J Inf Process Syst, Vol.13, No.2, pp.321~339, April 2017
https://doi.org/10.3745/JIPS.02.0054

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Multi-Objective TRIBES/OC-SVM Approach for the Extraction of Areas of Interest from Satellite Images

Wafaa Benhabib* and Hadria Fizazi*

Abstract

In this work, we are interested in the extraction of areas of interest from satellite images by introducing a MO-TRIBES/OC-SVM approach. The One-Class Support Vector Machine (OC-SVM) is based on the estimation of a support that includes training data. It identifies areas of interest without including other classes from the scene. We propose generating optimal training data using the Multi-Objective TRIBES (MO-TRIBES) to improve the performances of the OC-SVM. The MO-TRIBES is a parameter-free optimization technique that manages the search space in tribes composed of agents. It makes different behavioral and structural adaptations to minimize the false positive and false negative rates of the OC-SVM. We have applied our proposed approach for the extraction of earthquakes and urban areas. The experimental results and comparisons with different state-of-the-art classifiers confirm the efficiency and the robustness of the proposed approach.

Keywords

Image Classification, MO-TRIBES, OC-SVM, Remote Sensing

1. Introduction

The huge amount of remote sensing data requires appropriate techniques to extract areas of interest, such as damages, urban or flooding areas. Many existing methods operate on all areas of interest. They need the availability of a large database covering all of the needed classes.

To overcome this issue, researchers use one-class classification methods that need a database covering only one class. In particular, high interest has been devoted to studying the frameworks of anomaly detection, targets, and one-class classification [1]. Among these one-class classifiers, the One-Class Support Vector Machine (OC-SVM) was introduced by [2] as a support vector method for novelty detection [1]. The OC-SVM is based on the estimation of a support that includes training data. Data are classified according to their inclusion in the estimated support.

Classification performances depend highly on the learning set. For this, Bootstrap AGGregatinG (BAGGING) [3] is used to reduce the influence of the learning set selection. BAGGING is a machine learning technique that improves the performance of single classifiers [4,5]. It obtains higher performances by increasing the accuracy and the diversity of a classifier [5]. Different training data are

Manuscript received July 8, 2015; accepted May 11, 2016.
Corresponding Author: Wafaa Benhabib (wafaa.benhabib@univ-usto.dz)
* Faculty of Mathematics and Computer Science, University of Sciences and Technology of Oran Mohamed Boudiaf (USTO-MB), Oran, Algeria ({wafaa.benhabib, fizazi}@univ-usto.dz)

obtained using a randomly drawn subset from the initial one. The final classification is carried out by majority voting [4].

The authors in [6] proposed a method for controlling the random aspect of BAGGING with an initial decomposition of the learning data, a mono-objective optimization, and a multi-class classification. The final classification is carried out according to a vote that combines all of the sub-training data [5]. We noticed in this work that the initial decomposition of the learning base introduces a new parameter that significantly influences the results. In addition, optimization based on the maximization of the recognition rate can generate new confusions between some of the classes. We are proposing to generate and manage the different bases using the Multi-Objective TRIBES (MO-TRIBES) technique [7–10] in order to overcome these issues.

MO-TRIBES is a parameter-free optimization technique. It manages the search space in tribes composed of agents. Each agent represents a proposition of an optimal training data. MO-TRIBES makes different behavioral and structural adaptations to provide a non-dominated agent that minimizes the false negative and false positive rates of the OC-SVM.

We applied the MO-TRIBES/OC-SVM approach on two different applications. The first one was for extracting images of areas that were damaged caused by the earthquake in Bam, Iran from an IKONOS image. The second application was for extracting images of urban areas from a LANDSAT5 TM image of the western region of Oran, Algeria. The results obtained were compared to the those from classical OC-SVM and BAGGING/OC-SVM classifiers, as well as different classification methods, such as decision trees [11], random forest [12], naïve Bayes [13], and Quadratic Discriminant Analysis [14]. The comparisons showed that our MO-TRIBES/OC-SVM approach obtained the best results.

The paper is organized as follows: Section 2 describes the core methods (i.e., the OC-SVM and MO-TRIBES). Section 3 presents a detailed description of the MO-TRIBES/OC-SVM approach. Datasets used for the experiments, obtained results, and comparative studies are presented in Section 4. The last section concludes the paper.

2. Core Methods

2.1 One-Class Support Vectors Machine (OC-SVM)

Support Vectors Machines (SVMs) are a classification method that were developed by Vapnik [15]. They rely on a linear separation between data belonging to different classes. In order to increase the separability between data, the SVM performs a projection from an original space E to a feature space F of higher dimension. This is done according to a projection function $\varphi(\vec{x})$, such as:

$$\begin{aligned} &\emptyset : E \rightarrow F \\ &\vec{x} \rightarrow \varphi(\vec{x}) \; with \; CARD(E) < CARD(F) \end{aligned} \tag{1}$$

There are many linear separators that can be used. Among them, the SVM opts for the separator that presents the larger margin (i.e., the distance between the closest vectors, which are called the support

vectors). The larger margin is chosen in order to minimize classification errors. Thus, SVMs are naturally binary but their learning has been extended to mono-class classification [2].

Fig. 1 illustrates the OC-SVM principle. As can be seen, the search for the smaller hyper-sphere that contains all learning data is equivalent to the search for the linear separator that maximizes distance from the origin [16].

So, OC-SVM learning is based on the determination of a linear separator $h(\vec{x})$ between the l elements belonging to the same class and the origin. Such as:

$$h(\vec{x}) = \vec{w}.\varphi(\vec{x}) - \rho \tag{2}$$

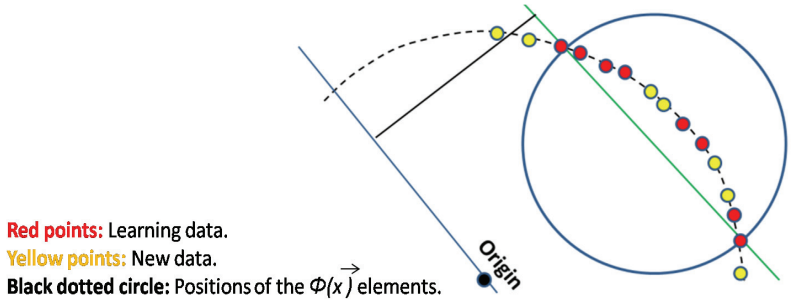


Fig. 1. OC-SVM principle: the search of the smaller hyper-sphere (blue circle) that contains all learning data is equivalent to the search of the linear separator (green right) that maximizes distance from the origin.

With:

- ρ : threshold.
- $\vec{w}$: normal vector.
- $\vec{x}$: vector of elements.
- φ: projection function.

