A Version of this paper has been published in Engineering Applications of Artificial Intelligence, 24, pp. 1070-1083, 2011

Comparative study of supervised classification algorithms for the detection of atmospheric pollution.

D. Gacquer[1], V. Delcroix[1], F. Delmotte[2], and S. Piechowiak[1]

[1]UVHC, LAMIH, F-59313 Valenciennes, France
[1]FSA, LGI2A, F-62400, Bethune, France

Abstract

The management of atmospheric pollution using video is not yet widespread. However it is an efficient way to evaluate the polluting rejects coming from large industrial facilities when traditional captors are not usable. This paper presents a comparison of different classifiers for a monitoring system of polluting smokes. The data used in this work are stemming from a system of video analysis and signal processing. The database includes the pollution level of puffs of smoke defined by an expert. Six machine learning techniques are tested and compared to classify the puffs of smoke: K nearest neighbor, naïve Bayes classifier, Artificial Neural Network, decision tree, Support Vector Machine and a fuzzy model. The parameters of each type of classifier are split in three categories: learned parameters, parameters determined by a first step of the experimentation, and parameters set by the programmer. We compare the results of the best classifier of each type depending on the size of the learning set. A part of the discussion concerns the robustness of the classifier facing the case where classes of interest are under represented, as the high level of pollution in our data.

Classification Machine Learning Multilayer Perceptron Artificial Neural Network Support Vector Machine Bayesian Network Nearest Neighbour decision tree Fuzzy model Air pollution

1 Introduction

The consideration in environment preservation has been considerably growing during the last decade. Pollution aspects are now considered from local to global dimension (Fenger, 2009). This consideration has led to the establishment of international environment law

and more particularly the Polluter Pays Principle, also known as extended polluter responsibility, where the polluting party pays for the damage done to natural environment.

This has drawn the interest of researchers in a wide range of applications. This is the context of the present article, which focuses on the control of air pollution and addresses this control as a supervised classification application. The objective is to determine automatically the impact, in terms of gravity, of hazardous smokes rejected by factories, to both the environment and the local population living in the area. Most of the studies concerning pollution detection systems are based on chemical analysis, using either chemical measurements or laser detection (Bakos and Tsagas, 1995), and little attention has been given to automatic systems for monitoring industrial facilities and classifying aerial pollution.

For example, in (Zolghadri et al., 2004) then authors present a study on ozone concentrations monitoring. Meteorological variables are used as input in order to obtain an estimate of the next days maximum ozone concentration. The proposed approach is based on the use of a multi-layer perceptron neural networks.

Unlike usual pollution control systems, the system described in this paper uses visual information from a camera to detect and predict the potential danger of polluting rejects. The choice of using a camera instead of local sensors is justified by the structure of the industrial estates where the system is used. Those complexes are very large, generally, they consist in administrative buildings and large factories close together; in these industrial sites, hazardous smoke rejections can occur everywhere over the complex. It is not possible to place local sensors on each chimney which can potentially reject polluting trails, because of their important numbers, but also due to the difficulty to access them (high altitude for instance). Additionally, local sensors are often specific to particular types of pollution whereas the video-based system allows to consider on the whole all the visible components of the pollution. Moreover, the time required to perform chemical analysis on the collected samples prevents this type of sensor from being used for online monitoring.

This paper proposes a novel approach to address the problem of air pollution. To solve the limitations of chemical sensors and provide a real time monitoring system, Machine Learning techniques are used to process the information recorded by the camera. The objective is to design a monitoring system which is able to automatically recognize polluting smokes from previously typical scenes recorded by the camera and used as a training set. The camera is the unique visual sensor used by the system. It is placed at a certain distance from the complex and captures the activities of the different factories. The camera provides a global view to detect hazardous smoke from anywhere. This automatic system does not consider pollution as a continuous phenomenon but as a series of discrete events. Indeed, the system only detects and classifies particular situations, when the scene recorded by the camera corresponds to a puff of smoke, whether it is dangerous or not. A specificity of this problem is that the classes of interest are under represented: the puffs of smoke corresponding to a high level of pollution are about ten times less frequent than smokes that are not or little dangerous.

In this context, the choice of a classification technique is not trivial. Broadly speaking, no classifiers is known to be systematically the best. The literature offers number of

comparison of classifiers for specific problems or specific criteria (Dietterich, 1998; Frias-Martinez et al., 2006; Muñoz Expósito et al., 2007). To answer this question, we compare on real industrial data six well known Machine Learning algorithms: K nearest neighbor, decision tree, naïve Bayes classifier, multilayer Perceptron (Neural Network), Support Vector Machine and a fuzzy model. The remainder of this paper is organized as follows: in the next section, we introduce both the architecture of the monitoring system and the database used for our experiments. The following two sections detail respectively the supervised classification problem and the six algorithms previously mentioned. In particular, we expose the experiments used to select the best parameters for each classifier. The last section concerns the comparison of these classifiers on industrial data. We first discuss about their robustness respectively to parameter settings, and then we compare their performances respectively to the size of the training set. The last discussion deals with the robustness of these classifiers in the particular case where we only have a few samples of the classes on which we focus.

2 Experimental material

2.1 Context motivation of the study

The ALOATEC Company designs mainly monitoring and quality control solutions for its industrial partners. This collaboration has led to the development of the DETECT system, a product designed for the visual detection of polluting smokes around large industrial complexes. The development of this system is motivated by the need to have a detailed and quantified summary of all hazardous events that occur in the region. The original classification program developed by ALOATEC is an expert system. This solution requires one or more human experts familiar with the industrial complex where the DETECT system is being used, to adjust the camera settings, but also to define the different rules of the classification program. This task can be especially complex since pollution can take various forms according to the activities of a given industrial complex. This implementation implies heavy constraints concerning installation and evolution of the system. Besides, the efficiency of expert systems mainly relies on the possibility to formalize the problem into an accurate and finite set of rules. Due to the complexity of the process, the first implementation of the system achieved an important number of misclassified objects. Moreover, it was not possible to adapt the system to different data sets from other problems. Another lack of the first monitoring system of polluting smoke is the high number of misdetections and over detections of severe pollution. To solve the limitation of the first system, the authors replace the original classification program with machine learning algorithms adapted to the pollution detection problem. Since the first implementation of the system has been used for a certain period over different industrial sites, a sufficient database of samples is available as training material for the work exposed in this article.

2.2 Extraction of the features

The camera monitors the activity of a given complex in a continuous manner, that is, image processing continuously translates visual information from the scene into numerical variations of different signals. The camera is placed outside the complex, so that all the areas where hazardous rejections may occur fall into the range of the picture. A typical example of puff of smoke is given in Figure 1. To protect the camera from weather conditions and limit the introduction of noise in the picture, it is placed into a box behind a glass panel, so that its objective remains clear.

Figure 1: Typical scene detected by camera.

A simplified architecture of this pollution detection system is illustrated by Figure 2. A detection algorithm determines time windows corresponding to potentially hazardous trail rejections. (cf. time interval [a, b] on figure 2). This is done thanks to a frequency analysis on a small windows of the video image. While the smoke continues, several signals are recorded into a temporary database. As soon as the end of the cloud is detected, the features are extracted from these records by using function on the time interval. The functions such as min, max, integral, mean, or delta between max and min depend on the physical signification of the signal. Each detected trail of smoke constitutes an event and is recorded with a frequency of one picture every five seconds. The features of the cloud are stored into a permanent database together with the corresponding pictures.

