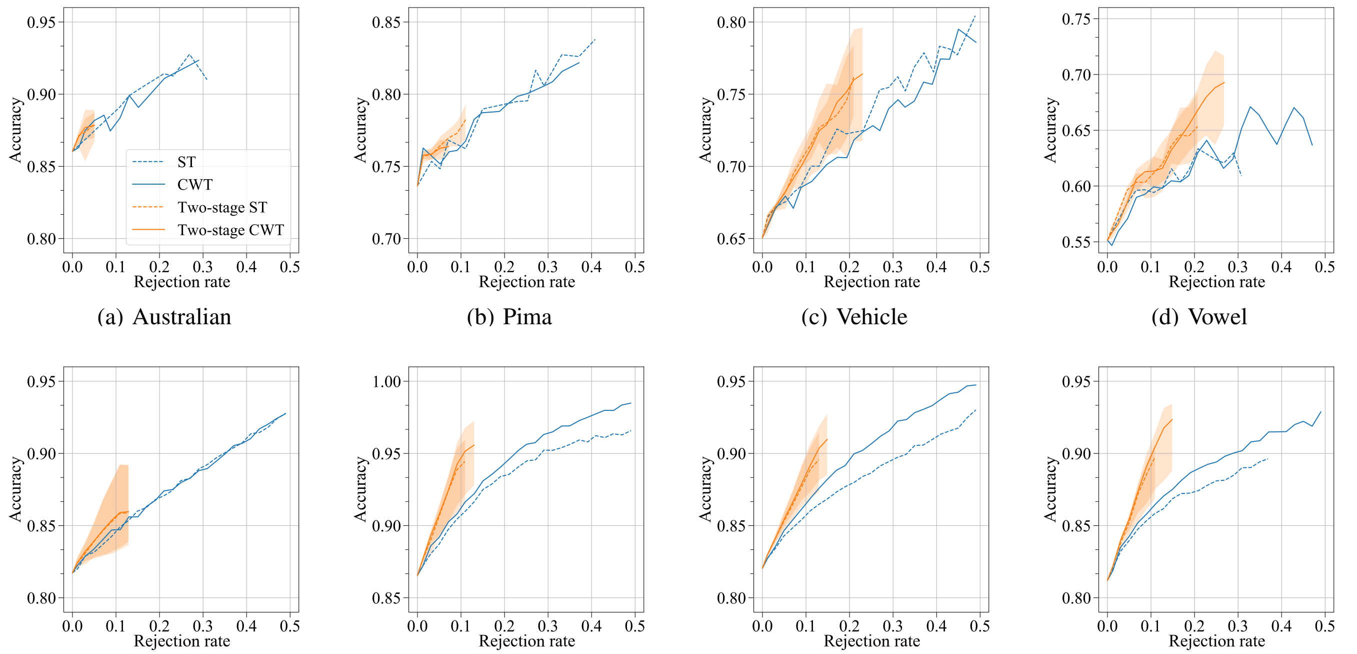
TABLE IV: Machine learning models and their hyperparameters tuned by grid search

ML model	Hyperparameter
AdaBoost	Adaboost-SAMME
Decision Tree	Max depths : {5, 10, 20}
Naïve Bayes	-
Gaussian Process	-
K-Nearest Neighbors	k: {2, 3, 4, 5, 6}
Multi-Layer Perceptron	ReLU, α: $\{10^{-5}, 10^{-4}, ..., 10^{2}\}$
Random Forest	Max depths : {5, 10, 20}
Linear Support Vector Machines	C: $\{2^{-5}, 2^{-4}, ..., 2^{15}\}$

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.

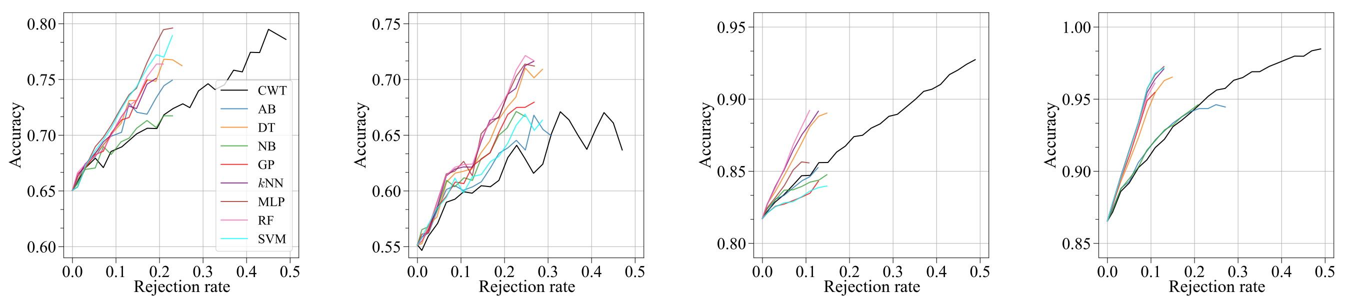
[2] H. Ishibuchi, T. Nakashima, and M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Berlin, Heidelberg: Springer-Verlag, 2004.



(a) Australian (b) Pima (c) Vehicle (d) Vowel

(e) Phoneme (f) Texture (g) Satimage (h) Penbased

Fig. 4: Accuracy-rejection rate tradeoffs on the test data. Blue and orange lines represent the threshold-based reject options and the two-stage reject options, respectively. Dashed and solid lines represent ST and CWT reject options, respectively.



(a) Vehicle (b) Vowel (c) Phoneme (d) Texture

Fig. 5: Detailed accuracy-reject tradeoffs on the test data. Two-stage CWT with each of eight ML models.

[3] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in *Int. Conf. on IPMU*, 2018.

[4] A. Fernández, S. García, J. Luengo, E. Bernadó-Mansilla, and F. Herrera, "Genetics-based machine learning for rule induction: State of the art, taxonomy, and comparative study," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 6, pp. 913–941, 2010.

[5] K. Hendrickx, L. Perini, D. V. der Plas, W. Meert, and J. Davis, "Machine learning with a reject option: A survey," *CoRR*, vol. abs/2107.11277, 2021.

[6] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[7] G. Fumera, F. Roli, and G. Giacinto, "Reject option with multiple thresholds," *Pattern Recognition*, vol. 33, pp. 2099–2101, 2000.

[8] H. Ishibuchi and T. Nakshima, "Fuzzy classification with reject options by fuzzy if-then rules," in *1998 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, vol. 2, pp. 1452–1457, 1998.

[9] C. Frelicot and L. Mascarilla, "A third way to design pattern classifiers with reject options," in *2002 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS)*, pp. 395–399, 2002.

[10] J. C. Hühn and E. Hüllermeier, "FR3: A fuzzy rule learner for inducing reliable classifiers," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 138–149, 2009.

[11] Y. Nojima and H. Ishibuchi, "Multiobjective fuzzy genetics-based machine learning with a reject option," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1405–1412, 2016.

[12] Y. Nojima, K. Watanabe, and H. Ishibuchi, "Simple modifications on heuristic rule generation and rule evaluation in michigan-style fuzzy genetics-based machine learning," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, 2015.

[13] G. Fumera, F. Roli, and G. Giacinto, "Multiple reject thresholds for improving classification reliability," in *Advances in Pattern Recognition* (F. J. Ferri, J. M. Iñesta, A. Amin, and P. Pudil, eds.), pp. 863–871, Springer Berlin Heidelberg, 2000.

[14] J. Derrac, S. Garcia, L. Sanchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multi-Valued Logic & Soft Computing*, vol. 17, pp. 255–287, 2015.