A Coevolutionary approach for classification problems: Preliminary results

Van Truong VU
Le Quy Don Technical University
Ha Noi, Viet Nam
truongvv@mta.edu.vn

Lam Thu BUI
Le Quy Don Technical University
Ha Noi, Viet Nam
lam.bui07@gmail.com

Trung Thanh Nguyen
Liverpool John Moores University
Liverpool, United Kingdom
T.T.Nguyen@ljmu.ac.uk

***Abstract*—In the current classification problems, how to eliminate redundant features, find out important features, and choose appropriate classifiers for these feature set play a vital role. This paper presents a co-operative co-evolution approach (COCEA) with dual populations for optimizing both the artificial neural network (ANN) model and feature subset selection simultaneously. In COCEA, each feature subset is encoded as a binary string. Meanwhile, an ANN is represented in a matrix-form with real values of its weight and bias. During the process of evolution, a co-Operation mechanism is used to integrate the two populations and the final solution is the combination of the two most elite individuals of each population in a hope that the final solution will satisfy both ANN optimization and feature subset selection. The performance of COCEA is examined on both the well-known benchmark problems and Oil Spill dataset in SAR Images. In comparison with the other algorithms, experimental results illustrate that COCEA can significantly outperform other peer algorithms in terms of classification accuracy.**

***Index Terms*—Feature Selection, Co-evolution, Co-operative Co-evolution, ANNs, Classification.**

I. Introduction

In machine learning, multiclass or multinomial classification is one of the fundamental tasks which classifies instances into two or more classes. Various algorithms have been developed to address multi-class classification problems, generally it can be classified into three broad categories: supervised, unsupervised and reinforcement learning algorithms. Among them, supervised classification is the most frequently utilized to solve real-world problems. The supervised classification algorithms can be grouped [1] into Logic based algorithms (decision trees and rule-based classifiers); Perceptron-based techniques (Single layered perceptrons, Multilayered perceptrons, Radial Basis Function (RBF) networks); Statistical learning algorithms (Naive Bayes classifiers, Bayesian Networks); Instance-based learning (k-Nearest Neighbour-kNN) and Support Vector Machines. In this research the authors utilize MLP as a base method for the co-operative co-evolution approach.

In machine learning, each instance in dataset is presented by same set of a large number of features. Many of these features are either irrelevant or redundant to the classification target [2]. The *Feature subset selection* techniques are often used in order to overcome these problems. Feature subset selection is the process of identifying and removing as many irrelevant and redundant features as possible [3]. This helps to reduces the dimensions of the data and enables algorithms to run faster and more effectively. In contrast to other dimensionality reduction techniques like PCA (Principal Component Analysis) or compression, feature subset selection techniques only select a subset of them rather than alter the original representation of the variables. In general, depending on how they combine the feature selection search with the construction of the classification model, the Feature subset selection techniques can be organized into: filter approach, wrapper approach [4]. In Filter approach [5], a feature relevance score is first calculated, and features having low-scoring are removed. After that, the selected features becoming input of the classification algorithm, they are done independently of the classification algorithm (they ignore the interaction with the classifier). Although they are computationally simple but they are assessed by another criterion rather than through classifiers, which leads to worse classification performance. Conversely, in the wrapper approach, the feature subset selection algorithm is wrapped around the classification function. Various subsets of features are generated and the predictor plays as a black box to evaluate these subsets. There is a number of search algorithms used to find out optimal subsets [4] such as the Sequential search [6] or Evolutionary algorithms such as Genetic Algorithm (GA) [7]. In general, the wrapper approach outperforms the filter approach.

Many studies have built wrapper approach models using ANN as the classification function. In [8] the authors proposed a hybrid method based on ant colony optimization and artificial neural networks (ANNs) to address feature selection for the medical diagnosis problems. In [9] the authors used a wrapper-based GA feature selection procedure for the learning disabilities (LD) diagnosis problem. The authors wrapped SVM within the GA feature selection procedure and using ANN learner in the classification stage. The experimental results show that ANN in general performs

978-1-5386-7983-8/18/$31.00 ©2018 IEEE

better than SVM and the wrapper method achieves the best prediction accuracy.

