Oil Spill Classification from Multi-Spectral Satellite Images: Exploring Different Machine Learning Techniques

Linda Corucci*, Fabio Nardelli, Marco Cococcioni
a Dipartimento di Ingegneria dell’Informazione: Informatica, Elettronica, Telecomunicazioni, largo Lucio Lazzarino, 1, 56122 Pisa, Italy; 
b Flyby S.r.l., Via Puini, 97, 57128 Livorno, Italy

ABSTRACT

This work describes the potential of oil spill classification from optical satellite images, as investigated by applying different machine learning techniques to a dataset of more than 300 oil spill candidates, which have been detected from multi-spectral satellite sensors during the years 2008 and 2009, over the entire area of the Mediterranean Sea. A set of geometrical and grey level features from Synthetic Aperture Radar (SAR) literature has been extracted from the regions of interest in order to characterize possible oil spills and feed the classification system. Results obtained by applying different machine learning classifiers to the dataset, and the achieved performance are discussed. In particular, as a first approach to oil spill classification, simple statistical classifiers and neural networks were used. Then, a more interpretable fuzzy rule-based classifier was employed, and performance evaluation was refined by exploiting Receiver Operating Characteristic (ROC) analysis. Finally, since oil spill dataset collection happens incrementally, a suitable technique for online classification was proposed, encompassing at the same time cost-oriented classification, in order to allow for a dynamic change of the misclassification costs. This latter goal has been achieved by building an ensemble of cost-oriented, incremental and decremental support vector machines, exploiting the concept of the ROC convex hull.

Keywords: oil spill, multi-spectral satellite images, statistical and neuro-fuzzy classifiers, support vector machines.

1. INTRODUCTION

The presence of oil spills at sea represents a relevant issue since they are causing serious damages to marine and coastal ecosystem. The detection of oil spills using satellite images represents an economical and easy way for monitoring large areas, thus offering many advantages in economical and time saving terms. SAR images have been widely used for oil spill detection, as they are not affected by local weather conditions and cloudiness. In particular, many algorithms for semi-automatic or automatic oil spill detection in SAR images, such as statistical modeling combined with a rule base approach [1], neural network-based classifiers [2] [3], fuzzy logic systems [4], have already been developed and presented in the literature. Unfortunately, radar backscatter values for oil spills are similar to backscatter values for very calm sea areas because the presence of an oil spill dampens capillary and short gravity waves. This causes a high number of false alarms.

Currently, many government institutions already use SAR technology-based services for oil spill detection, but the high false alarm rate makes these systems not enough reliable so that many detected spills are followed by no action, due to the risk of expensive in situ missions which could turn out to be false alarms.

On the other hand, the detection of oil spills using optical satellite images allows for large areas monitoring and remote zones control, providing more frequent (sometimes daily) information if compared to SAR images. The development of a new approach based on optical data could be used either on its own or as a support to SAR-based solutions, in order to meet the need of environmental protection authorities for efficient and cost effective monitoring tools. Moreover, the possibility of detecting oil spills by optical satellite sensors has already been demonstrated [5].

We have to consider, however, that oil spill detection using optical images appears to be quite problematic in comparison with SAR oil spill detection, because good weather conditions and day light are mandatory conditions to perform a correct detection. A number of other factors make the optical oil spill detection a difficult task, such as the presence of...
clouds, the known difficulty in performing an accurate atmosphere correction, the presence of sunglint, and, as investigated in [6] [7], the change of the contrast of oil spill with respect to sea water depending on both the oil type and the water type.

In this paper we first tackle oil spill detection by applying many different classical statistical classifiers and neuro-fuzzy classifiers to a significant dataset of oil spills and natural phenomena identified in multi-spectral satellite images. We then propose a new approach by building an online classifier based on an ensemble of cost-oriented Support Vector Machines (SVMs), and we adopt the ROC convex hull [8] as a method for the evaluation of classifier performance on the dataset.

The paper is organized as follows: in Section 2 we present a software architecture for satellite image processing and supervised classification. In Section 3 we describe the collected dataset and the features used as input to the classification system. In Section 4 we show some results obtained by using different statistical classifiers and neural networks, while in Section 5 we describe an approach based on a neuro-fuzzy classifier, together with performance evaluation in the ROC space. In Section 6 we introduce cost oriented classification for the oil spill classification problem, and we explain how to perform such a classification in the ROC space. In Section 7 we describe the proposed online cost-oriented incremental/decremental formulation of SVMs (COID-SVMs). In Section 8 we describe the implemented ensemble of COID-SVMs, used to perform the desired classification in the ROC space, and we show the experimental results. In Section 9 we draw some conclusions.