Thus, the goal is to find the linear separator that maximizes the margin and correctly classifies the training data. In practice, we have to introduce positive slack variables ξ_i with $i = 1..l$. These errors are weighted using a constant $\nu \in [0,1]$. Following these different considerations, the determination of the linear separator is modeled by the following optimization problem [16]:

$$\begin{cases} min_{w,\rho,\xi} \ \frac{1}{2}\|\vec{w}\|^2 - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \vec{w}.\varphi(\vec{x}_i) \geq \rho - \xi_i \ with \ i = 1 \dots l \\ \xi_i \geq 0 \ \ with \ i = 1 \dots l \end{cases} \tag{3}$$

Such as:

- If ν is small, we allow for a few errors.
- If ν is large, there is a large margin of errors.

In order to simplify the constraints, the optimization problem (3) is solved using its dual form. The

dual form is obtained by introducing variables α_i called the Lagrangian variables. Each α_i represents the contribution of a training vector $\vec{x}_i$ for the construction of the linear separator. The dual form obtained is:

$$\begin{cases} max_\alpha -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\varphi(\overrightarrow{x_i})\varphi(\vec{x}_j) \\ \sum_{i=1}^{l}\alpha_i = 1 \\ 0 \le \alpha_i \le \frac{1}{vl} \quad with \quad i = 1 \dots l \end{cases} \tag{4}$$

Such as:

- If $\alpha_i = 0$: the vector is correctly classified.
- If $0 < \alpha_i < 1/(v.l)$: the vector is a support.
- If $\alpha_i = 1/(v.l)$: the vector is misclassified.

As it can be noticed from (4), the higher the dimension of the feature space is, the greater the computation cost of the scalar product $\varphi(\vec{x}_i).\varphi(\vec{x}_j)$ is. In order to reduce this computation cost, kernel functions $k(\vec{x}_i,\vec{x}_j) = \varphi(\vec{x}_i).\varphi(\vec{x}_j)$ can be used to implicitly project data [17]. The usual kernel functions are summarized below in Table 1.

Table 1. Usual kernel functions

Kernel	Equation	Parameter
Linear	$\overrightarrow{x_i}.\overrightarrow{x_j}$	//
Gaussian	$exp(-\|\overrightarrow{x_i}-\overrightarrow{x_j}\|^2/(2.\sigma^2))$	σ : Standard deviation
Polynomial	$(\overrightarrow{x_i}.\overrightarrow{x_j}+1)^p$	p : Polynomial order

The kernel functions allow us to rewrite (2) and (4), as follows:

$$\begin{cases} max_\alpha -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j k(\vec{x}_i,\vec{x}_j) \\ \sum_{i=1}^{l}\alpha_i = 1 \\ 0 \le \alpha_i \le \frac{1}{vl} with\ i = 1 \dots l \end{cases} \tag{5}$$

$$h(\vec{x}) = \sum_{i=1}^{l}\alpha_i.\mathrm{k}(\vec{x}_i,\vec{x}) - \rho \tag{6}$$

The classification of an unknown element $\vec{e}$ is determined by its position according to $h(\vec{x})$. If $h(\vec{e}) \ge 0$, then $\vec{e}$ belongs to the concerned class, otherwise $\vec{e}$ does not belong to the concerned class.

2.2 Multi-Objective TRIBES (MO-TRIBES)

TRIBES [7–10] is a parameter-free particle swarm optimization [7,8]. This optimization technique structures the search space in different tribes. Each tribe is composed of a variable number of agents that are completely connected. The improvement of the performances is carried using *structural* and

behavioral adaptations. According to these updates, the best agent of each tribe is designated as the *Chief* or the *Shaman*.

Recently the TRIBES approach has been adapted to the multi-objective optimization that combines k objective functions $f_i(x)$, $i \in [1..k]$. In order to compare the performances of the agents, the MO-TRIBES introduces the dominance in the sense of Pareto, such as: x *dominates* y *if* $\forall i \in [1..k]\ f_i(x) \succcurlyeq f_i(y)$ *and* $\exists i \in [1..k]\ f_i(x) \succ f_i(y)$ [18]. Thus, if we have to minimize f_i, then $f_i(x) \succ f_i(y)$ means $f_i(x) < f_i(y)$. In the case of maximizing f_i, $f_i(x) \succ f_i(y)$ means $f_i(x) > f_i(y)$. Like many multi-objective optimization methods, the non-dominated solutions are stored in an *external archive*. This archive is linked to the Chief of each tribe (Fig. 2).

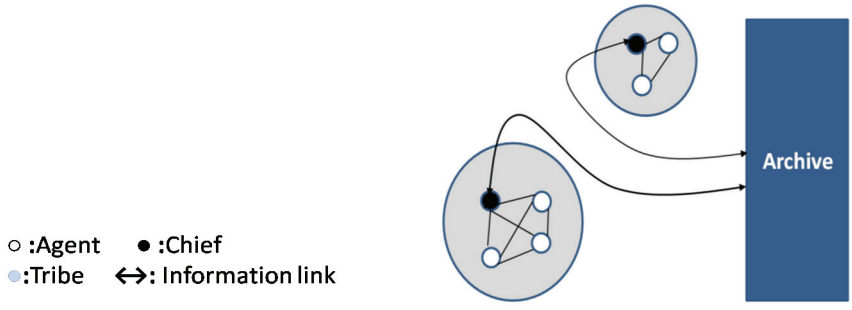


Fig. 2. Swarm's topology in MO-TRIBES: each tribe is composed of agents that are completely connected and linked to the external archive through chiefs.

2.2.1 Structural adaptations

Initially, the search space contains a single agent that represents a tribe. This structure will evolve according to the generation and the removal of agents. In order to allow a delay time for the propagation of information, the structural update is done every $NL/2$ iterations [7]. NL represents the number of interconnections between the agents, such as:

$$NL = \sum_{n=1}^{tribeNb} explorerNb[n]^2 + tribeNb.(tribeNb - 1) \quad (7)$$

With:

- *tribeNb* represents the number of tribes in the search space.
- *explorerNb*[n] represents the number of agents in the tribes n.

The search space is restructured as described below.

- **Removal of an agent:** To reduce the computing time, agents are removed as soon as possible. The removal of an agent cannot affect negatively the performances. Consequently, the worst agent of each *good tribe* (we declare a tribe as "good," if at least one agent of the tribe have improved its best location during the last iteration) is removed [9].

- **Generation of agents:** Each bad tribe (we declare a tribe as "bad," if none of its agents have improved its best location during the last iteration) generates $nb\_agent = max\ (2, (9.5 + 0.124(dim - 1))/tribeNb])\ agents$, dim represents the dimension of the search space. All the generated agents compose a new tribe. As summarized in Table 2, these agents could be confined or free [9].

