Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Annealing Based Approach to Optimize Classification Systems

Paulo V. W. Radtke, Robert Sabourin, *Member, IEEE*, and Tony Wong, *Member, IEEE*

Abstract— **Classification systems optimization is often performed with population based genetic algorithms. These methods are known for their efficacy on solving these problems, but are associated to a high computational burden with classification systems. This paper evaluates an annealing based approach to optimize a classification system, and compares results obtained with a multi-objective genetic algorithm in the same problem. Experiments conducted with isolated handwritten digits demonstrate the effectiveness of the annealing based approach, which encourages further research in this direction.**

I. Introduction

Image-based pattern recognition usually requires that pixel information be first transformed into an abstract representation (a feature vector) suitable for recognition with classifiers, a process known as *feature extraction*. A relevant classification problem is the intelligent character recognition, most specifically the offline recognition of isolated handwritten symbols on documents. A methodology to extract features must select the spatial location to apply transformations on the image [1]. The choice takes into account the *domain context*, the type of symbols to classify, and the *domain knowledge*, what was previously done in similar problems. The process is usually performed by a human expert in a trial-and-error process. We also have that changes in the domain context may manifest in the same classification problem, which also requires changes in the classification system.

To minimize the human intervention in defining and adapting classification systems, this problem is modeled as an optimization problem, using the domain knowledge and the domain context. This paper discusses the two-level approach to optimize classification systems in Fig. 1. The first level employs the *Intelligent Feature Extraction* (IFE) methodology to extract feature sets that are used on the second level to optimize an *ensemble of classifiers* (EoC) to improve accuracy.

One trend for these classification problems is to use genetic based approaches [2]–[5], specially *multi-objective genetic algorithms* (MOGAs). These approaches have been found effective to solve these problems, but with a high processing time. Population based approaches evaluate a large number of candidate solutions, hence the use of other algorithms may provide comparable solutions associated to a lower computational burden. The algorithm chosen for this comparative study is the *Record-to-Record Travel* (RRT) algorithm [6], an annealing based heuristic. This local search algorithm features an strategy to avoid local optimum solutions, a feature often required to optimize classification problems.

Paulo V. W. Radkte is with the Pontifícia Universidade Católica do Paraná, R. Imaculada Conceição 1155, CEP 80215-901, Curitiba, PR, Brazil (e-mail: paulo.radtke@pucpr.br).

Robert Sabourin and Tony Wong are with the Automated Manufacturing Engineering Department, École de technologie supérieure, University of Québec, 1100 Notre-Dame West, Montral (Québec) Canada, H3C 1K3 (e-mail: robert.sabourin@etsmtl.ca, tony.wong@etsmtl.ca).

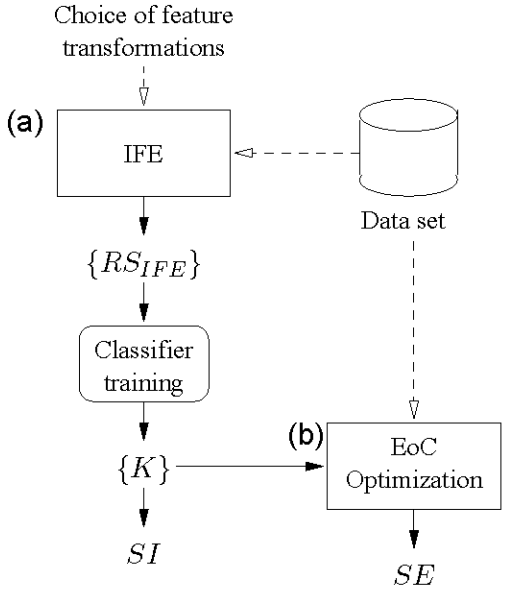


Fig. 1. Classification system optimization approach. Representations obtained with IFE are used to further improve accuracy with EoCs.

This paper extends the work in [7]. The new contribution is to investigate an annealing based approach to optimize classification systems. The paper has the following structure. The approach to optimize classification systems is discussed in Section II, and Section III discusses the RRT algorithm. Section IV details the experimental protocol and the results obtained. Finally, Section V discusses the goals attained.