2. A SOFTWARE ARCHITECTURE FOR OPTICAL OIL SPILL DETECTION

We have developed a software for image processing and oil spill detection as made of the components shown in the scheme in Figure 1. The input of the system is a MODIS multi-spectral image at level L1B. In the preprocessing phase the image is georeferenced, a land mask is applied and atmospheric effects are softened through the atmospheric correction module. The contrast is then enhanced using a local histogram equalization. Regions of interest are then identified by a clustering algorithm (k-means, fuzzy c-means, …) and a set of features describing the selected region are calculated by the feature extraction module. A classification algorithm is then applied in order to discriminate between oil spills and other phenomena. The output of the system is the probability of the selected object to be an oil spill, as estimated by the classification procedure.

This paper focuses on feature extraction and classification modules only.

![Figure 1. Scheme of the oil spill detection system.](image)

3. THE DATASET

We have built a dataset of regions of interest by collecting a number of optical images taken during the years 2008 and 2009 over the entire area of the Mediterranean Sea. We used MODIS-TERRA and MODIS-AQUA L1B data at 250 m spatial resolution. Only bands B1 and B2 [visible (VIS) at 0.65 µm, and near-infrared (NIR) at 0.85 µm] are available from MODIS at this resolution.

We selected those images where clear sky conditions occurred, and then possible oil slicks were detected by visual inspection performed by a trained interpreter, basing on his experience. For each selected image the corresponding Envisat-ASAR image at 150 m spatial resolution has been analyzed for comparison. This has been done in order to increase the reliability of those regions of interest that have been identified as oil spills in both optical and SAR images. Unfortunately, due to sunglint and weather conditions, only approximately 10% of the MODIS images could be used for comparison. However, for many of the selected cases ground truth was available.

After georeferencing MODIS images, we applied a local contrast enhancement based on histogram equalization in order to improve the visualization. Then, regions of interest have been selected through a fully manual procedure. The selected regions have been divided into two classes: the first class is composed by regions identified by photointerpretation as possible oil spills on the basis of the contrast between the regions and the surrounding areas (which are supposed to be
clean waters). The second class is composed by regions identified as look-alikes, which could be algal blooms, currents, etc. The analysis has been conducted examining the scene where the candidate oil spill or look-alike had been detected, that is, considering the context, the location and the possible presence of other elements in the surrounding area. For instance, this allows to distinguish between linear slicks, which might be caused by a moving ship releasing oil, as shown in Figure 2 a), and sea currents, usually occurring in particular spatial patterns, such as those shown in Figure 2 b).

![Figure 2. a) An oil spill case from the dataset. b) A look-alike case from the dataset.](image)

The number of oil slicks detected is approximately equal to the number of look-alikes for a total of 316 elements. We then performed a search for multivariate outliers, basing on the distribution of some features characterizing oil spills, which will be described in the following section. After this analysis we removed 12 elements, obtaining a dataset composed of 304 elements, 157 oil spills and 147 look-alikes. These outliers resulted to be represented by big non-linear shaped slicks and dark regions in a particularly complex background, which we decided not to handle. Actually, the proposed classifier is intended for the identification of illegal oil discharge by moving ships or of oil released during tank cleaning in the sea. These types of slicks have to be well represented in the dataset. Regarding very big slicks, produced for instance by oil tanker accidents, which can cause environmental disasters to happen, since the appropriate authorities are usually informed about these events, we decided not to treat such cases.

3.1 Feature description

In order to discriminate between oil spills and look-alikes, we exploited a number of physical and geometrical features characterizing the object to be classified. These features are computed by the feature extraction module and are used as input for the classification system.

Following the results of SAR oil spill detection [9] [1] [2], we used a set of gray level features, characterizing the differences between the object and the surrounding area, and a set of geometrical features, describing shape and extension:

1) Geometrical features
   - Area of the object (A) expressed in km².
   - Perimeter (P) expressed in km.
   - Complexity (C): defined as $C = P / 2 \sqrt{\pi A}$. This feature generally assumes small numerical values for regions with simple geometry and larger values for regions with complex geometry.
   - Spreading (S): computed performing a principal component analysis on the vector whose components are the coordinates of the pixels belonging to the object:
     $$ S = 100 \lambda_2 / (\lambda_1 + \lambda_2) $$
     where $\lambda_1$ and $\lambda_2$ are the two eigenvalues associated with the covariance matrix ($\lambda_1 > \lambda_2$). This feature assumes low values for long and thin objects and high values for objects closer to a circular shape.

2) Gray level features
- **Object Standard Deviation**: standard deviation of the intensity values of pixels belonging to the object.
- **Max Contrast**: difference between the background mean intensity value and the lowest intensity value inside the object.
- **Mean Contrast**: difference between the background mean intensity value and the object mean intensity value.

We did not make use of the spectral features since the two bands available from MODIS full resolution images are insufficient for a spectral analysis. Gray level features are indeed calculated on a two-band ratio (band 2/band 1) so as not to consider the atmosphere contribution.

In Table 1 and Table 2 we report some statistical parameters of the above mentioned set of features, computed for the 157 oil spill and 147 look-alike cases in the dataset where outliers had been removed. The tables show that oil spills are generally smaller and have a thinner shape.