Co-evolution approach is also used to solve this problem, In [10] the authors presented a hybrid approach based on a co-operative co-evolutionary algorithm with dual populations for designing the RBFNN (Radial Basis Function Neural Network) models with feature selection. In this research each feature is encoded as a binary string and a RBFNN structure is encoded in a real-encoded matrix-form. The authors used five objectives to evaluate the fitness of individuals. The proposed method is test on 26 datasets from UCI and Statlog Repository. The result showed that the proposed algorithm was able to obtain the better results on complicated classification tasks compared with other training algorithms.

Inspired by the work done by Jin Tian [10] and Potter [11], in this paper, we propose a wrapper method using a co-operative co-evolution approach with dual population for optimizing ANN classifier with feature selection. The proposed method is investigated on the oil spill dataset which are created by the authors and the other real-word classification problems.

II. RELATED WORKS

A. The ANN

Neural network is considered to be a powerful tool to solve the classification problems that are nonlinear and complex. When working with ANN there are two major problems that need to be addressed: choosing a network architecture and initializing the set of weights values. The weights initially are randomly generated then during the learning process; they will be calibrated gradually by back-propagation algorithm (BP). Since BP is a gradient descending process, it may get stuck in local minima in the weight-space. This greatly depends on the initial randomly generated value of the weight set. There have been many studies addressing this problem [12][13]. In this study, we will use a sub-population of co-evolution approach to optimize this initial set of weights.

B. DE Algorithm

The Differential Evolution Algorithm was first proposed by Storn and Price [13] to solve real-parameter optimization problems. There are some variants of the DE algorithm [14]: Variants with arithmetic recombination (DE/current-to-rand/1; DE/current-to-best/1); Variants with discrete recombination operator (DE/rand/1/bin); and Variants with combined arithmetic-discrete recombination (DE/current-to-rand/1/bin). Among them, "DE/rand/1/bin" is the most commonly used variation.

In DE, there are two important parameters are CR and F. In which, CR controls the influence of the parent in the generation of the offspring. Higher values mean less influence of the parent in the features of its offspring. F scales the influence of the set of pairs of solutions selected to calculate the mutation value.

C. CoEvolutionary mechanism

The co-evolutionary algorithm is a very young field of evolutionary computation. Coevolution has been applied mostly to two-agent games. Besides, it has also been used successfully in classification, by evolving NNs and decision trees [15].

There are two categories of CoEA: competitive and cooperative. In competitive approach, individual fitness is determined via competitions with other individuals. In co-operative approach, the fitness of an individual is determined in collaborations with another individual.

A general framework for cooperative Co-evolutionary has been introduced in [11] with the cooperative co-evolutionary genetic algorithm (CCGA). The general idea as follows: Divide the overall solution into many sub-components; build up a sub-population for each of these sub-components. Each of these sub-populations will undergo evolution to find the optimal sub-components. It is noted here that, at each generation, in order to calculate the fitness value of each individual, it is necessary to combine with individuals in other sub-populations to form a complete solution. The output solution will be created by integrating all of these sub-components.

III. PROPOSED METHOD

The general diagram of the co-Evolution approach (COCEA) is given in Fig.1. In this study, the authors utilize a dual population approach (each population corresponds to a sub-component) to simultaneously solve two tasks: selecting the most important input features and finding the optimal weight of ANNs for these input features. It means the overall solution will be divided into two sub-components, the first one being the ANN inputs and the second component is the ANN's weight. In order to calculate the fitness value, each individual in the first population needs to be associated with the individual of the second population and vice versa (i.e. each ANN needs to know the input features and the weight to calculate the output value and the fitness value). Undergoing evolution, the overall solution will eventually become a combination of the best individuals (elites) from two populations. A more detailed explanation of COCEA will be showed in here.

