

# A multi-objective competitive co-evolutionary approach for classification problems

Van Truong VU  
*Le Quy Don Technical University*  
 Ha Noi, Viet Nam  
[truongvv@mta.edu.vn](mailto:truongvv@mta.edu.vn)

Lam Thu BUI  
*Le Quy Don Technical University*  
 Ha Noi, Viet Nam  
[lam.bui07@gmail.com](mailto:lam.bui07@gmail.com)

Trung Thanh NGUYEN  
*Liverpool John Moores University*  
 Liverpool, United Kingdom  
[T.T.Nguyen@ljmu.ac.uk](mailto:T.T.Nguyen@ljmu.ac.uk)

**Abstract**—This paper proposes a multi-objective competitive co-evolutionary algorithm (MOCPCEA) based on the Prey-Predator model to solve classification problems. In the MOCPCEA, a data population acts as preys. To be specific, each prey represents a selected subset of the training dataset. Another population is ANN classifiers which play as Predators. The task of the Predators is to try to classify the data sets as correctly as possible, whereas the Preys try to find the data sets that are difficult to be classified. Through this interaction process, MOCPCEA generates a set of classifiers that are able to classify difficult data sets. The final classification result is given by the ensemble voting mechanism among these sets of classifiers. The performance of the proposed algorithm is performed on seven benchmark problems. Through comparison with other algorithms, the proposed algorithm indicates that it could create an ensemble of ANN networks that give high and stable classification results.

**Keywords**—competitive co-evolutionary, Prey-Predator, multi-objective optimization, classification, ensemble learning.

## I. INTRODUCTION

In the field of evolutionary computation (EC), co-evolution is a variant of the genetic algorithm (GA). The term *coevolution* is originally derived from the field of biology [1]. In 1990, Holland [2] illustrated in detail the process of competing between insects and plants and this was an important study for the introduction of the Prey-Predator competitive co-evolutionary model. In coevolution, the fitness value of each individual is calculated based on the interaction with individuals in other populations. Based on the type of interaction, coevolution can be divided into two main groups: co-operative and competitive coevolution. The main idea of co-operative coevolution is based on a "divide-and-conquer" technique. A problem is broken down into sub-problems, then using a sub-population to solve a sub-problem. An interesting point here is that in order to compute the fitness value of an individual in a sub-population, the individual needs to be combined with individuals from other populations to form a complete solution. The fitness value of an individual is measured based on its ability to coordinate with other ones in solving a problem. Meanwhile, in competitive coevolution, the evolution is a process of fighting between populations. The fitness of an individual will be relative fitness. It means the fitness is calculated based on the competition with individuals in other populations. Through this arms race process, all parties will be jointly developed.

Up to now, co-evolution has been used to solve many different problems. While most competitive co-evolution (CPCE) studies are often applied to the field of games and robots, etc.[3], co-operative co-evolution (COCE) studies are used in a more diverse way. In [4], the authors used a COCE

approach for the large scale global optimization. The COCE algorithms are also used in object recognition [5] and facial recognition problems such as locating the eyes in a face image [6] or securing high resolution grayscale facial images [7]. In the classification problem, most of the research focuses on using COCE approaches in optimizing the classifiers ([8], [9]). In recent years, the researches has started to be more interested in utilizing the competitive approaches in solving machine learning problems [10-14].

In [10] the authors proposed a competitive model by using a population of MLPANNs networks and a population of RBFANNs networks. To assess the fitness of individuals, the authors mixed two populations to form an arena. Each individual randomly selects some other individuals to create pairwise competitions. The proposed algorithm has been compared with ten other algorithms on many different benchmark problems. The results showed that the proposed algorithm has the ability to properly classify with high rates. In this study, the authors will use this algorithm as a basis for comparing.

In 2014, Ian Goodfellow and colleagues proposed a generative adversarial network (GAN) [11]. This was considered a revolution in the field of deep learning and was "the coolest idea in machine learning in the last twenty years" (Yann LeCun). GAN utilized two neural networks simultaneously a generative network (GN) and a discriminative network (DN). The basic idea of GAN was based on the Prey-Predator model. In particular, the GN played as the preys and vice versa. The mission of the GN was to try to create fake data sets as similar to the real data sets as possible, while the DN tried to distinguish those fake data sets from the real data sets. At the end of the co-evolutionary process, we could use the GN to generate fake data sets that were almost identical to the real data sets.