### Table 1. Statistical parameters of the features calculated for oil spill cases in the dataset.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (km²)</td>
<td>0.688</td>
<td>41.7</td>
<td>6.96</td>
<td>6.58</td>
</tr>
<tr>
<td>Perimeter (km)</td>
<td>2.91</td>
<td>94.6</td>
<td>20.4</td>
<td>15.9</td>
</tr>
<tr>
<td>Complexity</td>
<td>0.229</td>
<td>5.40</td>
<td>2.16</td>
<td>0.770</td>
</tr>
<tr>
<td>Std. Dev. band ratio</td>
<td>0.00610</td>
<td>0.0519</td>
<td>0.0199</td>
<td>0.00940</td>
</tr>
<tr>
<td>Mean Contrast band ratio</td>
<td>-0.0588</td>
<td>0.156</td>
<td>0.0505</td>
<td>0.0311</td>
</tr>
<tr>
<td>Max Contrast band ratio</td>
<td>-0.0255</td>
<td>0.216</td>
<td>0.0933</td>
<td>0.0449</td>
</tr>
<tr>
<td>Spreading</td>
<td>0.114</td>
<td>26.8</td>
<td>4.25</td>
<td>5.35</td>
</tr>
</tbody>
</table>

### Table 2. Statistical parameters of the features calculated for look-alike cases in the dataset.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area (km²)</td>
<td>0.875</td>
<td>73.9</td>
<td>14.4</td>
<td>14.7</td>
</tr>
<tr>
<td>Perimeter (km)</td>
<td>1.00</td>
<td>147</td>
<td>26.4</td>
<td>21.2</td>
</tr>
<tr>
<td>Complexity</td>
<td>0.0912</td>
<td>5.009</td>
<td>2.016</td>
<td>0.893</td>
</tr>
<tr>
<td>Std. Dev. band ratio</td>
<td>0.00521</td>
<td>0.0999</td>
<td>0.0283</td>
<td>0.0179</td>
</tr>
<tr>
<td>Mean Contrast band ratio</td>
<td>-0.00535</td>
<td>0.234</td>
<td>0.0733</td>
<td>0.0519</td>
</tr>
<tr>
<td>Max Contrast band ratio</td>
<td>0.0198</td>
<td>0.451</td>
<td>0.134</td>
<td>0.0822</td>
</tr>
<tr>
<td>Spreading</td>
<td>0.548</td>
<td>44.3</td>
<td>11.9</td>
<td>10.0</td>
</tr>
</tbody>
</table>

### 4. BATCH CLASSIFICATION APPROACH: FIRST ATTEMPT

As a first attempt to oil spill classification, we applied a set of statistical classifiers and neural networks to the extracted features. In particular, we adopted a batch approach, that is, we considered the overall dataset. These classifiers have been trained using 60% randomly chosen examples of the available dataset, while the remaining 40% were used as a test set in order to evaluate the classification performance of the different methods. We employed normal density-based linear and quadratic classifiers, as well as a logistic regression-based linear classifier. As a priori probability we used a value of 0.5 for both oil spill and look alike classes. Concerning non linear classification, we applied three k-nearest neighbor classifiers and two neural networks, namely a radial basis function (RBF) and a multi layer perceptron (MLP). The MLP neural network was trained by a backpropagation algorithm and different topologies were used (1 hidden layer and different numbers of hidden neurons).

In Table 3 we report classifier performances in terms of mean percentage of correct classification calculated on the training set and on the test set. The relative error is estimated by considering deviations from the mean value. Among the examined classifiers the best performance on the test set is obtained by the MLP neural network with 1 hidden layer composed by 6 neurons.
Table 3. Classifier performances: the estimated relative error is 2%

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Mean training set correct classification %</th>
<th>Mean test set correct classification %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal density based linear</td>
<td>79</td>
<td>74</td>
</tr>
<tr>
<td>Normal density based quadratic</td>
<td>77</td>
<td>75</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>80</td>
<td>74</td>
</tr>
<tr>
<td>1-nearest neighbor</td>
<td>100</td>
<td>71</td>
</tr>
<tr>
<td>2-nearest neighbor</td>
<td>92</td>
<td>75</td>
</tr>
<tr>
<td>3-nearest neighbor</td>
<td>88</td>
<td>76</td>
</tr>
<tr>
<td>RBF (10 neurons)</td>
<td>76</td>
<td>65</td>
</tr>
<tr>
<td>MLP (6 hidden neurons)</td>
<td>92</td>
<td>79</td>
</tr>
</tbody>
</table>

As regards misclassified examples, false negatives (i.e. oil spills erroneously classified as look-alikes) are usually characterized by high values for the spreading or small values for the area, while false positives (i.e. look-alikes erroneously classified as oil spills) are represented by a more heterogeneous set.

In a previous work we had used a smaller and earlier version of the presented dataset for the same batch classification approach [10].