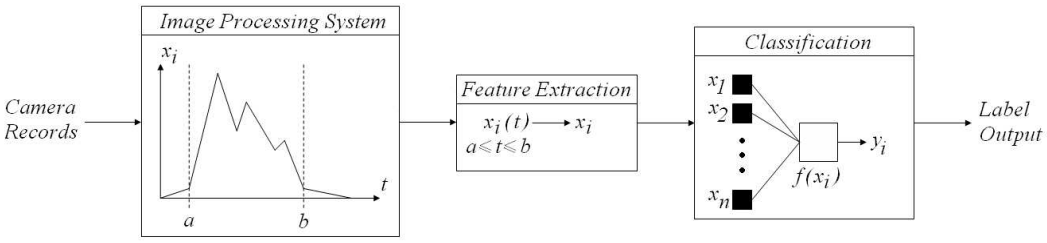

Figure 2: The pollution detection system based on image analysis.

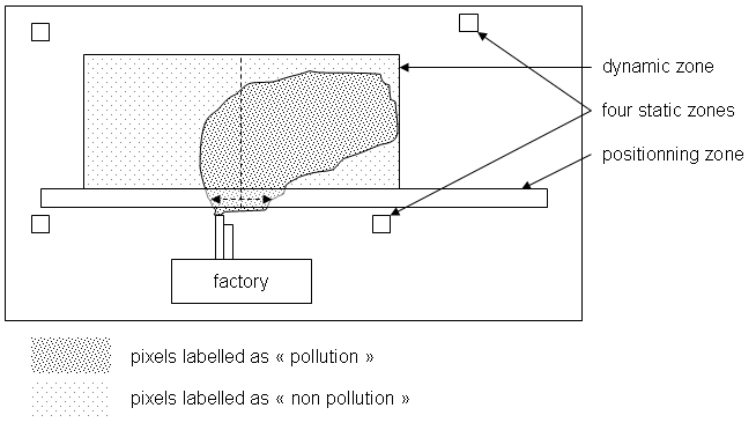

Figure 3: The different zones used to the analyse of the image.

Figure 3 shows the different zones used to analyse the image. The signals are recorded from static and dynamic zones, as soon as the beginning of a puff of smoke has been detected, and until the end is noticed. The centre of the emission is determined thanks to the positioning zone. The dynamic zone is automatically positioned on the cloud as soon as one is detected. The static zones are four little areas of the image without any intersection with the cloud. These zones allow to measure the difference between pixel marked as pollution and those marked as no pollution. Table 1 presents the list of the signals extracted from the filmed sequence corresponding to a puff of smoke. These pieces of information are the surface of the puff of smoke (signal 2 and 22), its density (signals 18 to 21), its colour and its duration. The appreciation of the colour of the cloud (signals 7, 16 and 17) is done by comparison to the local colour of the sky (signals 3 to 6 and 8 to 15). The red colour is a sign of high level of gravity, but this is to be modulate according to the surface, density and duration of the cloud. None of these signals can be used as a sufficient indicator of the gravity level of the cloud. They must be combined to give a usable information.

The feature extraction is done to reduce the data characterizing an event, while keeping the maximum of information, including interesting information coming from the dynamic of the signal. The temporal data are transformed in order to obtain quantitative variables that characterize the corresponding cloud. One of the following transformations is applied on a signal or a combination of these signals: integral, mean, maximum, minimum or delta between max and min. Most of the above signals have been aggregated (by product for example) in order to reduce the number of variables. This step has been achieved by hand without significant loss of accuracy on the quality of the classification on several individual test.

This process explains why the activity of the various factories is monitored as a succession of discrete events and not like a continuous process. Also, unlike conventional pollution detection systems, the control is performed through visual information from the

Table 1: *List of the signals corresponding to a puff of smoke.*

signal Number	Name	simplified explanation
1	luminosity	luminosity in a upper corner of the image, without any pollution used to detect the nightfall
2	surface	percentage of pixels of the dynamic zone labelled as pollution is computed
3 to 6	luminosity of static zones	luminosity of the four little static areas defined on the filmed image as a reference
7	luminosity of dynamic zone	Luminosity of the most representative point labelled as pollution in the dynamic zone
8 to 15	colour of the static zones	coordinates on the XY chromatic diagram of the colour filmed in the four static zones
16, 17	colour of the dynamic zone	coordinates on the XY chromatic diagram of the colour of the most representative point labelled as pollution in the dynamic zone
18 to 21	low, medium, high and very high density	proportion of points of the dynamic zone having more than a certain number of neighbours labelled as pollution
22	maxEmission	this signal reproduces the evolution of the importance of the cloud during the time window

trail and not from its chemical analysis.

2.3 The Database

The database provided by the ALOATEC Company consists in 2900 events recorded during several months of activity on the same industrial complex; it contains no missing value. A puff of smoke represents an event (or object) to be classified. It is characterized by the class label and 12 numerical features that are computed from the 22 signals described above. The details of the aggregation and transformation of the temporal signal into a scalar cannot be explained here due to confidentiality reasons. The class label is the severity level of the cloud defined by a human expert from the domain. The level ranges from 0 to 3 and zero level corresponds to a water vapour cloud, whereas a level three corresponds to the highest pollution. The different classes are not equally represented in the database, since there is a large amount of degree 0 pollution compared to the other classes. Class distribution is given in Table 2. The low ratio of critical cases makes more difficult the learning step of the classifiers despite the high number of learning cases.

Table 2: *Class distributions in the experimental dataset.*

	Class 0	Class 1	Class 2	Class 3
samples	2085	421	143	235
Ratio	72 %	15 %	5 %	8 %

3 A Classification Problem

The problem of estimating pollution levels from the camera monitoring is a classification problem. The problem is to determine what level of pollution is associated with an observed situation, ie to find which class of pollution should we relate this situation. There are two steps: first a learning step from the database defined previously, and second an inference step, to estimate on line the level of pollution.

Six common classification algorithms are used and compared in this study: k-nearest neighbour (knn), decision tree, naïve Bayesian network (NB), Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Takagi-Sugeno Fuzzy model.

Most classifiers include parameters that have to be tuned. In this study, an additional step has thus been added to determine, for each of the six classifiers, the parameter settings that are best suited for our detection system. This step is performed by running 10-fold cross validation (Kohavi, 1995) for each possible parameter configuration. Given a particular classifier, this process is repeted for each possible values of its parameters. These parameters are the number k for knn, the minimal number of objects on a node of a decision tree needed to split the node, the number of hidden units and of hidden layers for MLP, and the cost of constraint violation and the kernel parameter sigma (for gaussian kernel) for SVM. The results of this step is detailed in section 5 where the parameters of each classifier are explained as well as the way they are manually determined or learned during training.

Parameter values that lead to the best results are used during the second step of our study in order to compare the performances of the different classifiers. Hold-out validation is used on the whole dataset, which is divided in two different subsets containing respectively 60 % of the samples to train the classifiers and 40 % to compute their performances. This process is averaged across 50 independent runs to evaluate the performances of each classifier.