In [13], the authors proposed a competitive co-evolutionary algorithm based on the Prey-Predator model. In particular, Preys represented sub datasets, while Predators were classifiers. The fitness value of the Predator was measured based on the ability to capture the prey (or correctly classify), and the fitness value of the Prey was measured based on the ability to escape the Predator (or misclassified). This study has played an important premise research role for later studies. In [14] the author proposed two competitive co-evolutionary models named CANNT and CENNT. In particular, the CANNT model worked on the same mechanism as the proposed algorithm in the study [10] while in the CENNT, the data population has evolved through crossover and mutation steps like conventional EA algorithms. Experimental results on seven test data sets showed

impressive results. Inspired from these works, in this paper we propose a multi-objective competitive algorithm.

In comparison with the algorithms in [10,14], the proposed algorithm has some following differences:

- ✓ Using the multi-objective optimization algorithm instead of the single objective algorithms to evolve the classifier population. This helps the algorithm to avoid early convergence.

- ✓ The schemes of selecting candidates to participate in the competing process (or sampling schemes): In previous studies, the authors often follow two models: one-to-one (if the population sizes are the same) or one-to-many (if the population sizes are different). However, all of them were picked randomly. This means that the fitness of each individual is unfairly evaluated, it depends heavily on the random sampling process. To overcome this phenomenon we use a common environment (containing a set of the preys/predator) for all predators/preys participating in the fitness assessment process.

- ✓ Although algorithms use the same fitness update mechanism on the data population, the calculation method is completely different. To be specific, instead of iterating over all the datasets, we take an intermediate step to mix all of these data individuals and evaluate the fitness value only once. This approach not only ensures a correct evaluation of classification accuracy but also helps to reduce computation time.

- ✓ To make the final decision, instead of selecting the best individual, the proposed method selects a community of individuals that are considered equally good. The final decision is taken by majority voting. This community-based decision-making solution has been shown to produce more accurate and stable results.

The organization of this paper is as follows: Section II presents the proposed method. Section III shows the experimental results. Finally, conclusions are presented in Section IV.

## II. PROPOSED METHOD

The proposed algorithm diagram is shown in Fig 1. There are two populations. The first one (named population 1 or data population) represents selected data sets and the other (named population 2 or classifier population) represents classifiers (or ANNs). The general idea of the algorithm is as follows: data population tries to evolve to pick hard-to-categorize data samples and take them as input to the classifier population. In turn, when encountering the difficult training dataset, individuals in population 2 will try to evolve to properly classify them. Classification accuracy is one out of two criteria to evaluate which classifier is better. After the reproduction step, these individuals are ranked and sorted into different Fronts, where the first Front (F0) contains the most optimal individuals. The set of individuals on this F0 will then be selected as a general environment for evaluating and updating the fitness of data individuals. The more data samples are misclassified by F0, the more the fitness value increases so that in the next evolutionary generation the probability of being selected increases. The classifier population in turn also have to evolve in response to these difficult samples. In this way, both populations will be improved together.

Specifically, the proposed algorithm consists of two main phases:

Phase 1: The competitive coevolutionary process

Phase 2: The ensemble decision-making process.

Details of each phase will be presented below:

### A. Individual encoding

The two populations use two different encoding strategies. In population 1, we use binary string encoding. Assuming the size of a training data set is  $L$  {Sample 1, Sample 2, ..., Sample  $L$ }. We encode each data individual to be a bit string of length  $L$ . The "1" value corresponding to this sample data will be selected and vice versa. The value "0" or "1" will be determined based on the fitness function value of each sample data. In population 2, each one represents an ANN. Specifically, an individual is encoded as an array of real numbers, this is the weight set (including bias) of the ANN network.

### B. Competitive coevolution (Prey and Predator model)

The input of this step is the training data set and the two populations. The output will be a set of classifiers on F0. This process consists of two main steps: population initialization and coevolution loop.

#### Population initialization

In population 1, suppose  $Fitness[]$  is an array containing the adaptive values of each sample (or data row). The size of the array is the number of samples (i.e.  $L$ ).  $Fitness[i]$  is the adaptive value of the  $i^{th}$  sample. The more sample is misclassified by classifiers, the higher this value will be. Initially, the fitness values of all samples are initialized to 0.5 (i.e. the selected rate of each sample is 50%). The value of each gene in this individual is determined as (1).

$$Individual[i] = \begin{cases} 1, & \text{if } Rand() < Fitness[i] \\ 0, & \text{other wise} \end{cases} \quad (1)$$

Where  $Rand()$  is a function that randomly generates a real number in the range  $[0,1]$ . According to this formula, it is clear that the gene with high fitness value (or more misclassified), the higher the probability that the gene will receive value 1 (or be selected).