5. BATCH CLASSIFICATION APPROACH: SECOND ATTEMPT

We proceeded with the batch classification approach by investigating other more interpretable machine learning techniques, and by refining the performance evaluation procedure. In particular, among the existing neuro-fuzzy techniques, we adopted an ANFIS [11] (Adaptive Network-based Fuzzy Inference System), which will be described in the following section. As regards performance evaluation, we resorted to ROC analysis, which allows to overcome the limitations of the classification accuracy approach in a two class problem, as explained in Section 5.2.

5.1 The model

ANFIS is an adaptive network which is functionally equivalent to a fuzzy inference system. Such networks are always composed of five layers, each one has a specific function. In Figure 3 we consider a Takagi-Sugeno fuzzy inference system [12] and the equivalent ANFIS. The first layer performs the fuzzification of the inputs and is composed of adaptive nodes (i.e. parametric nodes), whose parameters are the premise parameters: each input variable is connected to a number of nodes equal to the number of fuzzy sets used to model the variable, and each node corresponds to a membership function. Thus, the outputs of the first layer are the membership degrees. The second and the third layers are both composed of fixed nodes (i.e. non parametric nodes), each one corresponding to a single rule in the fuzzy inference system. The output of each node belonging to the second layer represents the firing strength of a rule, while the output of each node belonging to the third layer is the normalized firing strength, that is the firing strength divided by the sum of all firing strengths. The fourth layer performs the inference step. It is composed of adaptive nodes, whose parameters are the consequent parameters. Each node computes the consequent part of a rule, multiplied by its normalized firing strength. The last layer, composed of a single fixed node, performs the defuzzification step, computing the sum of the outputs of the fourth layer nodes. Thus the overall output is obtained.

The parameters involved in the node functions are updated in a supervised learning procedure based on the gradient descent method.

The advantages of using ANFIS are manifold: i) the supervised learning procedure of an ANFIS adapts the membership functions to the input variable characteristics, which are thus derived from the dataset, while traditional fuzzy inference systems need a rule structure and membership functions which are predetermined by the user, ii) interpretability is preserved iii) ANFIS can be regarded as a grey-box model since it allows to describe the problem starting from the data, without going into details but keeping interpretability into account, iv) only few parameters must be provided.
The reader interested in recent advances on identification of Takagi-Sugeno systems and ANFIS networks can refer to [13] [14].

### 5.2 ROC analysis

Performance evaluation based on classification accuracy can be inappropriate because it assumes equal error costs for misclassification of target class elements. However, for many applications, such as oil spill detection, this condition is not satisfied. For instance, false positives should be avoided as much as possible since these could bring environmental protection authorities to an unnecessary (and expensive) action. Thus, in order to account for this, we resort to Receiver Operating Characteristic (ROC) analysis [8] [20].

The above described ANFIS is a continuous classifier, that is a mapping from the F-dimensional space (F is the number of features) to the interval [0,1]. To obtain a binary classifier for the oil spill two class classification problem, we must choose a threshold \( \tau \) on the classifier output. This resulting binary classifier can be coupled to a pair \((TPR, FPR)\), where \( TPR \) and \( FPR \) are, respectively, the True Positive Rate (the ratio of positives correctly classified to total positives) and the False Positive Rate (the ratio of negatives incorrectly classified to total negatives). The continuous classifier can thus be associated with a set of pairs \((TPR_\tau, FPR_\tau)\), each one corresponding to a certain threshold \( \tau \). This allows to evaluate the performance of the continuous classifiers in the space \((TPR, FPR)\), which is called the ROC space. The continuous classifier will be represented by a ROC curve: each point belonging to the curve corresponds to a certain threshold, and consequently to a certain binary classifier. So doing, the discrimination threshold can be chosen by observing the classifier ROC curve, basing on a trade-off between the \( TPR \) and the \( FPR \). In particular the target is to obtain the highest number of true positives, keeping the number of false positives as low as possible. The ROC curve is a two dimensional measure of the classifier performance. In order to have a scalar measure of it, we can consider the area under the ROC curve (AUC).

### 5.3 Results

We divided the dataset in a training set, composed of 60% randomly chosen examples, and a test set, composed of the remaining 40%, which will be used to evaluate the classification performance. For ANFIS training we used MATLAB `anfis` routine [11] [15]. The learning phase of `anfis` simultaneously tunes both antecedent and consequent parameters through a hybrid learning algorithm based on [16]: first the antecedent parameters, related to the membership functions, are determined by the backpropagation gradient descent method, then the consequent parameters are computed by means of a least-squares method based on Kalman filter. We trained ANFIS for 300 epochs, adopting Gaussian membership functions for the inputs and a linear membership function for the output. For the backpropagation algorithm we used an adaptive learning rate with an initial step size of 0.01, a step size decrease rate of 0.9, and a step size increase rate of 1.1.
In order to provide anfis with an initial Fuzzy Inference System structure we partitioned the data using a fuzzy $c$-means clustering algorithm. So doing, the number of clusters determines the number of rules and the membership functions for the antecedents and consequents in the generated FIS. We chose this number by training ANFIS for many times starting from different initial FIS structures, each one obtained by fuzzy $c$-means with a different number of clusters. We used 300 fuzzy $c$-means iterations and a fuzziness exponent of 2.0 for the clustering objective function. We performed each training running ANFIS for 300 epochs and we calculated the AUC on the test set. We iterated this procedure in order to reduce random initialization effects, and we chose the number of clusters corresponding to the maximum mean AUC on the test set. This number resulted to be 18.