The usual measure of the performance for a classifier is its accuracy, that is the proportion of correctly classified samples over the test set.

$$accuracy = \frac{\sum_{i=0}^{3} N_{ii}}{\sum_{i=0}^{3} \sum_{j=0}^{3} N_{ij}} \tag{1}$$

where N_{ij} is the proportion of events that have been assigned the class i by the expert and the class j by the classifier. Certain types of misclassifications are more important in this real world situation. The worst possible cases are false alarms and missed detections,

i.e. events of actual degree 0 (respectively 3) labelled as degree 3 (respectively 0) by the system.

Regarding this remark, the accuracy is not well adapted to evaluate the performances of the classifiers. The system has to provide a very good detection of the critical events (of level 3). Indeed, the inappropriate start of an alarm due to an error of classification has a very high cost. The same is true for the non detection of a serious pollution, that must absolutely be avoided. start of alarm and has very little bad consequences for the The ideal classifier should correctly classify all the events of level 3 and produce no wrong alarm, regardless of an acceptable number of misclassifications regarding events of lower level. Such a classifier would be preferred to another one with a higher accuracy, but including misclassification of level 3 events. Therefore another criterion, called efficiency, is also used in this work. This efficiency criterion has been defined by the end-user, and is equal to:

$$efficiency = \frac{0.3 \cdot (N_{13} + N_{31}) + 0.8 \cdot (N_{23} + N_{32}) + N_{33}}{\sum_{i=0}^{3} N_{i3} + \sum_{j=0}^{2} N_{3j}} \tag{2}$$

.

Both accuracy and efficiency are considered for parameter selection and to compare the performances of the different classifiers described in the next section.

4 Setting the parameters of the classifiers

Different types of classifiers are evaluated and compared in this study : three well-known classification algorithms (k-nearest neighbour, neural networks and decision trees), and a kernel-based classification algorithm, support vector machines (SVM), but also Naïve Bayesian algorithm (NB) which is generally used for text data, and additionally, we investigate the usefulness of a fuzzy model. Theses classifiers are briefly describes in order to support discussions about parameter selection. A distinction is made between parameters that we have manually set such as SVM kernel type for instance, and parameters that were determined using 10-fold cross validation. These parameter settings are used in the next section of this paper to compare the performances of the different classifiers applied to the pollution detection system.

Experiments are run on a Pentium IV CPU with 1 Go of memory, using Matlab environment. The following Matlab toolbox have been used: the BNT-SLP package (Leray and François, 2004) for Bayesian network, the OSU-SVM Matlab toolbox for SVM, the *nnet* package for neural network, the *stat* package for decision tree and fuzzy model.

4.1 K-Nearest Neighbours

K-nearest neighbour (knn) (Shakhnarovich et al., 2006; Dasarathy, 1990) is an instance-based learning algorithm (Aha et al., 1991) which considers the training data as a set of prototypes, for which the class is known. To classify a new input vector, Euclidean distances between each prototype and this new vector are computed. The k nearest samples

are retrieved from the training set, together with their class labels, and the class which is the most represented in the vicinity defined by those k samples is assigned to the new input vector. Several drawbacks of this algorithm are reported in (Aha et al., 1991).

In this study, the usual implementation of the algorithm is slightly modified so it better fits the pollution classification system. Classes are ordered integer values ranging from 0 to 3. The label output is considered as a continuous value in the interval [0, 3], and the usual majority vote used among the k nearest neighbors is replaced by the following decision rule:

$$class(x) = round(\frac{\sum_{i=1}^{k} 1/d_{xi} \cdot class(i)}{\sum_{i=1}^{k} 1/d_{xi}}) \tag{3}$$

where d_{xi} represents the distance between sample x and the i^{th} nearest neighbor, so that the closer a data point is from sample to be classified, the more it contributes in the final decision rule. Note that this particular modification of the original algorithm only makes sense when the class labels are ordered integer values.

Figure 4 plots the misclassification error and efficiency obtained with k-nn algorithm for different values of k using 10-fold cross validation. By increasing the value of k, the effect of noise is reduced, but a too large vicinity makes decision boundaries less accurate and lowers the predictive ability of the algorithm. However, since distances with all the training samples are computed whatever the value of k, the number of neighbors has no effect on computation times. Best results for both classification accuracy and efficiency were obtained with the value $k = 11$. This algorithm provides good overall performances, but suffers from the time required to compute the class label, since there is no training step for this method.

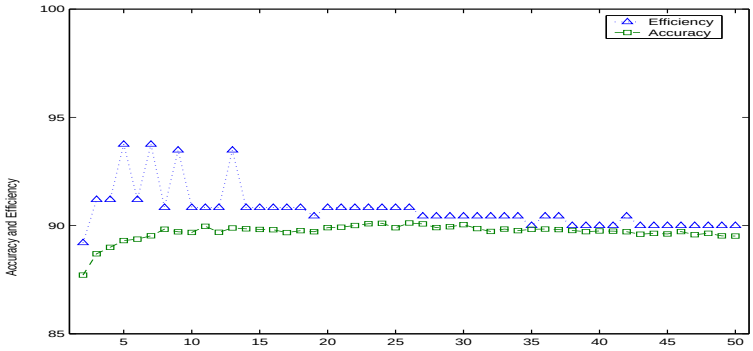

Figure 4: Influence of parameter k over performances of knn.

4.2 Decision Trees

Decision trees learn a classification procedure by hierarchically building a tree over the training samples. The main tree learning algorithms are *CART*, introduced by (Breiman et al., 1984), *ID3* (Quinlan, 1986) and *C4.5* (Quinlan, 1993). Each node is associated with a split on one feature. The feature chosen to split a given node can be determined with different evaluation functions (the Gini's diversity index, the entropy or the error index. According to Duda et al. (2001), the choice of this function is not as determinant as the choice of the pruning method used to determine the structure of the tree. In our implementation, we use the *CART* algorithm and the Gini's diversity index. The overfitting problem can be tackled using two methods: either by stopping the growth of the tree or by post-pruning the tree once it has been constructed, which is the method used in this paper. One of the conditions for a node to be split is that it carries a minimum number of observations. This number is a parameter of the classifier that we note *splitmin number* in the next sections of this paper.

The figure 5 shows the influence of the splitmin number over the performances of the decision tree for the pollution detection problem. Both accuracy and efficiency slightly increase when the minimum number of observations required for a node to be split increases. The higher is the split number, the smaller is the tree, and consequently, the faster is the algorithm, for both the learning step and the classification step.

4.3 Naïve Bayes Classifier (NB)

Bayesian Networks (Jensen, 2001; Pearl, 2000) are probabilistic graphical models that represent conditional dependencies between a set of variables. A naïve Bayes Classifier is used here. This simple probabilistic model relies on the strong assumption of independence between the feature variables. The class node is parent of all the observed featuresThe naïve Bayes classifier is surprisingly useful for many real-world problems and often works much more than expected [14, 12]. It is trained to determine the class label from an observed input vector. A new feature vector is labelled with the class whose posterior probability is maximal, using the decision rule.

In this study, parameter selection is not required for the Naïve Bayes classifier since the structure is already determined. Training consists in estimating the probability distributions for each node using the maximum likelihood. The continuous features provided by image processing need to be discretized first. The use of continuous variables for the Naive Bayes is not feasible here because most of the continuous variables that we use do not have a Gaussian distribution. To ensure optimal discretization, the algorithm exposed in (Olivier et al., 1994) is used prior to the training step. An alternative solutions to the naïve classifier is explored in (Gacquer et al., 2006).