Table 2. The generation of agents

Type of agent	Functions
Free agent	To diversify the population, the free agent is randomly generated according to a uniform distribution. Such as: $\begin{cases} in\ the\ whole\ search\ space:\ X_{generated-j} = U(x_{minj}, x_{maxj}), j = \{1, \dots, dim\} \\ on\ a\ side\ of\ the\ search\ space:\ X_{generated-j} = \begin{cases} U(x_{minj}, x_{maxj}), j \in I \subset \{1, \dots, dim\} \\ x_{boundj}, \quad j \in J \subset \{1, \dots, dim\} \end{cases} \\ on\ a\ vertex\ of\ the\ search\ space: X_{generated-j} = x_{boundj}, j = \{1, \dots, dim\} \end{cases}$ With $U(x_{minj}, x_{maxj})$ is a real number chosen from the uniform distribution in the interval $[x_{minj}, x_{maxj}]$. x_{boundj} represents x_{minj} or x_{maxj}. Also, I and J are two sub-spaces of $\{1, \dots, dim\}$ that are randomly generated for each new agent.
Confined agent	To intensify search inside an area, the new agent will be generated as following: $\vec{X}_{generated} = alea_{sphere}(\vec{P}_{lbest}, \|\vec{P}_{Xbest}, \vec{P}_{lbest}\|)$ Let us denote $\vec{X}_{best}$is the best particle of the generating tribe. $\vec{I}_{best}$ is the best informer of $\vec{X}_{best}$. $\vec{P}_{Xbest}$ and $\vec{P}_{lbest}$ are the best positions of $\vec{X}_{best}$ and $\vec{I}_{best}$. Thus, $alea_{sphere}(\vec{P}_{lbest}, \|\vec{P}_{Xbest}, \vec{P}_{lbest}\|)$ represents a point chosen randomly with a uniform distribution in a hyper-sphere with the center $\vec{P}_{lbest}$ and radius $\|\vec{P}_{Xbest}, \vec{P}_{lbest}\|$.

2.2.2 Behavioral adaptations

At each iteration, every agent $\vec{x}$ follows a strategy of displacement that combines its personal best position $\overrightarrow{p}$ and its social best position $\vec{g}$. The best social position $\vec{g}$ associated with an agent $\vec{x}$ is determined according to the status of $\vec{x}$. If the agent $\vec{x}$ is a Chief, $\vec{g}$ is randomly chosen from the archive. On otherwise, $\vec{g}$ represents the best position of the Chief.

However, the strategy of displacing an agent is chosen according to a qualitative description of its two last performances. The performances are described as a deterioration (–), an improvement (+), or as status quo (=). There are nine possible variations of history [10]. These variations have been divided into three groups that define the behavior of an agent as bad, average or good. Table 3 summarizes the different strategies of displacement [7,10], as:

$$\lambda(\vec{x}) = (1/k)\sum_{i=1}^{k}(f_i(\vec{x})/max_{j\in[1:k]}(f_j(\vec{x}))) \quad (8)$$

Let us denote by $alea_{sphere}(H_p)$, a point uniformly chosen in the hyper-sphere with center $\vec{p}$ and radius $\|\vec{p}-\vec{g}\|$ and $alea_{sphere}(H_g)$ is a point uniformly chosen in the hyper-sphere with center $\vec{g}$ and radius$\|\vec{p}-\vec{g}\|$. Thus, $alea_{normal}(g_j - x_j, \|g_j - x_j\|)$ represents a point randomly chosen with a Gaussian distribution with the center $g_j - x_j$ and standard deviation $\|g_j - x_j\|$ [10].

Table 3. Strategies of displacement

Strategy of displacement	Gathered statuses	Functions
Local by independent Gaussians	(+ +)(= +)	$x_j = g_j + alea_{normal}(g_j - x_j, \|g_j - x_j\|), j \in \{1, \dots, dim\}$
Disturbed pivot	(+ =)(- +)	$\vec{X} = c_1.alea_{sphere}(H_p) + c_2.alea_{sphere}(H_g)$ $c_1 = \lambda(\vec{p})/(\lambda(\vec{p}) + \lambda(g))$ *and* $c_2 = \lambda(\vec{g})/(\lambda(\vec{p}) + \lambda(g))$
Pivot	(- -)(= -)(+ -) (- =)(= =)	$\vec{X} = (1+b).(c_1.alea_{sphere}(H_p) + c_2.alea_{sphere}(H_g))$ $c_1 = \lambda(\vec{p})/(\lambda(\vec{p}) + \lambda(g))$ *and* $c_2 = \lambda(\vec{g})/(\lambda(\vec{p}) + \lambda(g))$ $b = N(0, ((\lambda(\vec{p}) - \lambda(g))/((\lambda(\vec{p}) + \lambda(g))$

2.2.3 Updating the archive

The maximum size of the external archive *archiveSize* is estimated according to three factors: the number of objective functions *k*, the number of non-dominated solutions found since the last modification of the archive size *nDomPrev*, and the number of swarms that have been reset since the beginning of the *reset* process. Such as [7,8,10]:

Algorithm 1: Archive's size processing

if **reset= 0** *then* **archiveSize** =[e^k]
else **archiveSize** = **archiveSize** + [**10** .**ln** (**1** + **nDomPrev**)]
endif

The diversity of the solutions are maintained using a criterion based on the crowding distance [7,10,19]. The crowding distance of an element *i* of the archive estimates the size of the largest cuboïd enclosing the point *i* without including any other point in the archive [10]. In Fig. 3, the crowding distance of the *i*th solution of the archive (shown by black dots) is the average side length of the cuboïd (shown by the dashed box) [10,19].

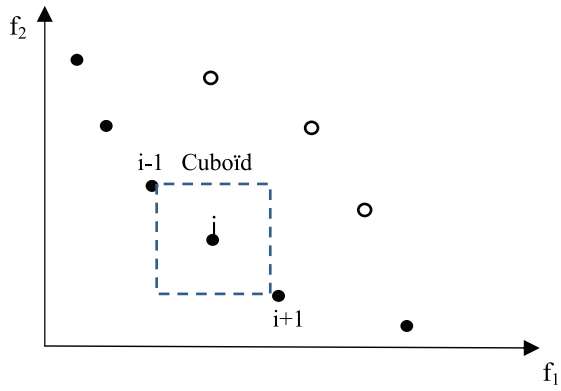


Fig. 3. Crowding distance: the crowding distance of the element *i* represents the size of the largest cuboïd enclosing the point *i* without including any other.

The idea is to maximize the distance between the solutions stored in the archive [7,10]. The archive is updated according to Algorithm 2 [7,8]. Let us denote that *tribe*[*i*].*agent*[*j*] is the agent *j* of the tribe *i* and *nonDomCtr* represents the number of non-dominated solution stored on the archive.