In population 2, each individual is an array of real numbers that are randomly generated in a range  $[-\delta, +\delta]$ .

#### Objective functions

After the initialization is completed, the objective function values of each individual are calculated. In this study, two objective functions are classification accuracy (ACC) and diversity (DIV). The ACC of the  $i^{th}$  individual is calculated as (2):

$$ACC_i = \frac{C}{L} \quad (2)$$

Where  $L$  is the total number of training data samples;  $C$  is the number of samples correctly classified by the  $i^{th}$  individual.

To evaluate the performance of a classifier it is necessary to determine which data individuals are used to pair up with. In this study, all data individuals in population 1 are used to

pair with one classifier individual. It can be easily seen with such a pairing strategy, it will actually take a lot of computation time, especially for large datasets. In this study, our approach is different from that in the study [14]. All

individuals in population 1 will be combined into a single one and this individual will act as a prey to pair with the classifiers.

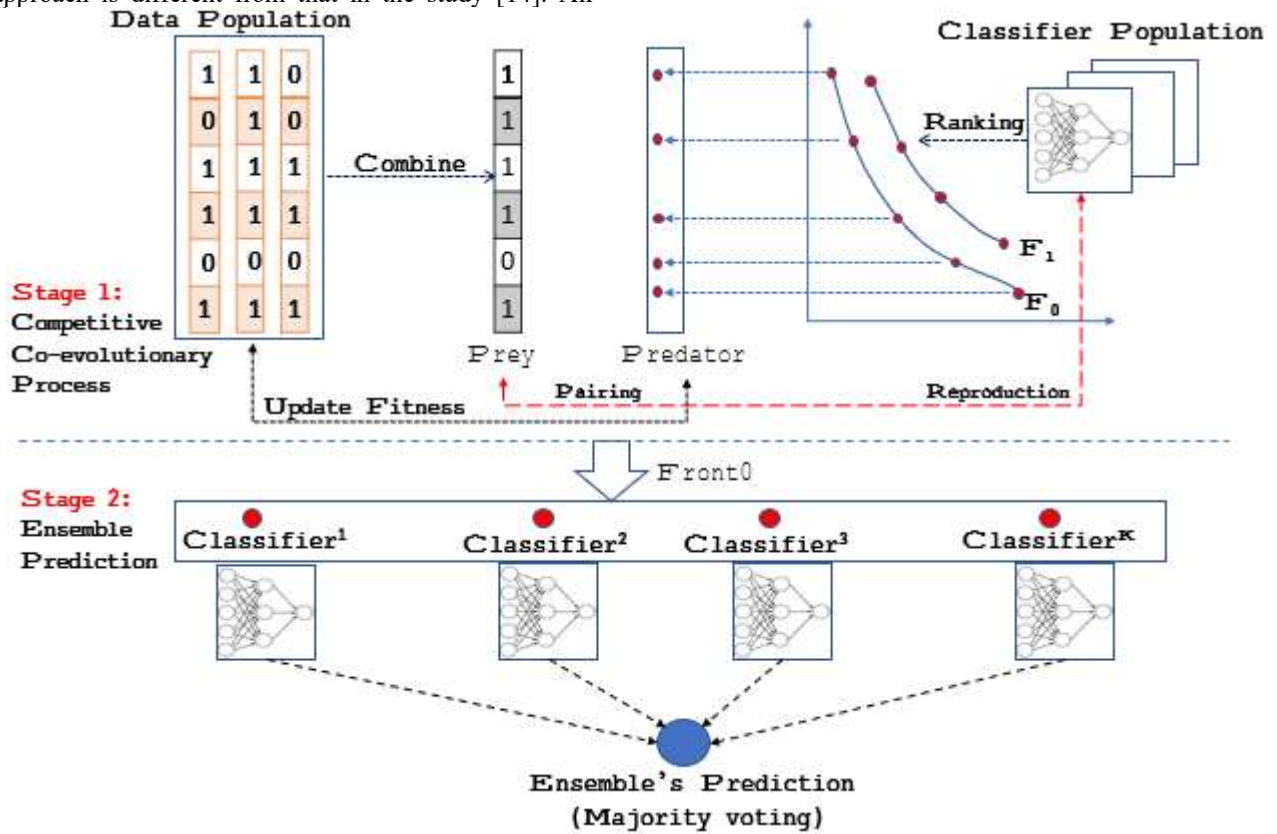


Fig 1. Diagram of the proposed method

It is noting that a common phenomenon in competitive coevolution is premature convergence. Individuals in the classifier population tend to converge with each other. To avoid this phenomenon, in this study we used an additional DIV objective function to make a difference between classifier individual. The DIV is calculated as (3):

$$DIV_i = \frac{1}{(K-1) \sum_{j=1, j \neq i}^N} d(i, j) \quad (3)$$

Where  $N$  is population size;  $d(i, j) > 0$  is the distance between the individual <sub>$i$</sub>  and the individual <sub>$j$</sub> , calculated as (4):

$$d(i, j) = |ACC_i - ACC_j| \quad (4)$$

$K$  is the number of individuals belonging to the circle whose center is current point and radius  $\delta$  is calculated according to (5):

$$\delta = \text{Max}(|ACC_i - ACC_{i+1}|) \quad (i = 1 \rightarrow N) \quad (5)$$

#### The co-evolutionary process

After the classifier individuals have calculated the objective functions, these individuals will be ranked based on Nondominated sorting and Crowding distance methods of the NSGA-II algorithm [15]. The individuals are arranged on different fronts. The individuals on Front 0 ( $F_0$ ) are the best individuals and we choose these individuals as candidates to pair with the data individuals. The Fitness value of each sample data is calculated according to (6).