II. Classification System Optimization

Classification systems are modeled in a two-level process (Fig. 1). The first level uses the IFE methodology to obtain the representation set RS_{IFE} (Fig. 1.a). The representations in RS_{IFE} are then used to train the classifier set K that is considered for aggregation on an EoC SE for improved accuracy (Fig. 1.b). Otherwise, if a single classifier is desired for limited hardware, such as embedded devices, the most accurate single classifier SI may be selected from K. The next two subsections details both the IFE and EoC optimization methodologies.

A. Intelligent Feature Extraction

The goal of IFE is to help the human expert define representations in the context of isolated handwritten symbols,

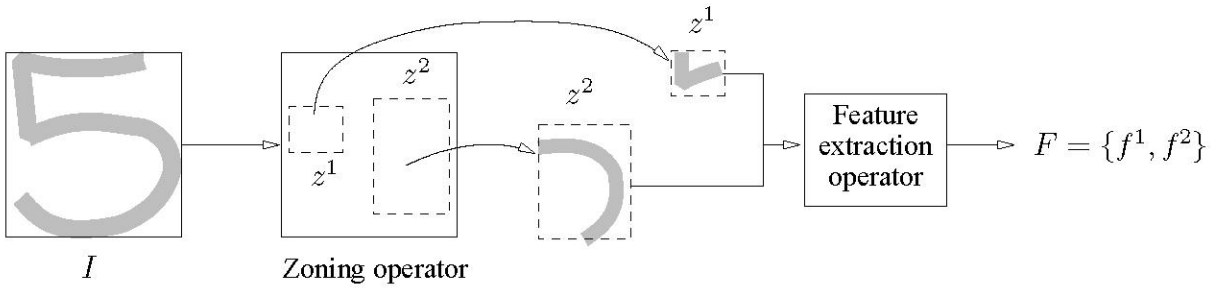


Fig. 2. IFE structure.

using a wrapper approach to optimize solutions. IFE models handwritten symbols as features extracted from specific *foci* of attention on images using *zoning*. Two operators are used to generate representations with IFE: a *zoning operator* to define foci of attention over images, and a *feature extraction* operator to apply transformations in zones. The choice of transformations for the feature extraction operator constitutes the domain knowledge. The domain context is introduced as actual observations in the *optimization* data set used to evaluate and compare solutions. Hence, the zoning operator is optimized by the IFE to the domain context and domain knowledge.

The IFE structure is illustrated in Fig. 2. The zoning operator defines the zoning strategy $Z = \{z^1, \ldots, z^n\}$, where $z^i, 1 \leq i \leq n$ is a zone in the image I and n the total number of zones. Pixels inside the zones in Z are transformed by the feature extraction operator in the representation $F = \{f^1, \ldots, f^n\}$, where $f^i, 1 \leq i \leq n$ is the partial feature vector extracted from z^i. At the end of the optimization process, the optimization algorithm has explored the representation set $RS_{IFE} = \{F^1, \ldots, F^p\}$ (for MOGAs, RS_{IFE} is the optimal set at the last generation).

The result set RS_{IFE} is used to train a discriminating classifier set $K = \{K^1, \ldots, K^p\}$, where K^i is the classifier trained with representation F^i. The first hypothesis is to select the most accurate classifier $SI, SI \in K$ for a single classifier system. The second hypothesis is to use K to optimize an EoC for higher accuracy, an approach discussed in Section II-B. The remainder of this section discusses the IFE operators chosen for experimentation with isolated handwritten digits and the candidate solution evaluation.

1) Zoning Operator: To compare performance to the traditional human aproach, a *baseline* representation with a high degree of accuracy on handwritten digits with a *multi-layer Perceptron* (MLP) classifier [8] is considered. Its zoning strategy, detailed in Fig. 3.b, is defined as a set of three image dividers, producing 6 zones. The *divider zoning operator* expands the baseline zoning concept into a set of 5 horizontal and 5 vertical dividers that can be either *active* or *inactive*, producing zoning strategies with 1 to 36 zones. Fig. 3.a details the operator template, encoded by a 10-bit binary string. Each bit is associated with a divider's state (1 for active, 0 for inactive).