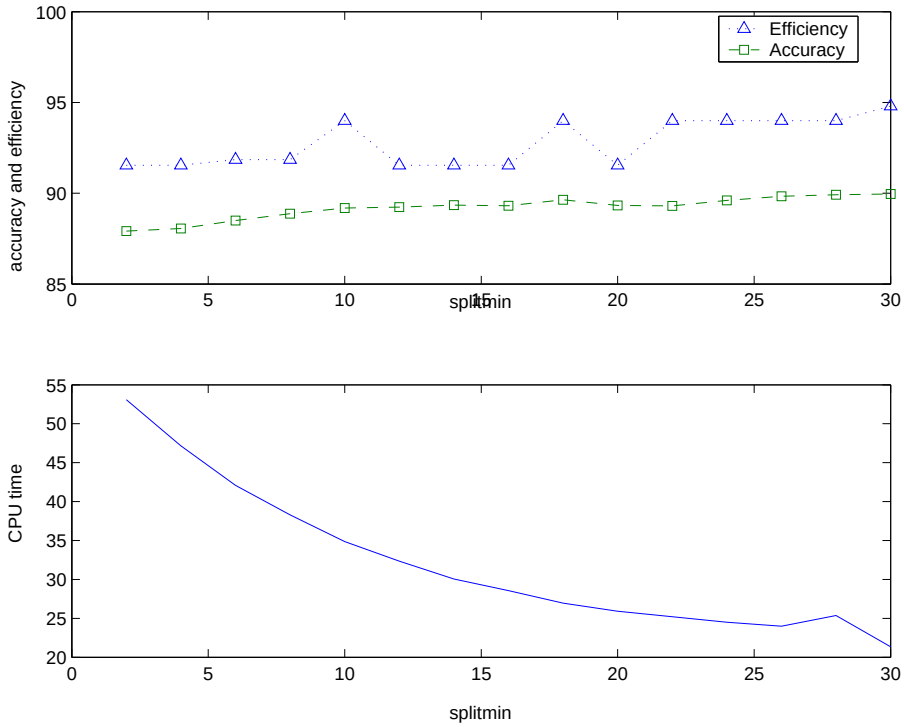

Figure 5: Influence of the splitmin number over performances of decision tree and the corresponding CPU time, in second, required for 50 runs

4.4 Multilayer Perceptron (MLP)

For the purpose of pollution detection, we use a Multilayer Perceptron which is a very common type of Artificial Neural Network (Ripley, 1996). It is used to map the feature vector into the corresponding class label. The target function $f(x)$ is given by $f(x) = g(\sum_{j=1}^{m} w_i h(\sum_{i=1}^{n} v_{ij} x_i))$, where g and h are the activation functions of the output unit and the hidden units respectively, and w_i and v_{ij} are the weights between the different units of the network. We perform the training of feedforward networks as follows: initializing the weights, using the algorithm described in (Nguyen and Widrow, 1990), and finding the weights, using Levenberg-Marquardt algorithm (Hagan and Menhaj, 1994). During training, early stopping is performed to avoid overfitting problems (Tetko et al., 1995; Sarle, 1995). A subset of randomly chosen training samples is used to stop training when generalization properties of the network are optimal. Sigmoid and linear activation functions are used respectively for the hidden units and the output units of the network. Prior to training of the network, data preprocessing is performed so that input values fall into the range $[-1, 1]$, to lessen the influence of outliers in the distributions. The output layer has four units, one for each class.

We first use a 3-layer perceptron, *i.e.* with one hidden layer, trained with different number of hidden units. Figure 6 shows the influence of this parameter over the performances

of the MLP. The predictive ability of the network is not very sensitive to the variations of the number of hidden units; around 91% of the events are correctly classified. However, the time needed to train the MLP increases significantly with the number of hidden units.

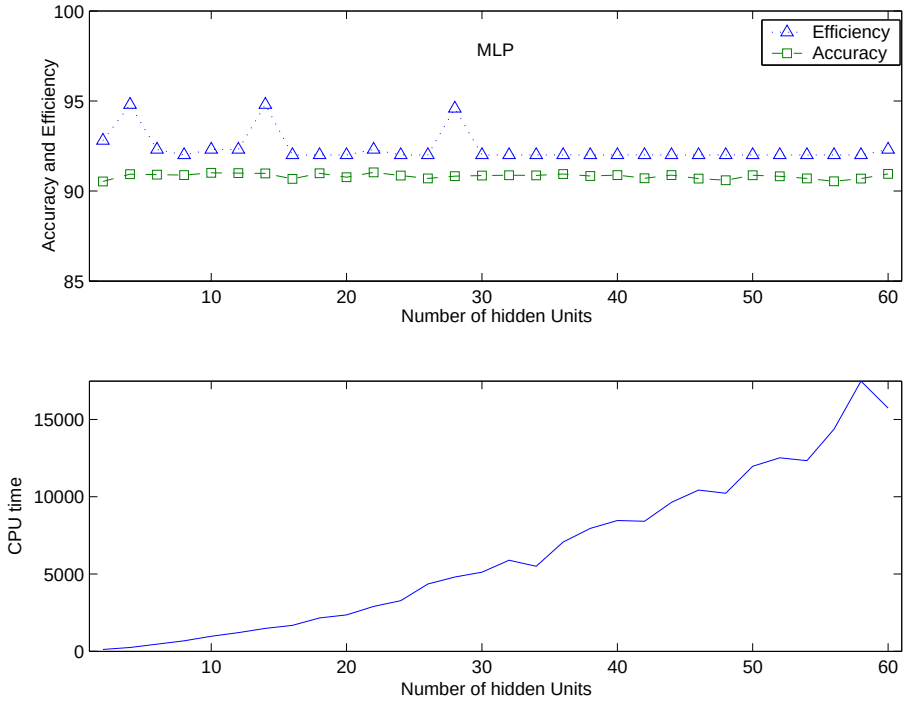

Figure 6: Influence of the number of hidden units on accuracy, efficiency for a 3-layer perceptron and the corresponding CPU time, in second, for 50 runs.

The second experiment concerns 4-layers perceptron, (with two hidden layers), that are trained with different number of hidden units on each hidden layer. The accuracy and efficiency of theses 4-layers perceptrons are plot on figure 7 as a function of the numbers of hidden units on each hidden layer.

The maximum of efficiency is $94,8$ both for a 3-layers MLP and a 4-layers MLP. This maximum is reached several times, for example with 4 or 14 hidden units in a 3-layers MLP, and for respectively 7 and 6 hidden units on the first and second hidden layer in a 4-layers MLP.