$$Fitness[i] = \alpha * Fitness[i] + \beta * \frac{M_i}{S_i} \quad (6)$$

Where  $M_i$  is the number of times the  $i^{\text{th}}$  sample data has been wrongly classified;  $S_i$  is the number of times the  $i^{\text{th}}$  sample data is selected in the data population;  $\alpha$  and  $\beta$  are two factors represent the impact of past and present data, ( $\alpha + \beta = 1$ ). If  $\alpha$  is large, past values will have a greater influence and vice versa.

After updating Fitness values, individuals in population 1 will be recreated based on (1) and the process is repeated.

The co-evolution process takes place according to the number of generations already defined.

#### C. Ensemble's prediction

At the end of the coevolutionary process, we obtained a population of classifiers. Instead of just picking the best one, we select all  $K$  individuals on  $F_0$  to form a community of classifiers. There are two ways to produce final classification results. Either combining these classifiers into a stronger one using the Adaboost algorithm [16], or using the majority voting technique to produce the final result. In this study we use the second solution, taking the result given by the majority of  $K$  classifiers.

### III. EXPERIMENTAL RESULTS

#### A. Data description

In this study, we use seven sample data sets to evaluate the proposed algorithm. They are benchmark data sets downloaded from the UCI Machine Learning Repository [17]. The properties of these data sets are summarized in Table I. All of these data sets are normalized to [0,1]. Besides, we use the 10-fold-cross-validation strategy to evaluate results. The final result is averaged.

TABLE I. THE BENCHMARK DATASET

Dataset	Number of instances	Class	Features
Heart	270	2	13
Iris	150	3	4
Pima	768	2	8
Sonar	208	2	60
Segment	2310	7	19
Vehicle	846	4	18
Inosphere	351	2	34

#### B. Parameter setting

The experimental parameters used in the algorithm are shown in Table II and Table III. In Table III, the  $\alpha$  parameter represents the influence of the past value while the  $\beta$  parameter represents the influence of the current result. To determine the best value of this coefficient pair, we experimented on different value pairs in the range [0, 10]. The best value will be our choice. From the results in Table III, it can be seen that except for the Iris dataset, for all remaining data sets, the  $\alpha$  value is higher or equal to  $\beta$  value. This shows that past values play a more important role in comparison with current values.