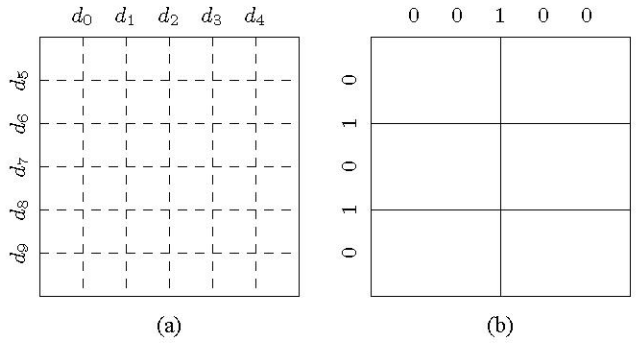


Fig. 3. Divider zoning operator (a). The baseline representation in (b) is obtained by setting only d_2, d_6 and d_8 as active.

2) Feature Extraction Operator: Oliveira *et al.* used and detailed in [8] a mixture of concavities, contour directions and black pixel surface transformations, extracting 22 features per zone (13 for concavities, 8 for contour directions and 1 for surface). To allow a direct comparison between IFE and the baseline representation, the same feature transformations (the domain knowledge) are used to assess the IFE.

3) Candidate Solution Evaluation: Candidate solutions are evaluated with respect to their classification accuracy. Thus, the objective is to minimize the classification error rate on the *optimization* data set (the domain context). To compare optimization methods, candidate solutions are evaluated with the *projection distance* (PD) classifier [9].

B. EoC Optimization

A recent trend in PR has been to combine several classifiers to improve their overall performance. Algorithms for creating EoCs will usually fall into one of two main categories. They either manipulate the training samples for each classifier in the ensemble (like Bagging and Boosting), or they manipulate the feature set used to train classifiers [2]. The key issue is to generate a set of diverse and fairly accurate classifiers for aggregation [10].

We create EoCs on a two-level process. The first level creates a classifier set K with IFE, and the second level optimizes the classifiers aggregated. We assume that RS_{IFE} generates a set K of p diverse and fairly accurate classifiers. To realize this task, the classifiers in K are associated with a

binary string E of p bits, which is optimized to select the best combination of classifiers using an optimization algorithm. The classifier K^i is associated with the i^{th} binary value in E, which indicates whether or not the classifier is active in the EoC.

The optimization process minimizes the EoC classification error on the *optimization* data set. This is supported by [11]. Evaluating the EoC error rate requires actual classifier aggregation. PD classifiers are aggregated by majority voting, and votes are calculated once and stored in memory to speed up the opimization process.

III. Optimization Algorithm

A local search algorithm is used to optimize the IFE and EoC. The algorithm chosen is the *Record-to-Record Travel* (RRT) algorithm [6], an annealing based heuristic. The RRT algorithm improves an initial solution i by searching in its neighborhood for better solutions based on their evaluation (classification error rate). The RRT algorithmn, detailed in Algorithm 1, produces after a number of iterations the record solution r. The algorithm is similar to a hill climbing aproach, but avoids local optimum solutions by allowing the search towards non-optimal solutions with a fixed deviation D. Earlier experiments indicated that the RRT algorithm over-fitted solutions during the optimization process. The global validation strategy discussed in [12] is used to avoid this effect, and Algorithm 1 includes support for this strategy.

Data: Initial solution i
Data: Deviation D
Result: Record solution r
Result: Explored solution set S
$r = i$;
$RECORD = eval(r)$;
$p = i$;
$S = \emptyset$;
repeat
  Create the solution set P, neighbor to p;
  $S = S \cup P$
  Select the best solution $p' \in P$ such as that p' has not yet been evaluated;
  if $eval(p') < RECORD + RECORD \times D$ **then**
    $p = p'$;
    if $eval(p') < RECORD$ **then**
      $RECORD = eval(p')$;
      $r = p'$;
    end
  end
until $eval(p) <= RECORD + RECORD \times D$;

Algorithm 1: Modified record to record travel (RRT) algorithm used to optimize classification systems with global validation.