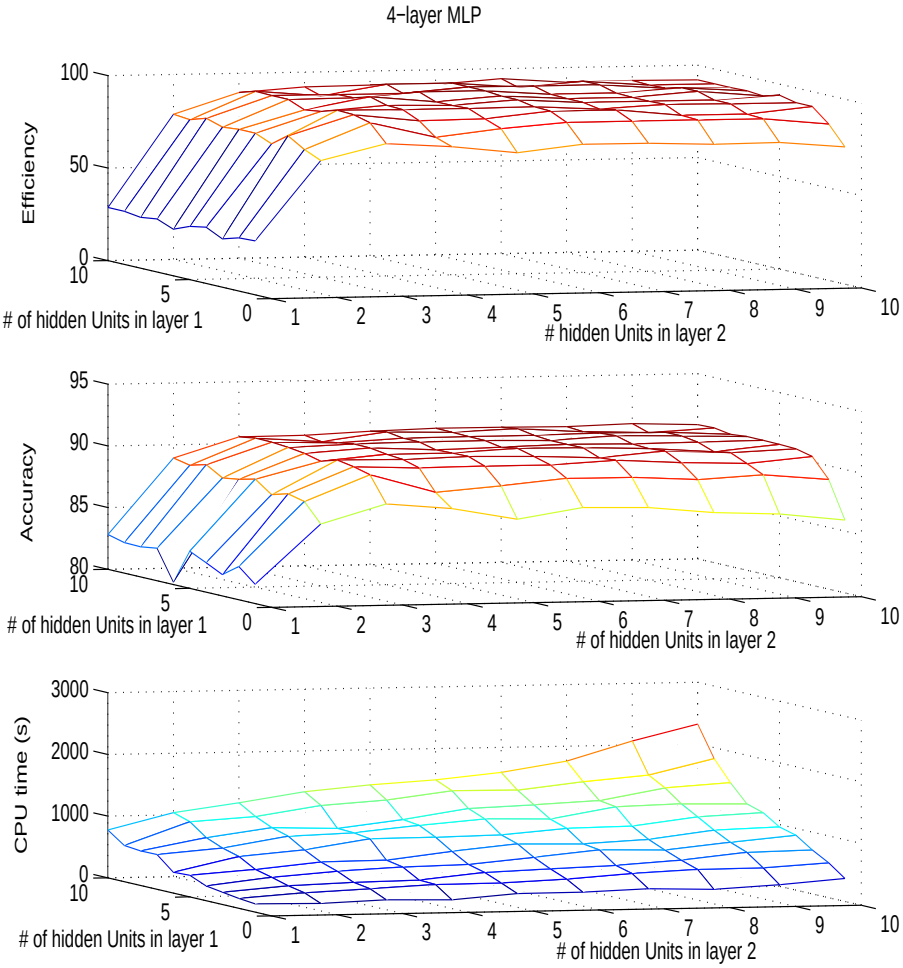

Figure 7: Influence of the number of hidden units on accuracy, efficiency with 4-layers perceptrons, and the corresponding CPU time for 50 runs.

4.5 Support Vector Machine (SVM)

Support Vector Machines (Vapnik, 1998; Cortes and Vapnik, 1995) are basically used for binary classification problems. Formulating classification into an optimization problem, SVMs find the hyperplane maximizing the distance between the two classes. This approach simultaneously minimizes the empirical classification error and increases the generalization properties by maximizing the geometric margin. Usually the input vector is mapped into a higher dimension space so that the samples become linearly separable. The decision function in the augmented space is $f(x) = \sum_{i=1}^{N_{SV}} \alpha_i \Phi(x_i)\Phi(x)$, where N_{SV} is the number of support vectors and Φ is the transformation function $\Phi : \mathbb{R}^n \to \mathbb{R}^m$. Since the decision function only depends on the scalar products $\Phi(x) \cdot \Phi(x')$, the new optimization problem is solved by choosing an appropriate kernel function $K(x, x') = \Phi(x) \cdot \Phi(x')$. Classification accuracy greatly depends on the choice of the kernel function. Polynomial ones can be very time consuming, so in this experiment, we use Gaussian kernel function $K(x, x') = \exp(-\sigma\|x - x'\|^2)$, where σ is a user defined parameter.

The previously described implementation only applies to binary classification. To solve multi-class problems, it is necessary to decompose the initial formulation into a set of binary sub problems. Several implementations of multi-class SVM can be found in the literature, like One versus One and One versus Rest decomposition schemes, or Error Correcting Output Codes (Chang et al., 2000; Dietterich and Bakiri, 1995; Allwein et al., 2001). Multi-class SVMs is still an active research topic. The computation depends on the number of constraints. In the case studied here, the One Versus One model has been chosen. See (Allwein et al., 2001) for further insights about this choice.

In this study, we select two parameters that require optimization for Gaussian kernel SVM classifiers: the cost of constraint violation C in the optimization problem, and the parameter σ of the kernel function. The adjustment of σ when using Gaussian kernel greatly influences the predictive ability of the classifier. To determine the optimal value of these parameters, we train several classifiers with different values of σ and C. Results are reported in Figure 8. The maximal value of classification accuracy is 89.4 and is obtained with $\sigma = 10^{-9}$ and a cost of constraint violation $C = 9100$; the maximal value of efficiency is 92.8 with $\sigma = 10^{-8}$ and 92.0 with $\sigma = 10^{-9}$ and the same cost $C = 9100$. The training time is maximal for the same cost $C = 9100$ but $\sigma = 10^{-7}$. The influence of the cost of constraint violation is displayed on figures 9 and 10 with $\sigma = 10^{-9}$. The higher C, the higher classification accuracy and efficiency both are, but also the higher the time required for training is. In such a case, the best choice of parameters is a compromise between time and performances.

A low value for C lowers the accuracy of the decision boundaries determined by the optimization problem, and reduces the number of correctly classified samples. On the contrary, setting C to higher values makes the optimization problem harder to solve and increases the time required to train the classifier, reported in Figure 10. For a value of C greater than 10^4, the number of correctly classified events can be slightly improved but the time required to train the classifier significantly increases. Discussion occurs about whether the improvement obtained with a high value for C justifies the important computation cost.

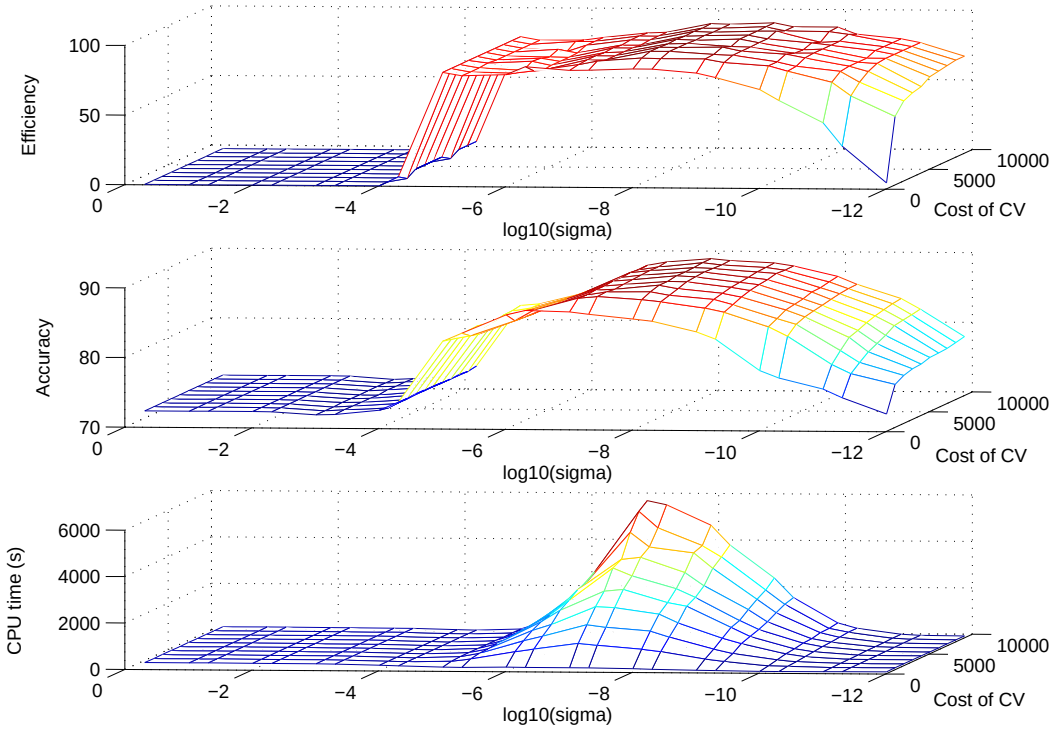

Figure 8: Influence of σ and the cost of constraint violation over performances of a SVM classifier and the corresponding CPU time for 50 runs.