Figure 4 shows a typical ROC curve obtained from the described ANFIS on the test set. The mean AUC achieved on the test set is 0.80.

6. **TOWARDS ONLINE COST-ORIENTED CLASSIFICATION**

In the framework of oil spill detection from optical satellite images, an online learning approach allows to easily improve the classification capability of the system when an existing dataset is updated by some oil spill candidates, which have been detected in a new available image. Once a dataset of oil spills and false alarms (look-aliases), has been collected, every time a new satellite image is downloaded and analyzed, the new candidates detected in the image will be included in the dataset and the proposed classification system will be able to learn quickly from these new candidates by means of the online learning process. The described situation well represents an operative scenario where a dataset of oil spills is available and it is being continuously updated (say daily) by the detection of other oil spills in some new images. These considerations suggest the investigation of an online approach to oil spill classification.

Another important issue is related to desired target conditions. More precisely, false alarms represent a relevant problem for oil spill classification. Thus, it is desirable to assign a different cost to the misclassification errors for each of the two classes, since this allows to perform an optimal classification with respect to a chosen cost index for class misclassification, that is, a cost-oriented classification.

In this framework, we tried to combine cost-oriented classification with an online learning approach, which also allows to use time varying costs and thus to change the desired target conditions (in particular the maximum false alarm rate) according to the latest classification results. In an operative scenario, this means that, for instance, if the coast guard verifies that too many look-aliases have been labeled as real oil spills, then the misclassification cost for the look-aliases, can be increased, thus improving the classification of future events.

In the following sections we show how to build an online cost-oriented classification system based on an ensemble of SVMs. We then show the results obtained by applying this classifier to the oil spill dataset.

However, we have to consider that the proposed approach meets the requirements of on-line classification, cost oriented classification and of using a suitable performance measure for a two class problem (as will be explained in the following sections) in despite of interpretability. Up to now we have not been able to include all these four requirements in a single framework, thus we focus on the first three.
6.1 Cost-oriented classification in ROC space

In Section 5.2 we have described ROC analysis for performance evaluation. In this section we will describe how the ROC space can be used to perform a cost-oriented classification.

As explained in Section 5.2, a continuous classifier can be associated with a set of pairs \((TPR_\tau, FPR_\tau)\), each one corresponding to a certain threshold \(\tau\). This allows to evaluate the performance of the continuous classifiers in the space \((TPR, FPR)\), that is the ROC space.

Basing on the definition of \(TPR\) and \(FPR\), in a two-class cost-oriented classification problem (\(p\) and \(n\) for positive and negative classes, respectively) the following cost index is often used:

\[
J(TPR, FPR) = FPR \cdot C(Y, p) + (1 - TPR) \cdot C(N, p)
\]

where \(C(Y, p)\) and \(C(N, p)\) are the cost functions associated to the misclassification of, respectively, a negative and a positive pattern, \(p(n)\) and \(p(p)\) are, respectively, the a-priori probabilities for negatives and positives. Such probabilities are typically estimated through the relative frequencies of positives and negatives in the available dataset. Here it is worth noting that the cost functions \(C(Y, p)\) and \(C(N, p)\) must be provided by the user and that they could change over time.

Once a continuous classifier has been trained, the optimal threshold \(\tau\) can be found by minimizing the cost index \(J\):

\[
\tau = \arg \min_{\tau_i} J(TPR(\tau_i), FPR(\tau_i)).
\]

As observed by Provost and Fawcett in [8], in the ROC space the iso-performance curves are straight lines when the cost index is the one defined in (1). Thus a set of costs and class distributions corresponds to a family of iso-performance lines characterized by the slope

\[
C(Y, n) \cdot p(n) + C(N, p) \cdot p(p)
\]

In particular, within a family of iso-performance lines, the line that is tangent to the ROC curve identifies the optimal point on the ROC curve (such point is \((TPR(\tau), FPR(\tau))\)). This allows to conclude that only points laying on the ROC convex hull, i.e., the convex hull of the ROC curve, can be actually optimal. When more continuous classifiers are available, as happens in classifier ensembles, just one convex hull can be computed for all the ROC curves, each curve being associated with one classifier. Figure 5 shows the ROC convex hull of three different ROC curves, and two different optimal points related to different cost index values.