Given the inicial solution i, the algorithm will copy it to the record solution r and store its evaluation value in $RECORD$. It also copies i as the current solution p. Next it will repeat the following process during a number of iterations, until the current solution is worse than the record solution plus the allowed deviation. First it will find the set P, solutions neighbor to p, and select the best neighbor $p', p' \in P$. To avoid cyclic optimization, solutions already evaluated are not considered for p'. If evaluating p' yields results within the allowed deviation, it is copied as p for the next iteration. Solution p' replaces the record solution r only if it yields better results. If p' is worse than the allowed deviation, the optimization process stops. The explored solution set S is responsible to store solutions tested by the RRT algorithm for the global validation strategy. At each iteration, the algorithm inserts into S the solutions in the neighbor set P. At the end of the optimization process, solutions in S are validated and the most accurate solution is selected. For the IFE process, S is the result set RS_{IFE} used to create the classifier set K.

Neighbors to solution X^i are created by swapping bits in the binary string with their complement. For a binary string E with p bits, a set of p neighbors is created by complementing each bit $i, 1 \leq i \leq p$ on solution E^i. For the IFE, solution in Fig. 4.a has solutions in Figs. 4.b and 4.c as two possible neighbors.

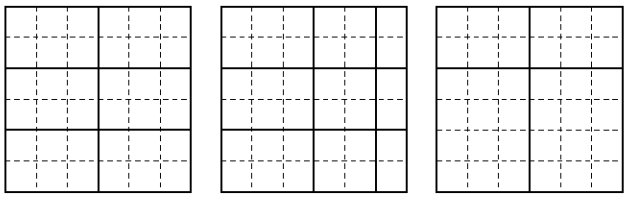


Fig. 4. Divider zoning operator – zoning strategy (*a*) and two neighbors (*b* and *c*).

IV. Experimental Protocol and Results

The tests are performed as in Fig. 1. The IFE methodology is solved to obtain the representation set RS_{IFE}, which is used to train the classifier sets K . For a single classifier system, the most accurate classifier $SI, SI \in K$ is selected. EoCs are then created with K, producing SE. To select resulting solutions, we use the global validation approach detailed in [12]. Solutions obtained are compared to the baseline representation defined in [8] and to solutions obtained with MOGAs in [7]. Unlike MOGAs which may produce different solutions on each run, the RRT algorithm will yield the same result set S for the same initial solution i. Thus, solutions obtained with the RRT are compared to both average results in [7] and to the best result obtained in 30 runs.

The data sets in Table I are used in the experiments – isolated handwritten digits from NIST-SD19. Classifier training is performed with the *training* data set. The *validation* data set is used to adjust the classifier parameters (PD hyper planes). The optimization process is performed with the

optimization data set, and the *selection* data set is used with the global validation strategy to select solutions. Solutions are compared with the test data sets, $test_a$ and $test_b$. Fumera's method [13] is used to implement a rejection strategy. A *softmax* function is used to convert classes distance to posterior probabilities, and PD votes are converted to posterior probabilities with Henses's method [14].

TABLE I
HANDWRITTEN DIGITS DATA SETS EXTRACTED FROM NIST-SD19.

Data set	Size	Origin	Sample range
training	50000	hsf_0123	1 to 50000
validation	15000	hsf_0123	150001 to 165000
optimization	15000	hsf_0123	165001 to 180000
selection	15000	hsf_0123	180001 to 195000
$test_a$	60089	hsf_7	1 to 60089
$test_b$	58646	hsf_4	1 to 58646

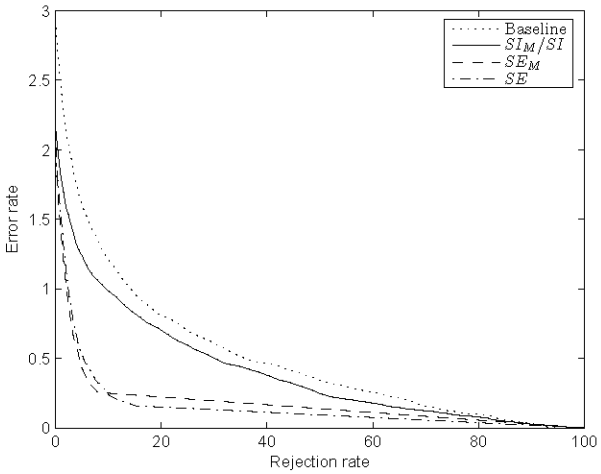
Both the IFE and EoC have initial solutions associated to empty strings. Thus, there are no dividers active in the initial IFE solution, and no classifiers associated to the initial EoC. The deviation D is set empirically to $D = 5\%$. The RRT is a deterministic algorithm, hence a single run is performed with both processes. All experiments were performed on a Athlon64 3000+ processor with 1GB of RAM memory.