For the experiments exposed in this paper, optimal values for σ and C are respectively set to 10^{-9} and 10^4, and experiments exposed in the next section are obtained with these values.

4.6 Takagi-Sugeno Fuzzy model

Fuzzy models (Zadeh, 1973) combine human-like reasoning style and machine learning properties to approximate non-linear functions. They model the relationships between inputs and outputs with a finite set of rules of the form $R_i : IF\ x\ is\ A_i\ THEN\ y\ is\ B_i\ \ i = 1, \ldots, r$, where r is the number of rules used by the model. A_i and B_i are fuzzy sets defined by membership functions. The shape of the membership functions relies on the designer's choice. Training a fuzzy model consists in finding the fuzzy ensembles A_i and B_i for each rule of the model. Prior to parameter estimation, a fuzzification step is required to transform the numerical inputs of the problem into fuzzy elements to fit the membership functions. An opposite process of defuzzification is required to map the fuzzy output of the model into the desired class label.

In this study, a Takagi-Sugeno model is used (Takagi and Sugeno, 1985). The premises

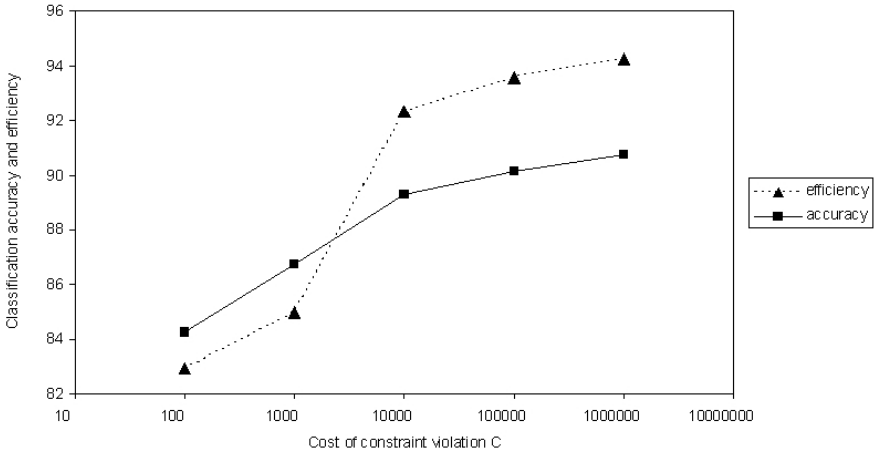

Figure 9: Influence of the cost of constraint violation over performances of a SVM classifier with $\sigma = 10^{-9}$.

are expressed as a conjunction of fuzzy sets, corresponding to the x_i, and the consequents correspond to an affine function of the inputs $R_i : IF\ x_1\ is\ A_{i1}\ AND\ \ldots\ AND\ x_p\ is\ A_{ip}\ THEN\ y_i = \sum_{j=1}^{p} a_{ij}x_j + b_i$, where a_{ij} are coefficients and b_i is a bias. The proposed TS model consists in five rules with Gaussian membership functions of the form $f(x_i) = \exp(\frac{x_i - c_i}{\sigma_i})^2$, where c_i and σ_i are respectively the mean value and the standard deviation of x_i. Since consequents are affine functions, they are defined by the vector of coefficients a_{ij} and the bias b_i. Those parameters are determined during the training step of the model.

Training is performed with the implementation proposed in (Bontempi and Bitattari, 1996). Premises are initialized using k-means clustering and the optimization of c_i and σ_i is based on an alternative of Levenberg-Marquardt algorithm. This implementation offers the possibility to adjust the convergence speed of the learning algorithm to find a compromise between accuracy and computation costs.

5 Comparison of the classifiers

We now compare the results of the six classifiers described in the previous section on a real world database. Each classifier has been built in order to fit as well as possible to this particular problem. Tables 3 and 4 presents the parameters directly set by the authors and the learning algorithms. Table 5 lists the parameters involved in the learning step and table 6 summarizes the parameters determined by selection, using 5 runs of a 10-fold-cross validation (Kohavi, 1995) and the value that maximized the performances of the classifier.

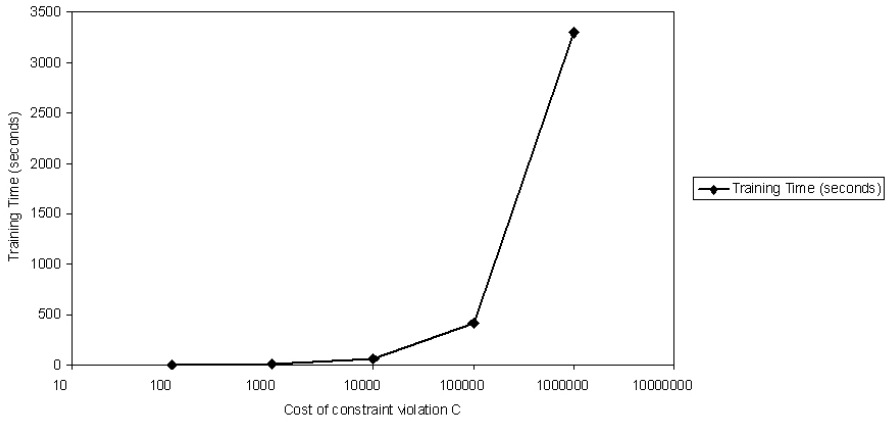

Figure 10: Influence of constraint cost violation over the time required to train the SVM classifier.

5.1 Robustness of classifiers towards the parameters

The analysis of the curves presented in the previous section show that most of the parameters that we considered have a moderate influence on the performances of the classifiers. Concerning multi Layer Perceptrons, the number of hidden units has no significant influence on the performances over three hidden units with one hidden layer and over three hidden units on each layer with two hidden layers (see figures 6 and 7). However, a too large number of hidden units loads down the classifier and increases unnecessarily the learning time. Concerning decision trees, the minimum number of observations needed for a node to be split is not a decisive parameter in the interval of value that we considered (see figure 5). The classifier knn is slightly more sensible to the choice of the number k of neighbors. In the industrial case that we considered, the criterion of efficiency leads to a value of k between 3 and 7, whereas the usual criterion of efficiency leads to a choice with $k > 10$ (see figure 4). For these three types of classifiers, MLP, knn and decision tree, the difference of performances does not exceeds 3% for a large range of values of the parameters.