Algorithm 2: Archive's update

```
for i=1 to tribeNb
  for j=1 to tribe[i].explorerNb
    if tribe[i].agent[j] dominates some elements of the archive then
          Delete the dominated elements
    endif
    if tribe[i].agent[j] is non-dominated then
          if archiveSize ≠ nonDomCtr then
           Add tribe[i].agent[j] to the archive.
          else
            Calculate the crowding distances.
            Sort the elements of the archive according to the crowding distance.
            Replace the solution with the lower crowding distance by tribe[i].agent[j .
          endif
    endif
  endfor
endfor
```

3. Proposed Methodology: MO-TRIBES/OC-SVM

To improve the performances of a single classifier, many approaches propose to pool the decisions of multiple classifiers [4,20–23]. Among these methods, BAGGING [5] has been developed to reduce the influence of the selection of the learning set on the performance of classifiers.

BAGGING is based on several classifiers that have been trained on different training data noted $S_{i=1..nbr\_c}$ where, nbr_c represents the number of classifiers to train. This training data must be slightly but sufficiently different from the original one to get different classifiers [24].

For this purpose, the training data is obtained according to a replacement sampling from the original training data noted *'Z'*. The final classification is done by plurality voting [4,23]. Thus, BAGGING follows Algorithm 3 [25].

Algorithm 3: A standard algorithm used for BAGGING

Initialize the parameters:

- *nbr_c, the number of classifiers to train*
- *D = { }, sets of bootstrap samples*

Training phase:

for ***k=1,…,nbr_c***

Take a bootstrap sample S_k form Z

Build a classifier D_k using S_k as the training data.

Add the classifier to the current ensemble, $D=D\cup D_k$

endfor

Return D

Classification phase:

Run D_1, …,D_k on the input x

Select the class with the maximum number of votes as label for x

Based on the BAGGING principle and in order to control its random aspect, previous work has been based on the initial decomposition of the training data, mono-objective optimization, and multi-class classification. Such as [6] :

- The approach begins with one sub-training data and stops when all the sub-training data are injected.
- The objective function is the Recognition Rate (RR), which should be maximized.
- The classification is done according to a vote that combines all the agents.

From this work, we have noticed that the initial decomposition of the training data introduces a new parameter that significantly influences the obtained results. Also, the injection of new sub-training data momentarily deteriorates the RR. To solve this issue, in this paper, we are introducing an automatic generation of agents [7]. These generations allow us to control the random aspect of BAGGING without including a new parameter.

We have also noticed that optimization based on the maximization of the recognition rate generates

new confusions between some classes. In this paper, this problem is solved using the OC-SVM and the MO-TRIBES optimization. OC-SVM allows us to focus on the recognition of a single class. On the other hand, MO-TRIBES performs a multi-objective optimization based on minimizing the two objective functions of the false negatives (FN) rate and the false positive (FP) rate.

Thus, our proposed approach is based on a multi-objective optimization, an automatic generation of agents, and mono-class classification. The MO-TRIBES/OC-SVM approach follows a phase of training and another of classification, as described below and in Fig. 4.

- **Training phase**
  Initially, we began the optimization by using the original training data representing a single agent that composes a tribe. In order to minimize the false negatives and false positives, the different structural and behavioral updates generate new agents. This process is stopped when the number of evaluations of the objective functions exceeds '10000.dim' [7].
- **Classification phase**
  The MO-TRIBES/OC-SVM approach is based on the generation of different bases evaluated according to a multi-objective optimization. So, the final classification will be done according to the non-dominated solution.

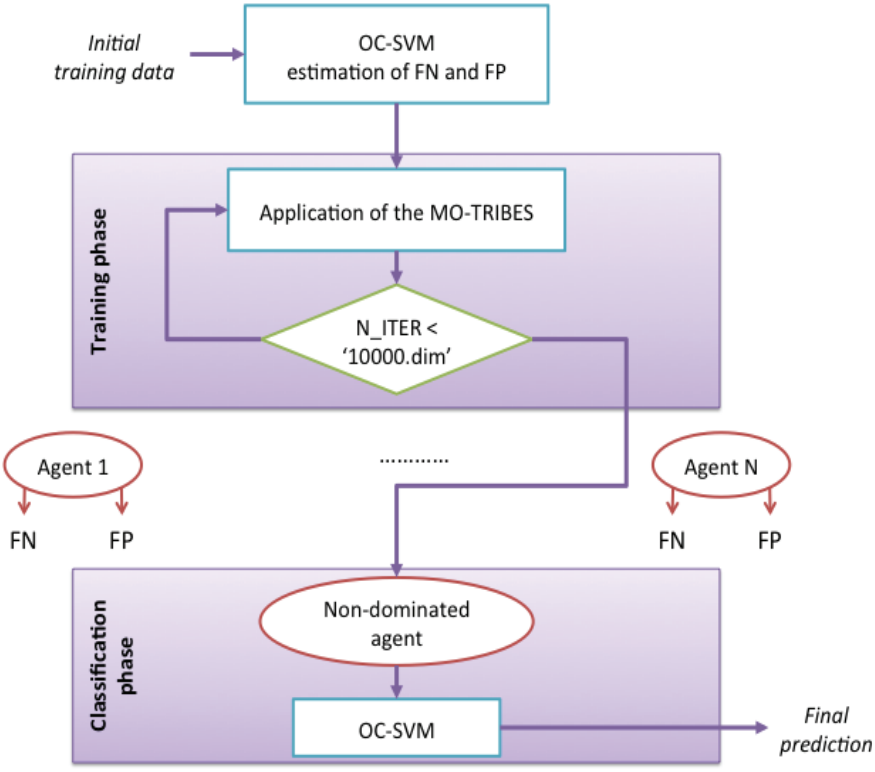


Fig. 4. MO-TRIBES/OC-SVM principle: by considering the initial training data as an agent, the MO-TRIBES makes different behavioral and structural adaptations to give a non-dominated agent that minimizes the false negative and the false positive rates of the OC-SVM.

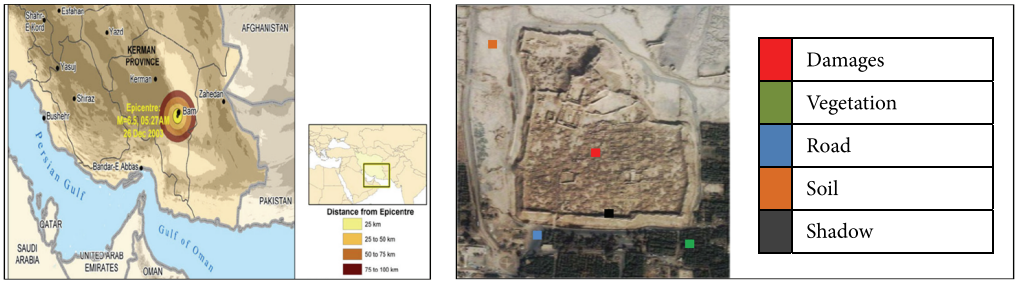
4. Experimental Results

4.1 Data Used

We were interested in applying the MO-TRIBES/OC-SVM approach to the extraction of an area of interest from satellite images. The evaluation of the extraction was done according to the FN rate, the FP rate, and the average error rate $AER = (FN+FP)/2$. For this purpose, we used the data give below.