In case of cost-oriented online classification based on an ensemble of continuous classifiers, if the training and the classification processes are performed over a sliding window, we can adopt the following strategy:

1) train each classifier over the current window;
2) draw the ROC curves by evaluating each trained classifier on a test set and by varying the threshold. Each classifier will produce a single ROC curve;
3) compute the overall convex hull, and select the best threshold associated with the best classifier.

Doing this, the optimal classifier and threshold are selected.

6.2 Repairing concavities in ROC curves

In order to improve the convex hull, we exploited a technique presented by Flach and Wu [17] for repairing concavities in ROC curves. The basic principle relies on the following observation: classifiers below the ascending diagonal in the ROC space perform worse than the random classifier, thus by simply inverting their predictions, the resulting classifiers will perform better than the random one. Inverting model predictions means obtaining a true positive rate equal to the complement to 1 of the original true positive rate, and a false positive rate equal to the complement to 1 of the original false positive rate. In the ROC space this corresponds to mirroring the original ROC point through the midpoint on the ascending diagonal. It is worth noting that this operation does not reduce the information content of the classifier. Since the ascending diagonal in the ROC space actually connects two classifiers, that are the classifier always predicting negative (the point in \((0,0)\)) and the classifier always predicting positive (the point in \((1,1)\)), this allows to generalize the prediction inversion procedure to linear segments connecting arbitrary classifiers.
Figure 5. Convex hull of the three ROC curves. Lines $\alpha$ and $\beta$ are two iso-performance lines, both tangent to the convex hull, but with different slopes, thus corresponding to different costs and class distributions.

Following this approach, the area under the convex hull can sometimes be increased by mirroring points under the convex hull to the other side [17]. This can be achieved because concavities give worse classification performance, thus, by swapping such points, the resulting mirrored points will be located in a more northwest region of the ROC space. Doing so, the ROC curves are modified and the new convex hull is consequently improved.

7. SUPPORT VECTOR MACHINES

A simple way to tackle a time varying cost scenario is to develop a single cost-oriented classifier, trained over a sliding window, and to dynamically select the optimal trade-off between false positives and negatives as explained above. So doing, however, one might run into classifier instability.

An empirical method to reduce instability is to employ a meta-classifier, consisting of a set of such cost-oriented classifiers, each one characterized by a static cost statistically chosen in the initialization phase. Within this set, the best classifier is dynamically selected as the one providing the best trade-off between false positives and negatives, according to the instantaneous classification error index.

For the sake of easiness, we assume that the meta-classifier is made of classifiers of the same type. The choice of the specific classifier type (decision trees, neural networks, etc) must consider, first of all, the availability of a cost-oriented training algorithm, and, possibly, the availability of an incremental and decremental training algorithm. In particular, this latter issue significantly improves adaptability, since the training algorithm exploits for each subsequent re-training only the patterns that enter (incremental part) or leave (decremental part) the sliding window, so the system is continuously updated. For this reason, we decided to adopt SVMs, which meet both requirements, and represent one of the most powerful classification techniques.

More precisely, we have integrated the cost-oriented formulation of SVMs (CO-SVMs) [18] and the incremental/decremental formulation of SVMs (ID-SVMs) [19] into a unique framework named COID-SVMs [20]. Then we have used an ensemble of COID-SVMs, each with its static misclassification error costs. The following section describes the proposed approach. For more details the reader can refer to [20].

7.1 COID-SVMs for online classification

The proposed method starts from the implementation of the ID-SVM by Cauwenberghs and Poggio [19], which has been modified to include cost-oriented classification in the ROC space, as described in Section 6.1, convex hull improvement through concavities repairing, and online classification. Regarding online classification, it is worth pointing out that in the original algorithm by Cauwenberghs and Poggio data acquisition was a batch process, while pattern evaluation was performed incrementally, following the same order as the acquisition. On the other hand, with the online approach, data acquisition happens incrementally, so the SVM structures can be dynamically modified, instead of being fixed by the dimension of the data acquired in batch mode. This intrinsic dynamicity improves the adaptability of the system to time varying conditions.
The online approach is handled by introducing a sliding window over which the training is performed. In order to build the online training, we structured the algorithm in the following three steps:

1) **SVM initialization:** this is performed by training the SVM over those patterns belonging to the window. In this way the SVM structure is initialized basing on a small data sample in batch mode. Thus, the resulting SVM trained on the window can be used in incremental mode. In this first step the only difference with respect to the original ID-SVM [19] is that the considered SVM is cost-oriented.

2) **SVM incremental learning:** a new pattern enters the window and the cost-oriented incremental learning starts.

3) **SVM decremental unlearning:** the oldest pattern exits the window and the decremental unlearning starts.

Besides the online training process, which is built by introducing the sliding window, performance evaluation in the ROC space (which implies drawing ROC curves, convex hull computation, choice of the optimum, possible performance improvement by repairing concavities) requires the presence of a validation set. In addition, performance should be then evaluated using a third set, the test set.