A. Encoding

Two populations have two different coding strategies. In the first population (Feature population), suppose that this population has m individuals $(Ind_1^1, Ind_2^1, ..., Ind_m^1)$, each individual Ind_i^1 will be encoded as a binary string. In particular, the value 1 corresponds to the feature being

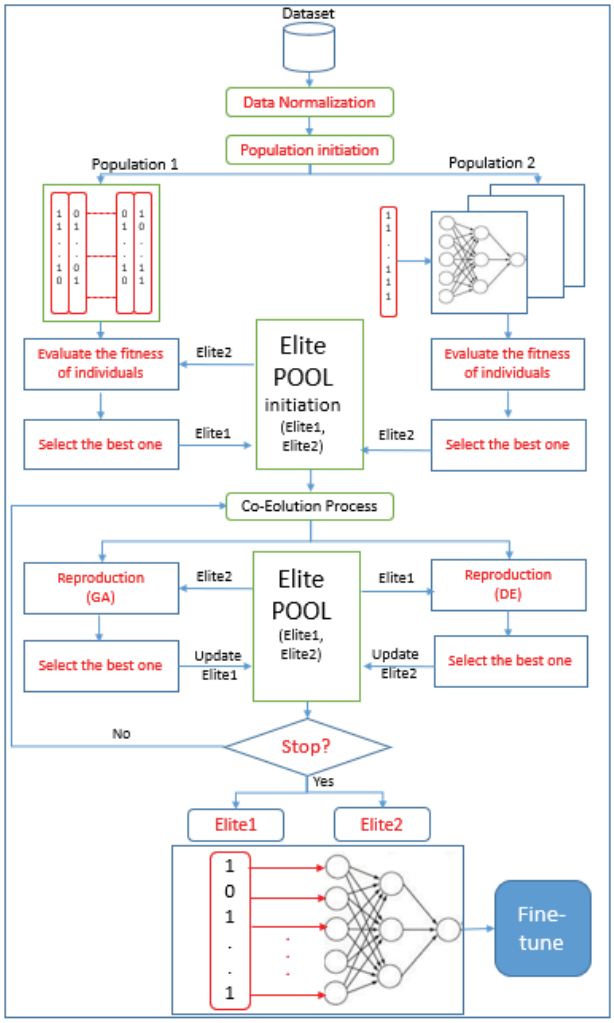

Figure 1. Diagram of the proposed method.

selected and vice versa. For example, with individual 1 $Ind_1^1 = [01100]$ the second and third features are selected and the other ones are eliminated.

In the second population, suppose that this population has n individuals $(Ind_1^2, Ind_2^2, ..., Ind_n^2)$, each individual Ind_i^2 is an ANN. Here, we use a real-encoding to represent an ANN. Note that, in ANN, weight values associated with each vector and node in the network (the weight values associated with nodes are also known as *biases*) can represent an ANN network. Therefore, in this study we encode each individual as a one-dimensional array of real numbers to encode the weight matrix.

B. Initialization

1) Data Normalization: In general, the input features may be continuous, categorical or binary, due to the neural networks can be performed only with numeric data, it is therefore necessary to encode the non-numerical data and carry out data normalization in advance. There are many ways of normalizing data, in this study we use the Min-Max Normalization as follows:

$$X_{normalizaed} = \frac{X_{current} - ((X_{max} + X_{min})/2)}{(X_{max} - X_{min})/2} \quad (1)$$

2) Population initiation: In general, each individual in each population is initialized with random values. While in population 1, individuals are randomly generated as a binary string of 0, 1 that satisfies two constraints: (1) this string must contain at least one bit of 1 (to ensure at least 1 feature selected). (2) There is only one string containing all bits of 1 (to ensure that all selected features are considered). In population 2, each individual will be randomly assigned an array of real numbers.

At first, individuals with all of the bits are 1 (i.e. all features are selected) will be used to calculate the fitness values of all individuals in the population 2, after that make a ranking based on the fitness value and choose the best individual as an elite (named $Elite_2$) and put it into the elite pool. $Elite_2$ is then used to calculate the fitness values of the individuals in population 1, after ranking, the best one (named $Elite_1$) is put into the elite pool.