Results obtained are detailed in Table II, regarding average MOGA results in 30 replications in [7], and in Table III, which regards the best MOGA result in these 30 replications. In both tables Z is the solution zone number and $|S|$ is the solution cardinality (either feature number or classifier number). In Table II, e_{test_a} and e_{test_b} are classification error rates on $test_a$ and $test_b$, whereas $test_a$ and $test_b$ in Table III are classifier accuracies, measured with both zero rejection (e_{max}) and with rejection using fixed error rates. Solutions SI_M and SE_M where obtained with MOGAs, and the baseline representation is defined in [8]. These solutions are included for comparison purposes. Figure 5 details the error-rejection curve for solutions detailed in Table III.

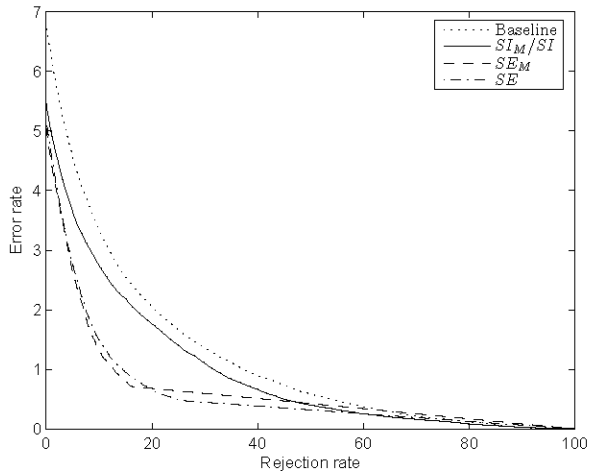
TABLE II
IFE AND EOC OPTIMIZATION RESULTS – MEAN VALUES ON 30 MOGA REPLICATIONS (STANDARD DEVIATION VALUES IN PARENTHESIS) AND ACTUAL ERROR RATES FOR REMAINING SOLUTIONS.

Solution	Z	$\|S\|$	e_{test_a}	e_{test_b}
Baseline	6	132	2.96%	6.83%
SI_M	15	330	2.18% (0)	5.47% (0)
SI	15	330	2.18%	5.47%
SE_M	–	24.67	2.00% (0.040)	5.19% (0.087)
SE	–	23	2.05%	5.20%

Solutions SI and SE obtained with the RRT algorithm outperform the baseline representation defined by the human expert. Figure 6 details the zoning strategy associated to SI and SI_M. Comparing these solutions to SI_M and SE_M, we conclude that the RRT performed similarly to an MOGA. Solution SI obtained by the RRT has the same zoning strategy as SI_M, and the error rate for SE is comparable to the average SE_M. The accuracy difference in this context is associated to the RS_{IFE} set obtained with the RRT, which is not the same set obtained with the MOGA. These results indicate that the RRT algorithm is effective to optimize classification systems.



(a) $test_a$



(b) $test_b$

Fig. 5. PD classifier rejection rate curves of solutions in Table III.

V. DISCUSSION

Solutions obtained with the RRT are comparable to solutions obtained with MOGAs. The advantage of the RRT algorithm is the lower computational burden to optimize the

TABLE III

IFE AND EOC BEST MOGA RESULTS IN 30 REPLICATIONS COMPARED TO RRT RESULTS. ACCURACIES MEASURED WITH AND WITHOUT REJECTION.

Solution	**Z**	$\|S\|$	$test_a$				$test_b$			
			e_{max}	1.5%	1%	0.5%	e_{max}	3%	2%	1%
Baseline	6	132	97.04%	92.67%	85.15%	63.78%	93.17%	85.31%	77.69%	61.64%
SI_M	15	330	97.82%	96.05%	90.15%	68.96%	94.53%	88.57%	81.64%	67.71%
SI	15	330	97.82%	96.05%	90.15%	68.96%	94.53%	88.57%	81.64%	67.71%
SE_M	–	23	98.04%	97.62%	94.86%	94.86%	94.94%	92.58%	90.87%	86.68%
SE	–	23	97.95%	97.48%	96.46%	94.15%	94.80%	92.62%	90.32%	84.33%