An exception must be underlined for gaussian kernel SVM, for which a wrong value of the parameter σ can drastically decrease the performances of the classifier (see figure 8). It has to be mentioned that the correct interval of values of this parameter depends on the problem. To a slightest extent, the value of the cost of constraint violation is also important since the efficiency can increase up to 10% in the problem we address (see figure 9). It seems that for a good value of σ, the cost has to be set as high as possible, depending on what is considered as reasonable in term of training time of the SVM (see figures 8 and 10).

Table 3: *Summary of User defined parameter for each classifier.*

Classifier	**User defined Parameters**	**Value**
knn	Distance metric	Euclidian distance
Decision Tree	Pruning	compute the full tree and the optimal sequence of pruned subtrees
	Criterion for choosing a split	Gini's diversity index
Naïve Bayes	Graph Structure	star graph
	Discretization algorithm	hist-ic : optimal histogram based on IC information criterion (Olivier et al., 1994; Leray and François, 2004)
	Decision rule	$class = argmax_C P(C \mid X)$
MLP	Hidden layers	1 or 2
	Output units	4
	Decision rule	$class = argmax_C f_C(x)$ where $f_C(x)$ is the output of unit associated with class C
SVM	Kernel function	Gaussian Kernel function
	Multi-class implementation	*One Versus One model*
TS Fuzzy model	Membership functions	Gaussian
	Number of rules	5

5.2 Performances of the classifiers depending on the size of the learning set

We now look at the influence of the training set size over the performances of the different classifiers. The whole data is divided between training (80% of the samples) and testing (20% of the samples). To estimate the influence of the training set size over the learning process, we use different subsets of the initial training set to build the different classifiers. These subsets contain respectively 20%, 40%, 60%, 80% and 100% of the total training set. Accuracy, efficiency and computation costs are averaged across 50 runs. The performances of the different classifiers are computed with their optimal parameter settings (given in table 6). The variations of accuracy and efficiency versus the number of samples used to train the different classifiers are plot in Figures 11 and 12. The standard deviation is also plot in order to better appreciate the results. Note that since knn classifier does not actually learn a model from data, values given for this particular classifier correspond to the computations needed to retrieve the k nearest neighbours of each test sample in the training set. Computation cost (figure 13) corresponds to the time required for one run for both the learning step and the computation of the class label.

It is important to note that the performances of the classifiers are not the same regarding

Table 4: *Learning algorithm for each classifier.*

Decision Tree	CART (Breiman et al., 1984)
Naïve Bayes	likelihood maximisation
MLP	Levenberg-Marquardt algorithm
SVM	based on Dr. Chih-Jen's LIBSVM algorithm
TS Fuzzy model	an alternative of Levenberg-Marquardt algorithm

Table 5: *Summary of learned parameter.*

Classifier	**Learned Parameters**
Decision Tree	Tree structure
Naïve Bayes	Probability distributions
MLP	Weights w_{ij}
SVM	Support vectors
Fuzzy model	Premises: centers and bases of the membership functions
	Consequents: coeff. a_{ij} and bias b_i of the linear function

whether they are based on accuracy or on efficiency. The accuracy of the six classifiers (Figure 11) is little sensitive to the variation of the size of the learning set: about 1 % of improvement by multiplying by four the size of the learning set. The performances of the six classifiers are very homogenous regarding the accuracy: there is less than 3 % of difference between the classifiers and a small standard deviation, whatever the classifier and the size of the learning set. The results concerning the efficiency are rather different. Beyond 1400 samples in the learning set, the conclusions about efficiency are approximately the same than above: little improvement by multiplying the size of the learning set by two, very similar performances of the classifiers, except for the fuzzy model which is globally less performant. On the other hand, with a smaller learning set, three classifiers turn out to be better regarding the efficiency: decision Tree, knn and SVM, whereas the efficiency of MLP, NB and still more the fuzzy model clearly degrades with a smaller learning set.

These conclusions concerning accuracy and efficiency are directly due to the definition of the efficiency and the unbalanced representation of the classes in the data. The smallest learning set (464 samples) allows a correct computation of the accuracy, but not of the efficiency (at least with the fuzzy classifier, NB and MLP). Indeed, this small learning set includes about 30 items of the class 3, that is too few to provide a correct learning of the critical class.

Regarding the computation cost, nearest neighbour classifiers and SVM are more sensitive to the size of the learning set. Although the time required to train the model is important for SVM, once the model is trained, classification of a new smoke trail can be performed quickly. However, nearest neighbour do not train a usable model from data and the distance metric between the training samples have to be computed each time the system faces a new event to classify. This can be a critical issue for on-line classification of polluting smokes unless prototype editing is applied in order to lower the dimensionality

Table 6: *Summary of parameter determined by cross validation for each classifier.*

Classifier	**CV determined Parameters**	**Value**
knn	Number k of neighbors	9
Decision Tree	Minimal number of observ. for a node to be split	30
MLP	Number of hidden units	4
SVM	Cost of constraint violation C	10^4
	Kernel parameter σ	10^{-9}

of the training set.

5.3 Comparison of the performances of classifiers using data with balanced / unbalanced classes

The previous results are based on data with very unbalanced classes (cf. table 2). In order to discuss the influence of this unevenness, the next results are based on a balanced subset of these data (see table 7). The new data set is constituted of 720 events (on 2900). Since very few events of the critical class 3 have been suppressed, it is interesting to compare the performances of the classifiers in this situation, both in term of accuracy and efficiency. Let's recall that the efficiency criterion has been defined by the industrial in order to focus on critical misclassified events that are wrong alarm and non detection regarding the class 3.

Table 7: *Class distributions in a selected subset of the experimental dataset.*

	Class 0	Class 1	Class 2	Class 3
samples	192	192	143	193
Ratio	27 %	27 %	20 %	27 %

Figures 14 and 15 show the performances of the six classifiers using data with balanced classes. The scale of the y-axis have been chosen in order to facilitate the comparison with figures 11 and 12. The first observation concerns the general loss of performance regarding the accuracy, whereas the efficiency keeps in the same intervalle of value after reducing the learning set from 2884 to 720 by suppressing the "over" cases of classes 0 and 1. The accuracy, that stands in the interval [86, 91] with the original unbalanced data, comes down in the interval [62, 77] using the balanced data. At the opposite, the efficiency of the six classifieurs sets exactly in the same interval [80, 95]. This observation means that the events of level 0 or 1, that do not influence the efficiency, have been more often misclassified than those of classes 2 or 3. This is understandable since a large number of classes 0 and 1 have been suppressed of the learning set to obtain a balanced representation of the classes. This reduction does not affect the quality of the classification of the events of level 3 since the efficiency is not modified whatever the data are balanced or not.