C. The Co-evolution process

After the initialization of the population is the process of co-evolution. This is an iterative process, basically the steps are quite similar to the initialization step when each sub-population will try to evolve and find the elites to update to the pool. The elite in this sub-population will be used for the other sub-population in the next iteration. The elitist selection mechanism is utilized here in both of sub-populations in order to keep the best solutions during the whole co-evolution process. Suppose the Elite Pool consists of two elements $< Elite1, Elite2 >$. If there is any one (denoted by $Elite_1^t$ at the loop t^{th}) dominating the corresponding elite in the pool (i.e. $Elite1$) then the $Elite1$ will be substituted by $Elite_1^t$ and the fitness of the new $Elite1$ is updated as well.

The main difference between the two sub-populations is the evolutionary operation (the Reproduction step in the Fig.1). Because of the different coding (binary and real encoding), evolutionary operation applied to two distinct sub-populations are vary. Specifically, in Population 1 we use two points crossover and *Bit flip mutation* operations. Meanwhile, the operations in the Differential Evolution (DE) algorithm are applied for Population 2. Details of these operations will be described below.

D. Mutation

The main purpose of mutation is to maintain the population diversity by modifying some genes of offspring with a

small probability. In the first sub-population we use *Bit flip mutation* operator to work with binary encoding. Meanwhile *Differential evolution mutation* used for the second sub-population.

1) Bit flip mutation: Assume $prop$ is the probability of mutation, an individual in sub-population 1 has the form $Ind_i^1 = [b_j]\,(b_j \in [0,1], j = 1,2,\ldots,N)$; N is the number of input features. If there is any b_j having the random value smaller than *prop* than flip b_i value (i.e. invert 0 to 1 and vice versa).

2) Differential evolution mutation: Foreach current individual ($x_{i,G}$) and randomly select three parent individuals (x_{r1}, x_{r2}, x_{r3}). A mutant vector is produced by:

$$v_{i,G+1} = x_{r1,G} + K.(x_{r1,G} - x_{i,G}) + F.(x_{r2,G} - x_{r3,G}) \quad (2)$$

Where $i, r_1, r_2, r_3 \in \{1, 2, ..., NP\}$ (NP is the population size) must be different from each other. F is the scaling factor and K is the combination factor.

E. Crossover

The main purpose of crossover is to explore the entire search space thereby avoiding local minima and towards global optimal. In general, there are many types of crossover, in this study we use two main types: Two points crossover and Differential evolution crossover operators.

1) Two points crossover: First, two parents ($Parent_A$ and $Parent_B$) are picked randomly from the sub-population 1. Next, randomly select two cutting points J and $K (J < K)$ and create two offspring. The sub-strings falling between the two cutting points in the parent A and B are swapped and *Offspring 1* and *Offspring 2* are generated.

The Offspring's fitness are then calculated and compared with their parents to decide whether to replace them or not.

2) Differential evolution crossover: The parent vector ($X_{ji,G}$) is mixed with the mutated vector ($v_{ji,G+1}$) to produce a trial vector $u_{ji,G+1}$

$$u_{ij,G+1} = \begin{cases} v_{ji,G+1} \text{ if } (rnd_j \leq CR) \text{ or } j = rn_i \\ x_{ji,G} \text{ if } (rnd_j > CR) \text{ and } j \neq rn_i \end{cases} \quad (3)$$

where $j = 1, 2, ..., D; rnd_j \in [0,1]$ is the random number; CR is crossover constant $\in [0,1]$ and $rn_i \in \{1, 2, ..., D\}$ is the randomly chosen index.

F. Fitness fuction

In the classification, the accuracy is the most important factor; it is to be measured through error values. There are many available metrics to quantify the error, but the most commonly used metric is the Mean Squared Error (MSE). In this paper the authors choose MSE which is calculated according to the formula 4 as the fitness function.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(R_t - O_t)^2 \quad (4)$$

Where: n - the number of data value; R_t- real value; O_t- output value.