- **Area 1**

  We applied our proposed approach to extract images of areas that were damaged by the earthquake in Bam, Iran. This earthquake occurred on December 26, 2003 at 5:26:26 (local time) [26,27] and measured at 6.5 on the Richter scale [28,29]. The different tests were done on an IKONOS image acquired on December 27, 2003 (Fig. 5). Thanks to [27,30], we were able to generate a learning base and a testing base that contained 5 classes.

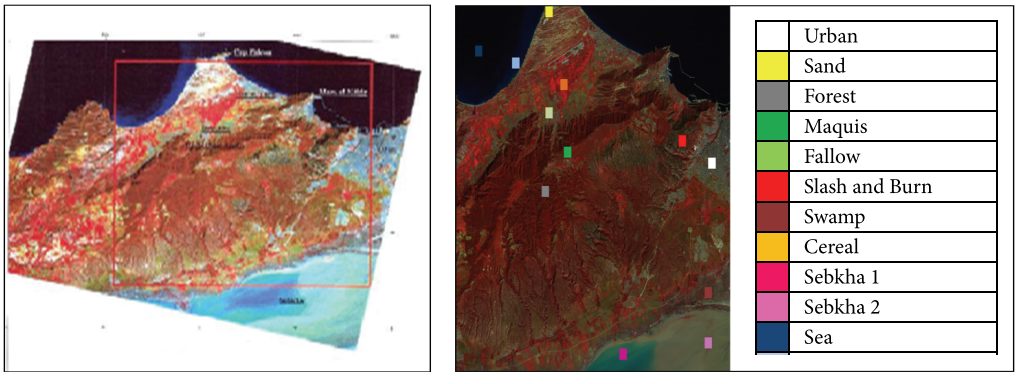


(a) (b)

Fig. 5. Area 1: (a) study area and (b) 5 classes identified, labeled and superimposed in IKONOS image.

- **Area 2**

  We also applied our proposed approach for the identification of urban areas from a satellite image captured by LANDSAT5 TM (TM1 TM3 and TM4). This satellite image represents the western region of Oran, Algeria on March 15, 1993 at 9:45 a.m. As we can see in Fig. 6, this study area was chosen for the diversity of classes (12 classes). The learning and testing bases that were used were based on our thematic knowledge.



(a) (b)

Fig. 6. Area 2: (a) study area and (b) 12 classes identified, labeled and superimposed in LANDSAT image.

Table 4. The obtained results using the OC-SVM

Parameter		Area 1			Area 2		
v	σ	FN (%)	FP (%)	AER (%)	FN (%)	FP (%)	AER (%)
0.1	0.50	17.82	13.53	15.67	11.88	2.97	7.42
	0.75	17.82	13.53	15.67	12.87	3.69	8.28
0.3	0.50	30.69	10.72	20.70	32.67	1.98	17.32
	0.75	30.69	10.72	20.70	34.65	1.80	18.22
0.6	0.50	35.64	7.09	21.36	62.37	0.09	31.23
	0.75	59.40	1.32	30.36	50.49	0.63	25.56

4.2 Results with the OC-SVM and the BAGGING/OC-SVM

The application of the OC-SVM to the extraction of areas of interest from satellite images was done using a Gaussian kernel, which induces a projection in an infinite dimension [17]. The tests were studied according to the variations of ν and σ. The results are summarized in Table 4 and Fig. 7. As can seen in Table 4, the performances of the OC-SVM are linked to the initial parameterization (σ and ν) [31]. Such as [16]:

- When σ increases, the Gaussian kernel grows. Consequently, the decision function shape becomes similar to a sphere that contains $(1-\nu)*l$ training elements. This increase may generate false positives.
- The increase of ν results in higher error rates. Consequently, false negatives increased.

Fig. 7 shows the images obtained with the best parameterizations of OC-SVM. These best parameterizations were determined according to the AER (the best AER was obtained with σ=0.5 and ν=0.1). The images obtained show some false positives (Fig. 7). In Area 1, false positives are situated in the classes of roads and soil. However, in Area 2, the false positives are located in the classes of fallow, Sebkha, swamps, slash and burn. As we can see on the figures below, the false negatives are lower.

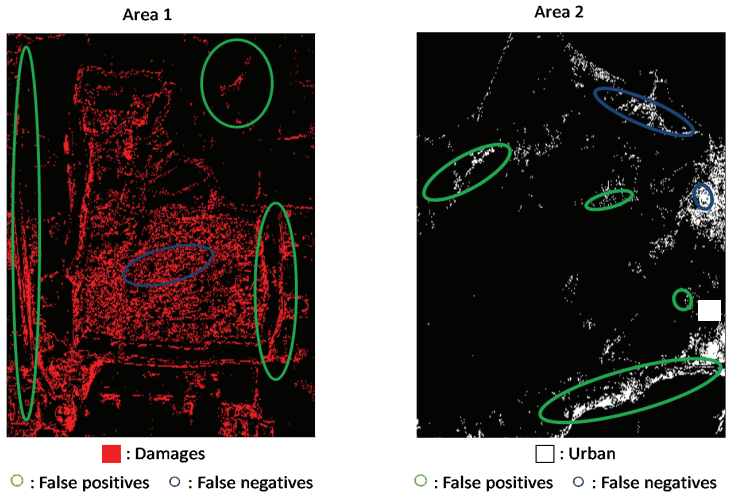


Fig. 7. Images obtained with the OC-SVM. In the 'Area 1' and the 'Area 2' we use the parameterization σ=0.5 and ν=0.1 that gives the best 'AER'.

In order to improve the performances of the OC-SVM, we applied BAGGING/OC-SVM according to the best parameterizations of the OC-SVM. The results obtained are summarized in Table 5. The BAGGING/OC-SVM approach can improve the performances of OC-SVM. However, we have also noticed a deterioration of some results (for example: Area 1, Test 1, the number of sets on each bootstrap=30, nbr_c=9) and a sensitivity due to the random aspect (for example, the number of sets on each bootstrap =30, nbr_c=9).