TABLE II. THE PARAMETERS SETTING

Method	Parameters	Value
ANN	Learning rate	0.02
	Number hidden nodes	6
	Alpha value	2
	Epoch	200
	$[-\delta, +\delta]$	$[-1.5, 1.5]$
NSGA-II	Population size	200
	Probability of mutation	$1/(\text{Individual length})$
MOCPC EA	Iterations	100
	Population size 1	20
	Population size 2	100

TABLE III. THE COEFFICIENTS FOR EACH DATASET

	Heart	Inosphere	Iris	Pima	Sonar	Vehicle	Segment
$\alpha$	9	7	0	8	6	5	9
$\beta$	1	3	10	2	4	5	1

TABLE IV. THE RESULTS OF THE MOCPC EA AND STATE-OF-THE-ART ALGORITHMS

	ComCoE(DET)	ComCoE RBFANN	MOCPC EA
Heart	83.4	-	84
Inosphere	92.5	92.3	95

#### C. Test scenarios

##### a) Scenario 1: Performance comparison with baseline algorithms.

The three baseline algorithms used in this study include ANN using the backpropagation algorithm (BP); Genetic algorithm (GA) and NSGA-II algorithm (NSGA). The experimental results are shown in Fig 2. It can be easy to see that the proposed algorithm is better than all baseline algorithms on all test problems. Especially with the Segment problem, the proposed algorithm gives remarkable results. While the BP only gives an accuracy of 39%, algorithms using single and multiple-objective (GA and NSGA) have much better results, reaching 73% and 72% respectively. However, compared to the results achieved by the MOCPC EA, the results are even more impressive. It can reach 93%, an increase of 54% compared to the BP and about 20% compared to the other two algorithms. Through this experiment, we can see the strength of the proposed algorithm

##### a) Scenario 2: Verify the effect of the data population (or the sample weighting mechanism).

To see the effect of the data population, we modified the MOCPC EA algorithm and created a new version named MOV T. The MOV T differs only from the MOCPC EA in evaluating the ACC objective, instead of only evaluating on difficult samples, the MOV T evaluates on the entire data set. The result is shown in Fig 3. The MOCPC EA algorithm continues to give better results across all test problems. Differences range from 1% (for the Heart dataset) to 7% (for the Segment dataset). Although the results are not too large, it is clear that the effect of using a data population to weight and select difficult data samples.

##### a) Scenario 3: Performance comparison with other competitive co-evolutionary algorithms.

In order to see the strength of the proposed algorithm, we compared the proposed algorithm with the ComCoE (DET) and the ComCoE RBFANN [10]. The results are shown in Fig 4. In general, the results of the algorithms are quite the same. The MOCPC EA gives the best results 4/7 test data sets. While in other data sets, the different figures are very small. In Inosphere and Vehicle, the proposed algorithm gives better results, the difference is 2.5% and 2.1% respectively. The two algorithms have been published and proven their strength. Meanwhile, the proposed algorithm can give similar and better results in some cases compared to these two algorithms. Therefore, it shows the feasibility and strength of the proposed algorithm.

<b>Iris</b>	97.8	-	98
<b>Pima</b>	78.2	77.5	78
<b>Sonar</b>	85.1	71.8	85
<b>Vehicle</b>	77.9	69.4	80

Note: “-” denote unavailable results.