In order to integrate all these features, we adopted a sliding window having a size twice the size of the desired window for the learning. Each time, two new elements enter the window and two are excluded from the window. The SVM is trained over the odd elements within the window, while the even patterns compose the validation set, used to evaluate the expected costs, to build the ROC curves and the convex hull, and to repair concavities, if necessary. We can note that this procedure allows only one pattern at a time to enter the training set.

The test set is instead chosen outside the window, so that the pattern used for testing are not known by the SVM, thus, the settings established during the validation phase can be evaluated in an independent way. In particular, we adopted as a test set a number of patterns following the moving window equal to the size of the number of elements used for training and for validation. The size of the test set is thus half of the size of the sliding window. Figure 6 shows a scheme of the sliding window and the test set.

![Sliding window](image)

**Figure 6. Sliding window used for the online COID-SVM implementation.**

In the following section we will explain how the implemented model for online COID-SVM has been included in an ensemble of classifiers.

### 8. THE ENSEMBLE OF COID-SVMS IN THE ROC SPACE

Instead of using just one COID-SVM with dynamically changing $C(Y, n)$ and $C(N, p)$, we decided to use a set of COID-SVMs with static misclassification costs. This implies some advantages:

i) an ensemble tends to provide a performance that can outperform the single best classifier;

ii) an ensemble tends to provide more robust and stable performance;

iii) static cost functions give better stability to the algorithm used for the SVM online training.

The only drawback is the increase in computational complexity. However, the use of incremental and decremental learning mitigates this problem. Moreover, for the application to oil spill detection, we are not interested in building a classifier which is able to work in real time conditions, but rather to adapt the classification considering the most recent patterns. In particular, the computational complexity of the method is negligible if compared to the time needed to download a new satellite image, select, within the image, those regions which could contain possible oil spills, and extract the input features for the classifier.
All the COID-SVMs composing the ensemble undergo the training process described in Section 7.1, therefore the validation set is used to draw the ROC curves corresponding to each SVM, and to compute the overall convex hull. This means that the information derived by each classifier is automatically selected by means of the convex hull, and the optimum can be chosen according to the iso-performance line, taking into account all the classifier ensemble. In the same manner the overall concavities can be recovered, allowing for a fusion of the information provided by each classifier in place of a dynamic classifier selection. In Figure 7 we show a flowchart of the proposed approach for the classification based on the described ensemble of online COID-SVMs in the ROC space.

![Flowchart of the proposed approach for the classification based on an ensemble of online COID-SVMs.](image)

**Figure 7.** Flowchart of the proposed approach for the classification based on an ensemble of online COID-SVMs.

### 8.1 A software for online cost-oriented classification

We integrated the proposed algorithm for building an ensemble of online COID-SVMs, described in Section 8, in a software tool, which is structured following the flowchart in Figure 7 [20]. The implemented ensemble is composed of five SVMs, namely COID-SVM 1, COID-SVM 2, COID-SVM 3, COID-SVM 4 and COID-SVM 5.

The software is provided with a graphic interface which allows the user to set the inputs and to obtain the optimum for the classification basing on the convex hull method (see Section 6.1). The user can select the dataset and can set the static misclassification costs for each COID-SVM in the ensemble ($M(Y, n)$ and $M(N, p)$), the cost functions which define the iso-performance lines ($C(Y, n)$ and $C(N, p)$), and the size of the sliding window. In particular, the cost functions can be either constant or time varying. In the latter case we will use the symbols $C(Y, n)(t)$ and $C(N, p)(t)$. For each data subset entering the sliding window, the software produces a plot showing the ROC curve for each SVM, the convex hull and the optimum for the classification, all computed over the validation set. Moreover, the cost index $J(\text{TPR}, FPR)$, defined in (1), is computed for the ensemble on the validation set and on the test set. Possible concavities in the convex hull can be repaired by enabling the corresponding function.

### 8.2 Online cost-oriented classification results

In the following we present the results obtained by applying the proposed online COID-SVM ensemble to the oil spill dataset, and we investigate the effect of repairing convex hull concavities

**Experiment without concavities repair**

In Figure 8 we show a typical result for an online classification step. The figure shows the ROC space where the ROC curves for the ensemble of online COID-SVMs are drawn. The violet line represents the overall convex hull and the black line is the iso-performance line tangent to the convex hull. The black squared mark shows the optimum resulting from the algorithm. In order to find a trade-off between the size of the dataset (304 elements) and the necessity to test the...
online learning procedure, we chose a sliding window of 100 elements, 50 used for training and 50 for validation. According to Figure 6, we tested each online epoch on 50 test elements.

We used time varying sigmoid cost functions $C(Y, n)(t)$ and $C(N, p)(t)$ with values in the interval $[1, 2]$. As shown in Figure 9, where the costs are plotted as a function of the online epoch number, $C(Y, n)(t)$ decreases with time while $C(N, p)(t)$ increases.