Table 5. The results obtained using the BAGGING/OC-SVM

	Parameter		Area 1			Area 2		
	Number of sets on each bootstrap	nbr_c	FN (%)	FP (%)	AER (%)	FN (%)	FP (%)	AER (%)
Test 1	30	5	8.91	13.53	11.22	14.85	2.70	8.77
		7	17.82	13.53	15.67	13.86	2.97	8.41
		9	22.77	13.53	18.15	10.89	2.97	6.93
	70	5	17.82	13.53	15.67	10.89	2.97	6.93
		7	17.82	13.53	15.67	11.88	2.97	7.42
		9	17.82	13.53	15.67	10.89	2.97	6.93
Test 2	30	5	17.82	13.53	15.67	13.86	2.88	8.37
		7	17.82	13.53	15.67	15.84	2.97	9.40
		9	17.82	13.53	15.67	13.86	2.88	8.37
	70	5	17.82	13.53	15.67	10.89	3.06	6.97
		7	17.82	13.53	15.67	10.89	2.97	6.93
		9	17.82	13.53	15.67	10.89	2.97	6.93

Fig. 8 shows the images obtained with the best parameterizations of BAGGING/OC-SVM. In Area 1, we used Test 1 with number of sets on each bootstrap=30 and nbr_bootstrap=5. In Area 2, we used Test 1 with the number of sets on each bootstrap=30 and nbr_bootstrap=9. The images obtained show a slight improvement in Area 2 and an overestimation of damages in Area 1 (see the zoom in on both images).

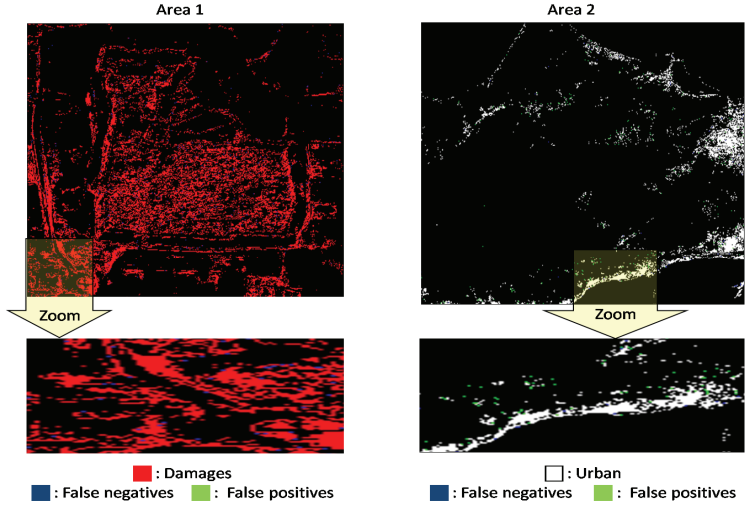


Fig. 8. Images obtained with the BAGGING/OC-SVM. In the 'Area 1' the image is obtained with the Test 1, number of sets on each bootstrap=30 and nbr_bootstrap=5. In the 'Area 2' the image is obtained with the Test 1, number of sets on each bootstrap=30 and nbr_bootstrap=9.

4.3 Results with the MO-TRIBES/OC-SVM

The evaluation of the MO-TRIBES/OC-SVM approach was done according to the best OC-SVM parametrization. As it can be noticed in Fig. 9, MO-TRIBES/OC-SVM learning organizes the search space in different tribes that are composed of a different number of agents.

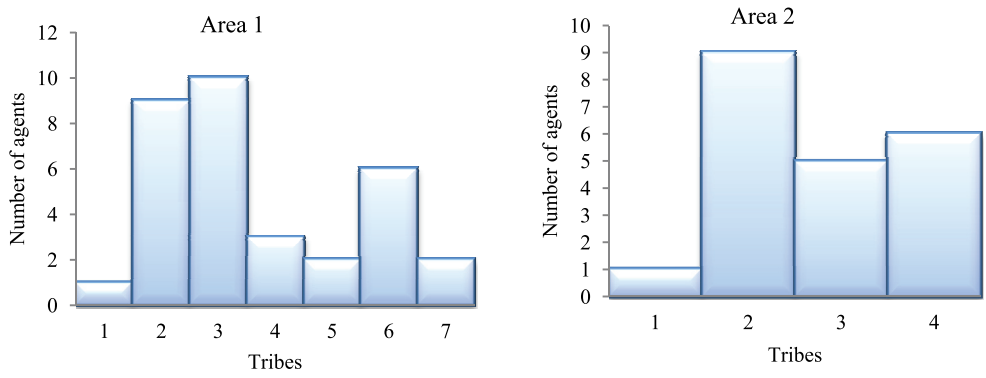


Fig. 9. Swarm's structure: number of agents in each tribe generated.

We see from Fig. 10 that the approach improves the performances of OC-SVM. We also noticed that the improvement of false negatives was more significant in Area 2. On the other hand, the stabilization of the objective functions was faster in Area 1. However, the two areas required less than 30,000 evaluations to stabilize their objective functions. So, we were able to stop the process at 20,000 evaluations. At 20,000 evaluations, Areas 1 and 2 stabilized their objective functions.

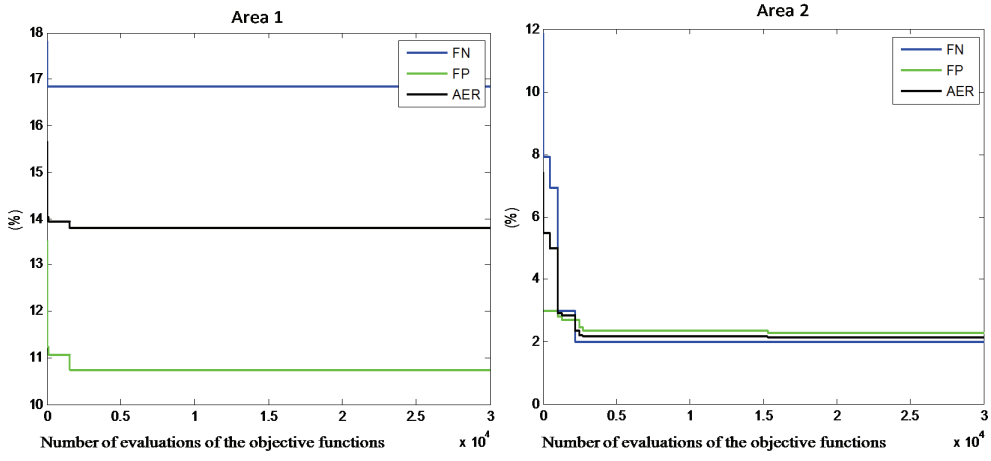


Fig. 10. Evolution of the objective functions (FN, FP, and AER).

To confirm these results, we evaluated the MO-TRIBES/OC-SVM approach according to testing bases. Table 6 summarizes the obtained results of the MO-TRIBES/OC-SVM approach, the BAGGING/OC-SVM, and the OC-SVM. Thanks to the optimization of multi-objectives, our proposed approach was able to improve the AER without generating more false positives and false negatives.