G. Fine-tune

The COCEA approach can be easy to find out the regions that contain global maxima points but they are very hard to find out exactly these extreme points. From this characteristic, we used MOEA to optimize the ANNs first (i.e. use COCEA to take ANNs escape from the extreme locals and approach to global extremes), then used BP (Fine-tune) to put them closer to the global extreme points.

IV. Experiments

In order to evaluate the performance of the proposed method, the authors performed experiments on 4 different datasets (Include cases of small (8), medium (34) and large numbers (60) of input features). Among them, there is an oil spill dataset on satellite images (made by the authors), and 3 data sets on UCI repository (Ionosphere, Pima, Sonar). Most datasets are divided into three subsets (training set: 50%; validation set: 25% and 25% for testing), except Sonar dataset is divided into two subsets (training set: 50% and 50% for testing). For each dataset, the final result is averaged over 20 runs of the algorithms. The author compares the algorithm proposed with the conventional algorithms (i.e. ANN, DE) and some state-of-art algorithms in [10].

A. Dataset description

1) Oil spill dataset: SAR (Synthetic Aperture Radar images) images have been widely used for oil spill detection. In general, oil spill detection algorithms traditionally base on the features for classifying dark objects on SAR images into oil spills or look-alike.

The features can be generally grouped in three major categories [16]: the geometrical characteristics (e.g. area, perimeter, complexity), the physical behavior of oil spills (e.g. mean or max backscatter value) and the oil spill context in the image (e.g. number of other dark formations in the image).

The data [17] used in this research were taken in 2007, 2008, 2009 in the East Sea. The total SAR image data are 16 photos with 108 dark objects, including 68 oil spills and 40 look-alikes. There are 8 features Perimeter (P), Area (A), the shape index (Sf), the complexity of the (PT), the standard deviation of gray values belong oil spill (Osd), the average number of gray levels belong oil spill (Osm), the largest value within the gray oil spill (Max), and the smallest gray level values belong oil spill (Min)).

2) UCI Datasets: UCI repository contains lots of sample datasets for classification problems. In this paper the authors choose three typical numerical two-class classification datasets to conduct experiments. These datasets are briefly characterized in Table I.

Table I
DATASET USED IN EXPERIMENTS

Dataset	Train	Val	Test	Class	Features
Ionosphere (Inono)	175	88	88	2	34
Pima	384	192	192	2	8
Sonar	104	-	104	2	60
Oil Spill	54	27	27	2	8

B. Parameter settings

Parameters of algorithms used in this study are presented in the Table II.

Table II
PARAMETER SETTING

Method	Parameters	Value
ANN	Learning rate	0.3
	Number hidden nodes	20
	The iterations of BP	1000
	Alpha value	2
DE	CR / F	0.9 /0.5
	Lower bound /Upper bound	-1.5/ 1.5
COCEA	The iterations of COCEA	10
	The iterations of DE	50
	The iterations of Fine-tune	100
	The Population size	100

C. Results and analysis

1) Experiment 1: Verify the proposed algorithm with single algorithms using oil spill dataset: In this experiment, the authors examine whether the proposed method (i.e. the dual-population Co-evolution approach using DE and MLP) is better than DE and MLP or not? At the same time, we want to check whether this evolutionary algorithm inherit the best of both algorithms or not? There are two points to note here, the DE algorithm, a population-based evolutionary algorithm, has the ability to find out the regions containing global extremes but it is difficult to find exactly these extremes. Meanwhile, the MLP algorithm, a gradient descending-based algorithm, is easy to converge on a global extreme if it is initialized close to this position, but it is also may get stuck in local extremes.

After 20 runs of algorithms, we analyze the following statistical values: Average (Ave), Standard Deviation (Std), $Best$ (the smallest error value), $Worse$ (the biggest error value). As can be seen from Table III, the proposed method outperforms DE and MLP in both of the train and test data in terms of average, best and worse values followed by MLP and finally DE algorithm. In addition, with regard to the Std value it can be seen that, although the results are inferior (because only the area around the extremes can be found), the results of the DE algorithm are most stable over 20 runs. The second best result is the proposed algorithm and worst is the MLP algorithm. This is in full accordance with the notice given above. Besides, it also shows that the proposed algorithm inherits elitism from both base algorithms (i.e. can finds a region containing global extreme and converges to this point).