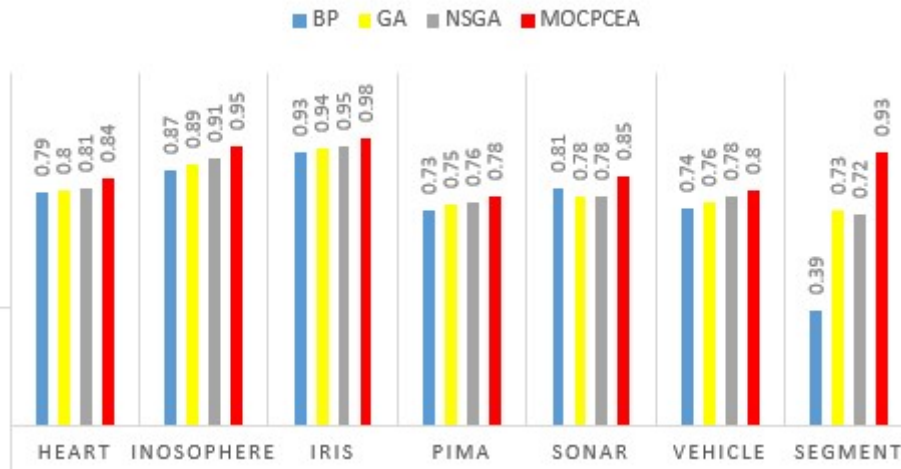


Fig 2. The classification results of the MOCPCEA and Baseline algorithms

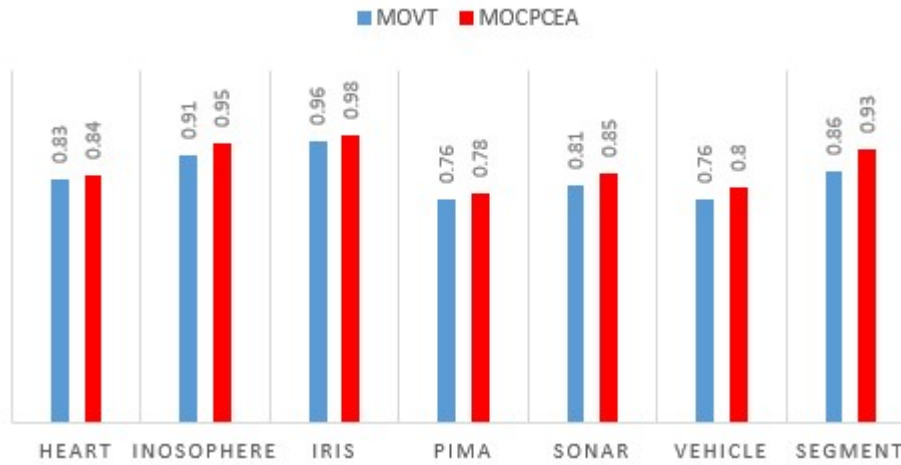


Fig 3. The classification results of the MOCPCEA and MOVt- a version do not use data population

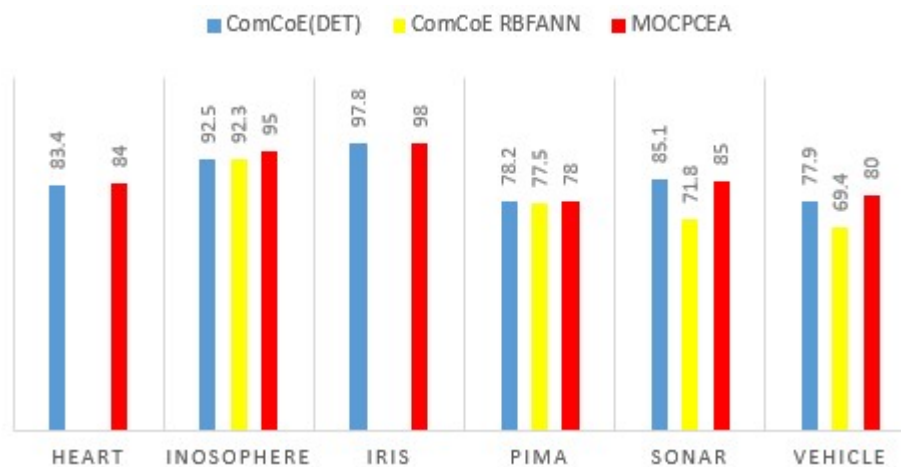


Fig 4. The classification results of the MOCPCEA and the state-of-the-art competitive co-evolutionary algorithms

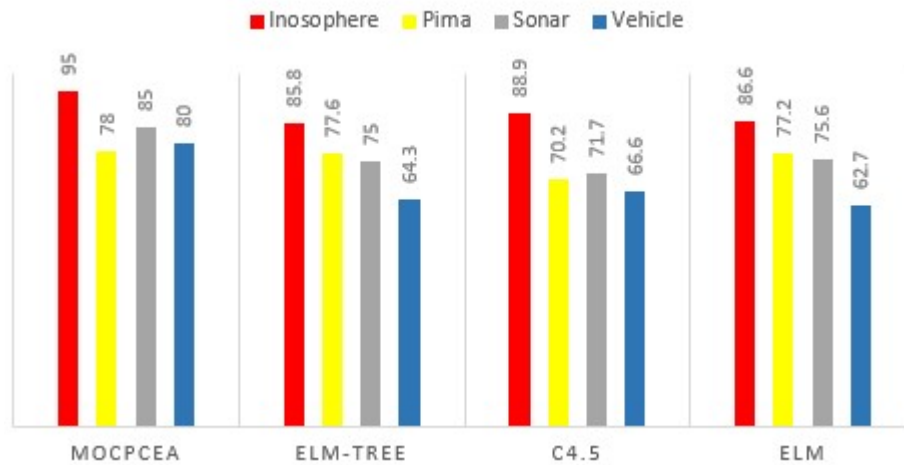


Fig 5. The classification results of the MOCPEA and other machine learning algorithms