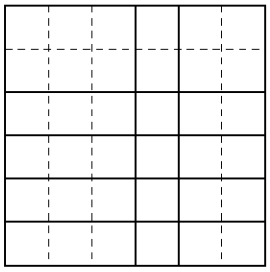


Fig. 6. Zoning strategy associated to SI and SI_M.

classification system discussed. To optimize the IFE a total of 76 candidate solutions were evaluated by the RRT, whereas the MOGA evaluates 64 candidate solutions only on its initial population in [7] to optimize the same problem. The same is observed with EoCs, the RRT evaluated a total of 14000 solutions to optimize the EoC, and the MOGA evaluated 166000 solutions thoughout the same optimization process.

Solutions obtained with the RRT algorithm were also over-fitted to the *optimization* data set. The global validation strategy detailed in [12] selected better results in S than simply selecting the record solution r obtained at the end of the optimization process. This reinforces the conclusion that the optimization of classification systems using wrapped classifiers is prone to solution over-fit.

This research indicates two directions for further development. The first is to optimize the IFE and EoC using MLP classifiers. The aproach in [7] used the PD as a meta model to optimize representations for MLP classifiers, thus a different zoning strategies set may be selected when using an actual MLP classifier during the optimization process. The second is to compare algorithmic performance with other optimization problems, such as classification system optimization for handwritten letters or feature subset selection.

ACKNOWLEDGMENTS

The first author would like to acknowledge the CAPES and the Brazilian government for supporting part of this research through scholarship grant BEX 2234/03-3. The first author would also like to acknowledge the Pontificia Universidade Católica do Paraná (PUCPR, Brazil) for supporting this research. The other authors would like to acknowledge the NSERC (Canada) for supporting this research.

REFERENCES

[1] Z.-C. Li and C. Y. Suen, "The partition-combination method for recognition of handwritten characters," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 701–720, 2000.

[2] L. I. Kuncheva and L. C. Jain, "Design classifier fusion systems by genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 327–336, 2000.

[3] G. Tremblay, R. Sabourin, and P. Maupin, "Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm," in *17th International Conference on Pattern Recognition – ICPR2004*. Cambridge, U.K.: IEEE Computer Society, August 2004, pp. 208–211.

[4] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Sequential genetic search for ensemble feature selection," in *Proceddings of International Joint Conference on Artificial Intelligence*, 2005, pp. 877–882.

[5] J. Handl and J. Knowles, "Feature subset selection in unsupervised learning via multiobjective optimization," *International Journal on Computational Intelligence Research*, vol. 3, no. 1, pp. 217–238, 2006.

[6] J. W. Pepper, B. L. Golden, and E. A. Wasil, "Solving the traveling salesman problem with annealing-based heuristics: A computational study," *IEEE Trans. on Systems, Mand and Cybernetics – Part A: Systems and Humans*, vol. 32, no. 1, pp. 72–77, 2002.

[7] P. V. W. Radtke, T. Wong, and R. Sabourin, "Classification system optimization with multi-objective genetic algorithms," in *Proceedings of the 10^{th} International Workshop on Frontiers in Handwriten Recognition (IWFHR 2006)*. IAPR, 2006, pp. 331–336.

[8] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1438–1454, 2002.

[9] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Handwritten Numeral Recognition using Autoassociative Neural Networks," in *Proceedings of the International Conference on Pattern Recognition*, 1998, pp. 152–155.

[10] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[11] D. Ruta and B. Gabrys, "Classifier Selection for Majority Voting," *Information fusion*, vol. 6, pp. 63–81, 2005.

[12] P. V. W. Radtke, T. Wong, and R. Sabourin, "An evaluation of over-fit control strategies for multi-objective evolutionary optimization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006)*. IEEE Computer Society, 2006, pp. 6359–6366.

[13] G. Fumera, F. Roli, and G. Giacinto, "Reject option with multiple thresholds," *Pattern Recoginition*, vol. 33, no. 12, pp. 2099–2101, 2000.

[14] L. K. Hensen, C. Liisberg, and P. Salamon, "The error-reject tradeoff," *Open Systems & Information Dynamics*, vol. 4, no. 2, pp. 159–184, 1997.