In addition to the classification results, the number of features has been reduced and as shown in Fig.2, it is possible to know which features are most important and vice versa. Forexample, in Fig.2, the SF, P, A, and OSM are the most important features (The number of times selected after 20 runs is 20, 19, 19 and 18 respectively), while other features will play less important roles, especially OSD (only used 14 times out of a total of 20).

Table III
THE PERFORMANCE OF THE ALGORITHMS ON OIL SPILL DATASET

	Test Data			Train Data		
	COCEA	MLP	DE	COCEA	MLP	DE
Ave	**0.961**	0.888	0.650	**0.993**	0.938	0.570
Std	0.083	0.113	**0.018**	0.016	0.067	**0.012**
Best	1.000	1.000	0.667	1.000	1.000	0.593
Worse	**0.741**	0.444	0.630	**0.944**	0.667	0.759

2) Experiment 2: Verify the proposed algorithm with other state-of-art algorithms using UCI datasets: In order to evaluate the potential of the proposed algorithm we conduct a comparison with the reference algorithms used in the study [10]. We mainly want to compare the proposed method with another co-evolution algorithm named DC-RBFNN from [10].

The performance comparisons are presented in Table IV via the Ave, Std, $Best$ and $Worse$ values. The best mean metric value is highlighted.

Similar to the observation in Table III, the proposed method can achieves better metric values in all of comparisons in term of Ave and $Best$ values. Especially with Sonar and Pima dataset, the proposed method outperforms the others (84.2% compared with 78.0% in Sonar and 91.9% compared with 76.3% in Pima). However, there is one

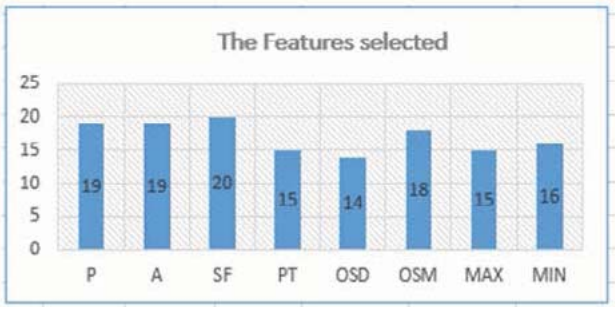

Figure 2. Number of features selected by the proposed method.

point to note: with the Pima dataset, despite giving the best average result, in the 20 experiments there are still some

Table IV
TESTING ACCURACIES OF THE PROPOSED METHOD (COCCEA) AND SOME OTHER METHODS

Datasets		COCEA	DC-RBFNN	Bayes	Boost	C4.5	K-means	KNN	MLP	PNN	SMO
Sonar	Ave	**0.842**	0.780	0.656	0.816	0.725	0.709	0.758	0.820	0.532	0.773
	Std	0.042	0.038	0.063	0.037	0.044	0.054	0.034	0.051	0.016	0.044
	Best	0.923	0.856	0.769	0.885	0.827	0.846	0.837	0.923	0.567	0.863
	Worst	0.769	0.702	0.558	0.745	0.635	0.587	0.683	0.726	0.500	0.673
Iono	Ave	**0.942**	0.941	0.831	0.922	0.898	0.885	0.791	0.896	0.942	0.885
	Std	0.054	0.028	0.045	0.024	0.035	0.045	0.034	0.032	0.022	0.024
	Best	1.000	0.977	0.908	0.977	0.954	0.955	0.864	0.955	0.989	0.921
	Worst	0.864	0.864	0.750	0.875	0.784	0.773	0.716	0.820	0.909	0.841
Pima	Ave	**0.919**	0.763	0.754	0.752	0.735	0.732	0.715	0.748	0.694	0.773
	Std	0.193	0.019	0.026	0.030	0.035	0.024	0.030	0.024	0.030	0.025
	Best	1.000	0.802	0.797	0.807	0.792	0.781	0.766	0.792	0.750	0.813
	Worst	0.448	0.729	0.708	0.677	0.646	0.688	0.656	0.693	0.630	0.724

runs (about 2 to 3 times) in which COCEA gives unusually poor results. This is a reason why in Pima dataset, the COCEA's Std and $Worse$ values are significantly higher than other algorithms. This may be explained by the fact that maybe the iterative co-evolution is not sufficient to find regions containing the global extremes, leading to solution still trapped at the local extremity after the refining step (fine-tune).