#### IV. CONCLUSION

In this paper, the authors proposed a Prey-Predator model with dual-population multi-objective competitive co-evolutionary algorithm. The proposed method utilized two sampling schemes: All-versus-all and Elites sampling. In the first scheme, a predator (i.e. classifier) was paired with all Preys (i.e. data individuals). In the second one, prey was paired with predators which were the best individuals in terms of both the ACC and the DIV objectives. The best predators at the last generation of the co-evolutionary process were selected to make the final decision. The authors utilized a majority voting mechanism to get the final classification result from an ensemble of the selected predators. Through the comparison results on seven benchmark problems with baseline algorithms, state-of-the-art competitive co-

#### REFERENCES

- [1] P. R. Ehrlich and P. H. Raven, "Butterflies and plants: A study in coevolution," *Evolution*, vol. 18, no. 4, pp. 586–608, 1964.
- [2] Holland, John H. "ECHO: Explorations of evolution in a miniature world." In *Proceedings of the Second Conference on Artificial Life*. 1990.
- [3] E. Uchibe and M. Asada, "Incremental coevolution with competitive and cooperative tasks in a multirobot environment," *Proc. IEEE*, vol. 94, no. 7, pp. 1412–1424, 2006.
- [4] Yang, Zhenyu, Ke Tang, and Xin Yao. "Large scale evolutionary optimization using cooperative coevolution." *Information Sciences* 178, no. 15 (2008): 2985–2999.
- [5] Krawiec, Krzysztof, and Bir Bhanu. "Coevolutionary feature learning for object recognition." In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 224–238. Springer, Berlin, Heidelberg, 2003.
- [6] Hajati, Farshid, Caro Lucas, and Yongsheng Gao. "Face Localization using an Effective Co-Evolutionary Genetic Algorithm." In *2010 International Conference on Digital Image Computing: Techniques and Applications*, pp. 522–527. IEEE, 2010.
- [7] Rabil, Bassem S., Safa Tliba, Eric Granger, and Robert Sabourin. "Securing high resolution grayscale facial captures using a blockwise coevolutionary GA." *Expert Systems with Applications* 40, no. 17 (2013): 6693–6706.
- [8] Van Truong VU, Lam Thu BUI, Trung Thanh NGUYEN, "A Coevolutionary approach for classification problems: Preliminary results." In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 81–86. IEEE, 2018.
- [9] Van Truong VU, Lam Thu BUI, Trung Thanh NGUYEN, "An Ensemble Co-Evolutionary based Algorithm for Classification Problems." *Research and Development on Information and Communication Technology* 2019, no. 1 (2019).
- [10] B. Y. Hiew, S. C. Tan, and W. S. Lim, "A double-elimination-tournament-based competitive co-evolutionary artificial neural network classifier," *Neurocomputing*, vol. 249, pp. 345–356, 2017.
- [11] I. Goodfellow et al., "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [12] Chandra, Rohitash, and Gary Wong. "Competitive two-island cooperative co-evolution for training feedforward neural networks for pattern classification problems." In *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2015.
- [13] Paredis, Jan. "Coevolutionary computation." *Artificial life* 2, no. 4 (1995): 355–375.
- [14] M. Castellani, "Competitive co-evolution of multi-layer perceptron classifiers," *Soft Comput.*, vol. 22, no. 10, pp. 3417–3432, 2018.
- [15] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evolut. Comput.* 6 (2002) 182–197.
- [16] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: *Proceedings of the European Conference on Computational Learning Theory*, 1995, pp. 23–37.
- [17] [46] K. Bache, M. Lichman, UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013.
- [18] [40] R. Wang, Y.L. He, C.Y. Chow, F.F. Ou, J. Zhang, Learning ELM-tree from big data based on uncertainty reduction, *Fuzzy Sets Syst.* 258 (2014) 79–100.

#### ACKNOWLEDGMENTS

This research is funded by project "A smart surveillance system based on facial recognition for border and immigration controls" under Bilateral and Multilateral Research Programs (Grant no. HNQT/SPDP/14.19).