Table 6. The results obtained using the OC-SVM, BAGGINIG/OC-SVM and the MO-TRIBES/OC-SVM approach (learning and testing base)

		Learning base			Testing base		
		FN (%)	FP (%)	AER (%)	FN (%)	FP (%)	AER (%)
Area 1	OC-SVM	17.82	13.53	15.67	21.89	41.79	31.84
	BAGGING/OC-SVM	8.91	13.53	11.22	41.79	22.30	32.04
	MO-TRIBES/OC-SVM	16.83	10.72	13.77	19.73	38.30	29.01
Area 2	OC-SVM	11.88	2.97	7.42	1.99	3.50	2.74
	BAGGING/OC-SVM	10.89	2.97	6.93	1.99	3.41	2.70
	MO-TRIBES/OC-SVM	1.98	2.25	2.11	1.99	1.19	1.59

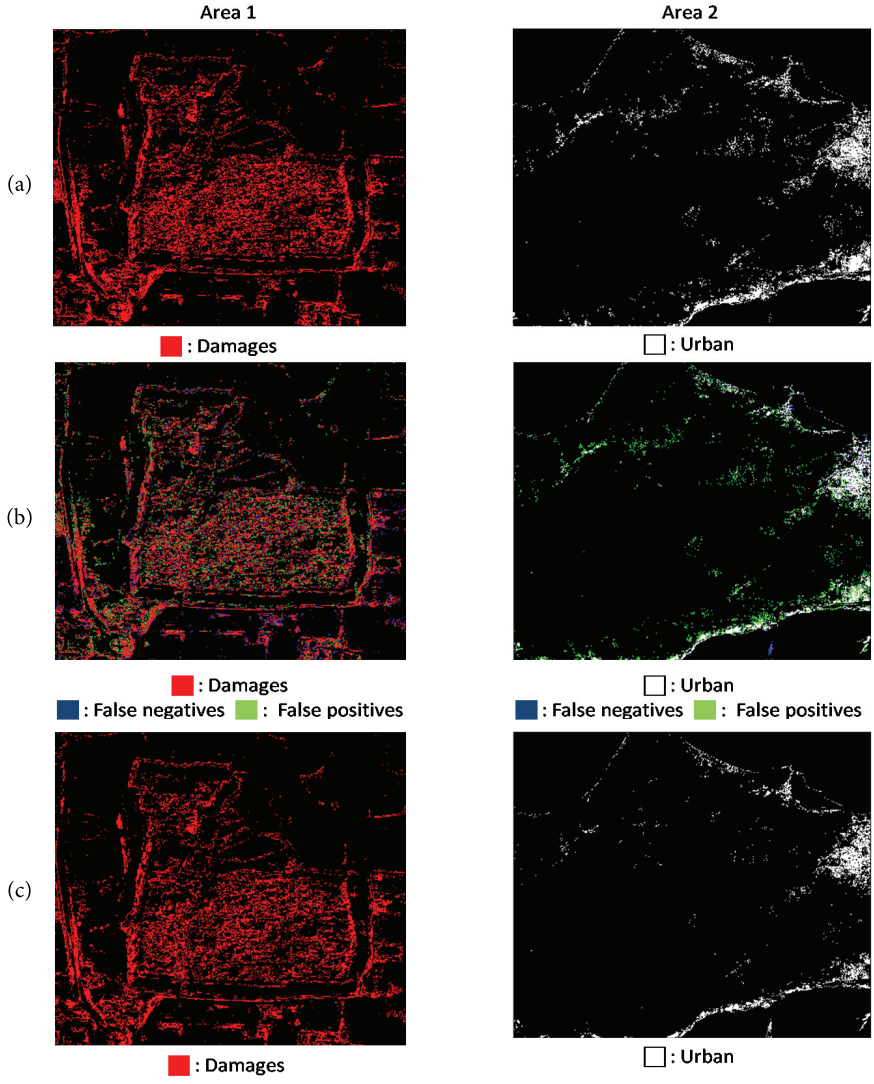


Fig. 11. Images obtained with the MO-TRIBES/OC-SVM approach. (a) Images obtained using the OC-SVM. (b) Detection of errors using the MO-TRIBES/OC-SVM approach. (c) Final images obtained using the MO-TRIBES/OC-SVM approach.

The classification of images was done according to a non-dominated agent (Fig. 11). Fig. 11(a) represents the images obtained using the OC-SVM and Fig. 11(b) highlights the detection of errors with the MO-TRIBES/OC-SVM approach. Fig. 11(c) also shows the extraction of areas of interest obtained with the MO-TRIBES/OC-SVM approach.

As can be observed, the images obtained confirm the improvement of results. In Area 2, the improvement of the false positives (represented in green in Fig. 11(b)) is more significant than the improvement of false negatives (represented in blue in Fig. 11(b)). In Area 1, the false negatives and false positives were detected more or less identically. However, some false positives persisted. In Area 1, the false positives are situated in the classes of roads and soil. The false positives in Area 2 are located in the classes of fallow, Sebkha, and swamps.

4.4 Comparison with Other Classifiers

In order to compare the MO-TRIBES/OC-SVM approach to other classifications methods, we applied the same data to the following different algorithms: decision trees [11], random forest [12], naïve Bayes [13], and Quadratic Discriminant Analysis [14].

Table 7 summarizes the evaluation of these algorithms according to testing bases. Trained on the same learning data and evaluated on the same testing bases, the MO-TRIBES/OC-SVM approach presents the best AER. These results confirm the effectiveness of our proposed approach.

Table 7. Comparative studies

	Methods	FN (%)	FP (%)	AER (%)
Area 1	Decision trees (maximum depth of the tree=3)	58.21	0.00	29.10
	Random forest (maximum depth of the tree=3, number of trees in the forest=10)	50.25	13.43	31.84
	Naive Bayes	37.31	20.81	29.06
	QDA	59.70	0.00	29.85
	MO-TRIBES/OC-SVM	19.73	38.30	29.01
Area 2	Decision trees (maximum depth of the tree=3)	16.37	0.11	8.24
	Random forest (maximum depth of the tree=3, number of trees in the forest=10)	4.39	0.04	2.21
	Naive Bayes	100.0	0.00	50.00
	QDA	100.0	0.00	50.00
	MO-TRIBES/OC-SVM	1.99	1.19	1.59

5. Conclusions

We have introduced the OC-SVM classifier, a one-class classification model for identifying areas of interest without including other classes of the scene. The performances of the OC-SVM classifier were improved by using BAGGING, which is based on the generation of several classifiers trained on bases that are slightly but sufficiently different enough from the original training base.