V. Conclusion

In this paper, a dual-population based co-operative co-evolutionary approach (COCEA) is presented. In COCEA, two sub-populations are simultaneously maintained to achieve both good classification accuracy and prominent input features. These sub-populations interact with each other via an elitist selection mechanism. The performance of the proposed algorithm is compared with the conventional algorithms and other state-of-art methods on the 4 datasets. The empirical results on the test instances demonstrated the effectiveness of new co-operative co-evolutionary approach for designing the ANN to solve with the classification problems. This is our preliminary study, a series of other co-evolutionary variants with better results on other benchmark problems will be tested and presented in the near future.

Acknowledgment

This work was supported by grants from the Newton Fund, under the NAFOSTED - UK Academies collaboration programme.

References

[1] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160 (2007): 3-24.

[2] T.-S. Li, "Feature Selection for Classification By Using a Ga-Based Neural Network Approach," J. Chinese Inst. Ind. Eng., vol. 23, no. 1, pp. 55–64, 2006.

[3] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," J. Mach. Learn. Res., vol. 5, pp. 1205–1224, 2004.

[4] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," Comput. Electr. Eng., vol. 40, no. 1, pp. 16–28, 2014.

[5] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," Bioinformatics, vol. 23, no. 19, pp. 2507–2517, 2007.

[6] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," Pattern Recognit. Lett., vol. 15, no. 11, pp. 1119–1125, 1994.

[7] K. Deb, "Genetic Algorithm in Search and Optimization: The Technique and Applications," Proc. Int. Work. Soft Comput. Intell. Syst., pp. 58–87, 1998.

[8] R. K. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," Expert Syst. Appl., vol. 33, no. 1, pp. 49–60, 2007.

[9] T. K. Wu, S. C. Huang, and Y. R. Meng, "Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities," Expert Syst. Appl., vol. 34, no. 3, pp. 1846–1856, 2008.

[10] J. Tian, M. Li, and F. Chen, "Dual-population based coevolutionary algorithm for designing RBFNN with feature selection," Expert Syst. Appl., vol. 37, no. 10, pp. 6904–6918, 2010.

[11] M. A. Potter and K. A. Jong, "A cooperative coevolutionary approach to function optimization," pp. 249–257, 1994.

[12] L. T. Bui, V. Truong Vu, and T. T. Huong Dinh, "A novel evolutionary multi-objective ensemble learning approach for forecasting currency exchange rates," Data Knowl. Eng., vol. 114, no. July 2017, pp. 40–66, 2018.

[13] T. T. H. Dinh, V. T. Vu, and T. L. Bui, "A multi-objective ensemble learning approach based on the non-dominated sorting differential evolution for forecasting currency exchange rates," Proc. - 2016 8th Int. Conf. Knowl. Syst. Eng. KSE 2016, pp. 96–102, 2016.

[14] E. Mezura-Montes, M. Reyes-Sierra, and C. A. C. Coello, "Multi-objective Optimization Using Differential Evolution: A Survey of the State-of-the-Art," Adv. Differ. Evol., pp. 173–196.

[15] A. P. Engelbrecht, Computational Intelligence, 2007.

[16] K. N. Topouzelis, "Oil spill detection by SAR images: Dark formation detection, feature extraction and classification algorithms," Sensors, vol. 8, no. 10, pp. 6642–6659, 2008.

[17] Le Minh, Hang, and Truong Vu Van. "A Combination Method of Differential Evolution Algorithm and Neural Network for Automatic Identification Oil Spill at Vietnam East Sea." Journal of Geological Resource and Engineering 4 (2016): 184-193.