The static misclassification costs associated with each COID-SVM in the ensemble are shown in Table 4. Note that these misclassification costs could assume any value. In particular for each SVM, $M(Y, n)$ and $M(N, p)$ are not necessarily equal. In this application we found that using equal misclassification costs for the positive and negative classes, for each COID-SVM, gives better performance. This could be explained with the fact that the oil spill dataset is actually balanced, since the number of elements belonging to the oil spill class is approximately equal to the number of elements belonging to the look-alike class. Indeed, according to the cost oriented formulation of SVMs used in this paper [18], the static misclassification costs $M(Y, n)$ and $M(N, p)$ represent the upper bound on the maximum allowed distance between an element and the separating hyperplane associated, respectively, to the negative class and to the positive class. Thus, using equal values for the two misclassification static costs does not exactly mean that we are considering the cost of misclassifying a positive element equal to the cost of misclassifying a negative one.

![ROC curves for the ensemble of online COID-SVMs. The violet curve represents the convex hull, whereas the black line represents the iso-performance line. The black squared mark is the optimum.](image1)

![Time varying cost functions $C(Y, n)(t)$ and $C(N, p)(t)$ represented as a function of the online epoch number.](image2)

A measure of the classification performance of the system can be achieved by considering the cost index $J(TPR, FPR)$, defined in (1), computed on the test set at the optimal threshold for the ensemble. (Note that the optimum is computed by the algorithm on the validation set). However, we have to consider that each online step produces an optimal threshold for the classification of the elements belonging to the current sliding window.
Taking this intrinsic dynamicity of the system into account, in order to obtain a performance index that involves the complete dataset, we sum all the cost indexes, each one corresponding to a single online epoch, over all the online steps, so as to obtain an integrated global cost index for the ensemble applied to the dataset. On the resulting 77 online epochs we obtained a global cost index of 92.89 on the test set and of 77.39 on the validation set.

<table>
<thead>
<tr>
<th>misclassification costs associated to each COID-SVM in the ensemble.</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(Y, n)</td>
</tr>
<tr>
<td>COID-SVM 1</td>
</tr>
<tr>
<td>COID-SVM 2</td>
</tr>
<tr>
<td>COID-SVM 3</td>
</tr>
<tr>
<td>COID-SVM 4</td>
</tr>
<tr>
<td>COID-SVM 5</td>
</tr>
</tbody>
</table>

Regarding the mean execution time on the considered dataset, in terms of elapsed CPU seconds, for a single online step this is of 5.52 s, on AMD Athlon X2 2.6 GHz processor, with 2Gb RAM.

**Experiment with concavities repair**

The application of the technique described in Section 6.2 results in a modification of the convex hull, which brings to an increase in the area below the convex hull. In order to estimate the effect of repairing concavities in the convex hull, we applied the corresponding function at each online step, and we computed the area under the convex hull. On the 77 online epochs we obtained a mean area under the convex hull of 0.70, while repairing concavities at each online epoch we obtained a mean area under the convex hull of 0.72, thus increasing the area of about 3%. Moreover, the increase in the area corresponds to a decrease in the global cost index of the 2% on the test set and of the 5% on the validation set. In Figure 10 we show an example of the effect of repairing concavities in the convex hull. In the figure the variation in the area is highlighted in grey, the optimum computed on the improved convex hull is represented by a red squared mark while the old optimum is represented by a black squared mark. From the theoretical point of view, such an improvement is even more significant, because we have demonstrated that combining classifiers provides better results, in this case, than performing a classifier dynamic selection.

![Figure 10. Effect of repairing convex hull concavities.](image-url)
9. CONCLUSIONS

In this paper we have tackled oil spill classification from optical satellite images by exploiting many different machine learning techniques. In particular, we started our analysis by employing simple statistical classifiers (normal density-based linear and quadratic classifiers, logistic regression-based linear classifier, k-nearest neighbor classifiers) and two neural networks (a radial basis function and a multi layer perceptron), measuring the performance by means of the correct classification percentage. Then, taking into account the limitations of performance evaluation based on classification accuracy in a two-class problem, we have moved to ROC analysis, using a more interpretable classifier (a fuzzy rule-based classifier embodied in an ANFIS network). Finally, since, in practice, the oil spill dataset is collected incrementally, we have looked for a technique which could be suitable for online classification. Moreover, we tried to encompass cost-oriented classification, in order to allow for a dynamic change of the misclassification costs, according to the changing needs. We have shown how using an ensemble of cost-oriented, incremental and decremental SVMs is a way to address our classification requisites, exploiting the concept of the ROC Convex Hull and of concavities repair.

We performed all the above mentioned studies on a dataset of more than 300 oil spills and natural phenomena detected from multi-spectral satellite sensors during the years 2008 and 2009, over the entire area of the Mediterranean Sea.

The promising results that have been achieved highlight the potential of using optical satellite data for oil spill detection. In future works we will try to address the need for interpretability of the classifier (which is a desiderata of end-users), preserving in a single framework the cost-oriented and online features achieved so far.

REFERENCES