We have also proposed enhancing the BAGGING technique with the MO-TRIBES optimization algorithm. It generates and manages different training bases according to multi-objective optimization. Our approach is based on the minimization of the false negative and the false positive rates of the OC-SVM, and it ensures an optimal choice of training data and control of the random aspect of BAGGING.

We applied the MO-TRIBES/OC-SVM approach to the extraction of areas of interest from satellite images. The results suggest that our method outperforms both OC-SVM and BAGGING/OC-SVM. It also improved the AER without generating new errors thanks to the optimization of multi-objectives. Its efficiency and the robustness were confirmed via a comparison to different classification methods.

References

[1] J. Munoz-Mari, F. Bovolo, L. Gomez-Chova, L. Bruzzone, and G. Camp-Valls, "Semisupervised one-class support vector machines for classification of remote sensing data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 8, pp. 3188–3197, 2010.

[2] B. Scholkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[4] J. Li, "Bagging and boosting: brief introduction," Department of Statistics, The Pennsylvania State University [Online]. Available: http://sites.stat.psu.edu/~jiali/course/stat597e/notes2/bagging.pdf.

[5] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases*. Heidelberg: Springer, 2010, pp. 135–150.

[6] H. Fizazi and W. Benhabib, "Approche tribale des SVMs pour la classification des images satellitaires," *Nature & Technologie*, no. 6, pp. 9–15, 2012.

[7] Y. Cooren, "Perfectionnement d'un algorithme adaptatif d'Optimisation par Essaim Particulaire: application en génie médical et en électronique," Ph.D. dissertation, Paris-East University, 2008.

[8] Y. Cooren, P. Siarry, and M. Fakhfakh, "Application of MO-TRIBES to the design of analog electronic circuits," in *Proceedings of 16th IEEE International Conference on Electronics, Circuits, and Systems*, Yasmine Hammamet, Tunisia, 2009, pp. 263–266.

[9] Y. Cooren, M. Clerc, and P. Siarry, "Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm," *Swarm Intelligence*, vol. 3, no. 2, pp. 149–178, 2009.

[10] Y. Cooren, M. Clerc, and P. Siarry, "MO-TRIBES, an adaptive multiobjective particle swarm optimization algorithm," *Computational Optimization and Applications*, vol. 49, no. 2, pp. 379–400, 2011.

[11] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[12] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[13] T. Fomby, "Naive Bayes classifier," Department of Economics, Southern Methodist University, Dallas, TX, 2008.

[14] W. Williams, "Classification using linear discriminant analysis and quadratic discriminant analysis," 2009 [Online]. Available: http://www.cs.colostate.edu/~anderson/cs545/assignments/solutionsGoodExamples/assignment3Williams.pdf.

[15] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[16] O. Zammit, "Détection de zones brûlées après un feu de forêt à partir d'une seule image satellitaire SPOT 5 par techniques SVM," Ph.D. dissertation, Nice Sophia Antipolis University, 2008.

[17] M. Geist, O. Pietquin, and G. Fricout, "Astuce du noyau & quantification vectorielle," in *Proceedings of Actes du 17ème colloque sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10)*, Nice, France, 2010, pp. 1-8.

[18] M. Schoenauer, "Optimisation évolutionnaire multi-objectif," 2007 [Online]. Available: http://www-lisic.univ-littoral.fr/~fonlupt/yravals2/yravals-MO.pdf.

[19] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature*. Heidelberg: Springer, 2000, pp. 849–858.

[20] J. Laumonier, "Méthode d'apprentissage de la coordination multiagent : application au transport intelligent," Ph.D. dissertation, Laval University, Quebec, Canada, 2008.

[21] A. L. Bazzan and F. Klugl, *Multi-Agent Systems for Traffic and Transportation Engineering*. Hershey, PA: IGI Global, 2009.

[22] A. Saidane, H. Akdag, and I. Truck, "Une approche SMA de l'Agrégation et de la Coopération des Classifieurs," in *Proceedings of 3th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT'2005)*, Sousse, Tunisia, 2005, pp. 1–8.

[23] D. Ghimire and J. Lee, "Extreme learning machine ensemble using bagging for facial expression recognition," *Journal of Information Processing Systems*, vol. 10, no. 3, pp. 443–458, 2014.

[24] A. Lechervy, "Bagging et boosting," University of Caen, France, 2014.

[25] A. Srinet and D. Snyder, "Bagging and boosting," [Online]. Available: https://www.cs.rit.edu/~rlaz/prec20092/slides/Bagging_and_Boosting.pdf.

[26] ReliefWeb, "Iran: satellite map of Bam City (Kerman Province) - northern part," 2003 [Online]. Available: http://reliefweb.int/map/iran-islamic-republic/iran-satellite-map-bam-city-kerman-province-northern-part.

[27] Y. Yano and F. Yamazaki, "Building damage detection of the 2003 Bam, Iran earthquake using Quickbird images based on object-based classification," in *Proceedings of the 27th Asian Conference on Remote Sensing*, Ulaanbaatar, Mongolia, 2006, pp. 1-6.

[28] S. Eshghi and M. Zare, "IIEES: Bam (SE Iran) earthquake of 26 Dec 2003, Mw6.5: a preliminary reconnaissance report," 2003 [Online]. Available: http://reliefweb.int/report/iran-islamic-republic/iiees-bam-se-iran-earthquake-26-dec-2003-mw65-preliminary.

[29] United States Geological Survey, "Magnitutde 6.6 - Southeastern Iran," 2003 [Online]. Available: http://earthquake.usgs.gov/earthquakes/eqinthenews/2003/uscvad/.

[30] SERTIT, "Earthquake in Iran," 2004 [Online]. Available: http://sertit.u-strasbg.fr/SITE_RMS/Archives/16_iran_2003/iran_2003.html.

[31] H. Fizazi and W. Benhabib, "Contribution of the OC-SVM to the extraction of earthquake damages," in *Proceedings of 1st International Conference on Information Systems and Technologies (ICIST2011)*, Tebessa, Algeria, 2011.

Wafaa Benhabib

She received her Engineering degree from the University of Mohamed Boudiaf in 2008 and Her M.S. degree in remote sensing, analyze and automatic processing of remotely sensed data from the University of Mohamed Boudiaf, in 2010. Since 2011, she is with the University of Mohamed Boudiaf as a PhD candidate. Her research interests include: image processing, classification and clustering algorithms applied to remotely sensed data.

Hadria Fizazi

She received her Engineering degree in Engineering from the University of Mohamed Boudiaf in 1981, the "Docteur Ingénieur" industrial computer science automation from the University of Lille 1 in France, in 1987 and the Ph.D. degree in computer science from the University of Mohamed Boudiaf in 2005. Currently she is a professor in the Faculty of Mathematics and Computer Science in the Mohamed Boudiaf. Her research interests cover image processing and the application of pattern recognition to remotely sensed data.