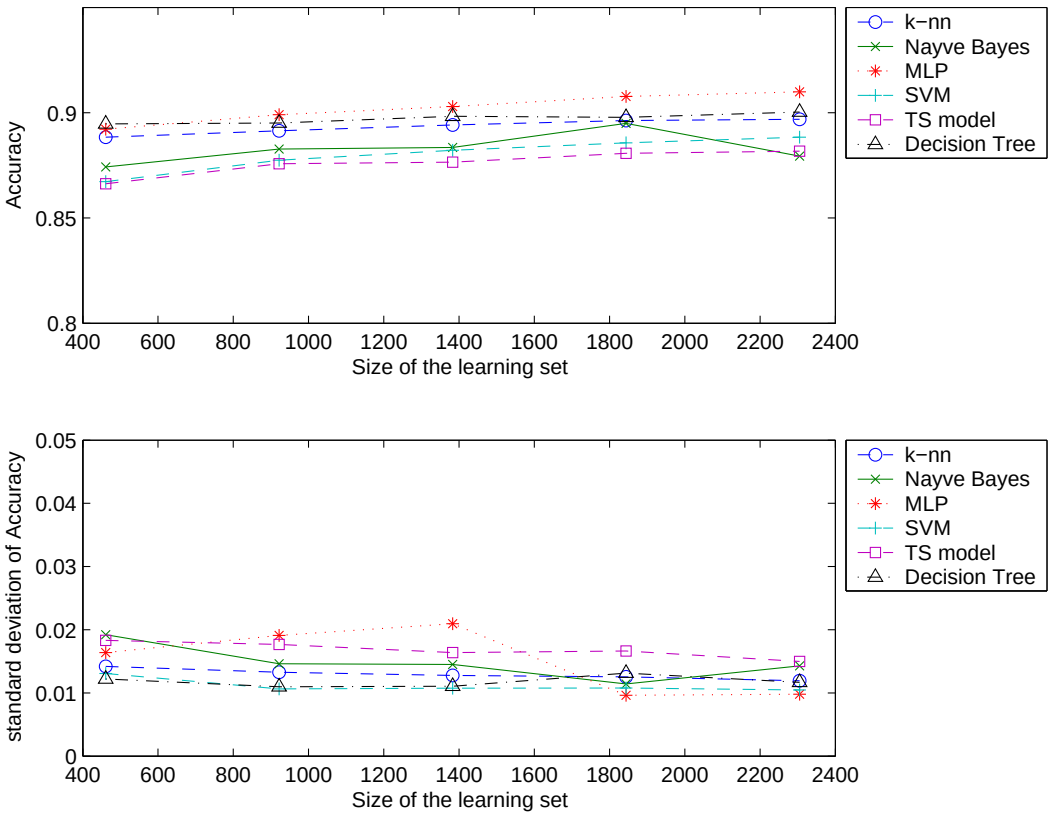

Figure 11: Influence of the size of the learning set over accuracy of six basic classifiers (using data with unbalanced classes).

This conclusion leads to propose a way of managing data with unbalanced classes regarding the context and the objective. When the classes have the same importance, the accuracy is a good way to evaluate the quality of the classifier, since all kind of misclassifications are equally considered. The whole data set is useful for the learning step, regardless its unevenness. On the other hand, when a class (or more) represents critical events on which the study has to focus, and that class is precisely under-represented, it's better using a specific criterion to compute the performance of the classifiers, such as efficiency. In that case, the learning set can be reduced by removing some of the "non critical" events in order to obtain a balanced representation of the classes. This decrease can be interesting for the learning step, without being harmful to the efficiency of the classifiers.

A more refined analyse between classifiers reveals several interesting points in order to select an appropriate classifier. First, the fuzzy classifier is always less performant than the others, whether it be for accuracy or efficiency criterion. Secondly, the MLP classifier withstand less well than the others to the decrease of the size of the learning set (standard deviation of MLP is 2 to 4 times more important for the small size of learning set). The other classifiers (decision Tree, knn, SVM and NB) present very close results and are little

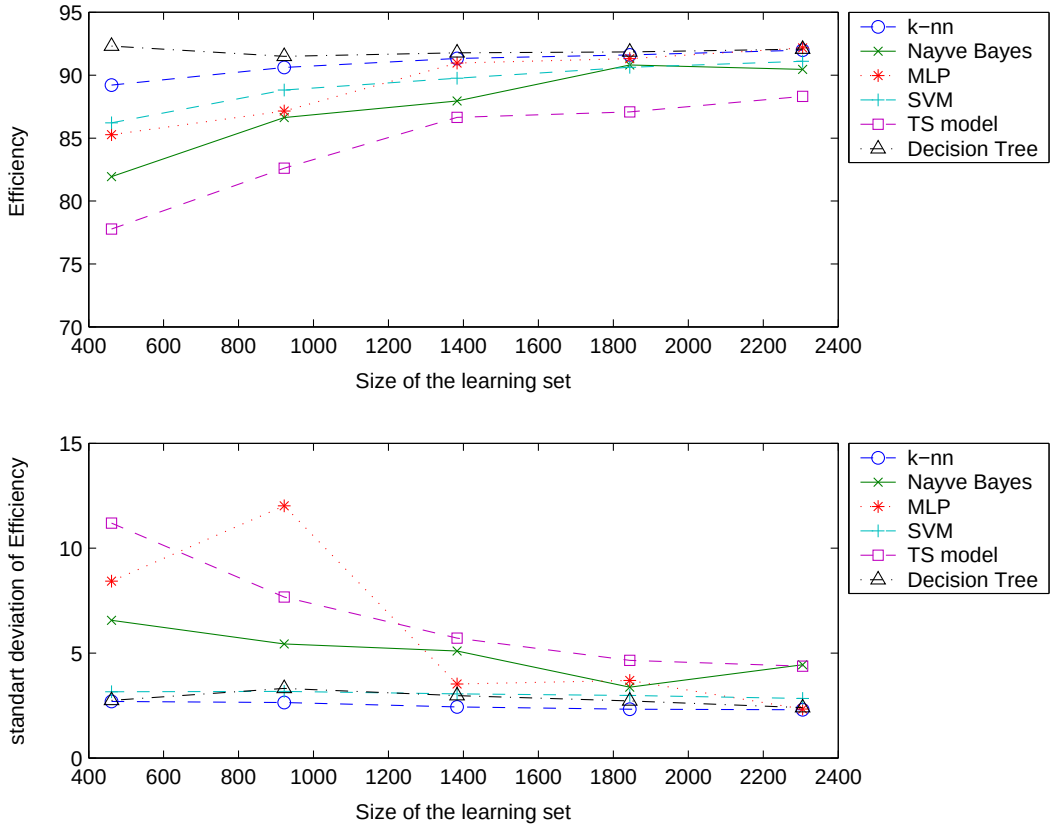

Figure 12: Influence of the size of the learning set over efficiency of six basic classifiers (using data with unbalanced classes).

sensitive to the decrease of the size of the learning setFinally, Bayesian classifier learnt with a smaller data set with balanced classes seems more resistant to the decrease of the size of the learning set than the Bayesian classifier learnt on the whole data.

5.4 Comparison by pairs and with the original expert system

Figure 16 compares two by two the 6 classifiers plus the first expert system designed to classify the degree of severity of the smoke trails. It uses a learning set of 1844 (resp. 460) events and a test set of 576 (resp. 144) events for the data with unbalanced (resp. balanced) classes. Each bar is an average over 50 runs.

The original expert system exhibits an accuracy of 70.9 and an efficiency of 48.0. The performances of the six classifiers tested in this paper clearly outperform the original system of ALOATEC. The Wilcoxon rank-sum test applied to each pair of classifiers returns 0 if the two classifiers are not significantly different. Table 8 shows the results of the Wilcoxon rank-sum test applied on the data used to plot the figure 16, with a level of significance of 0.05. These tests allow to maintain that all the classifiers, except the TS fuzzy model,

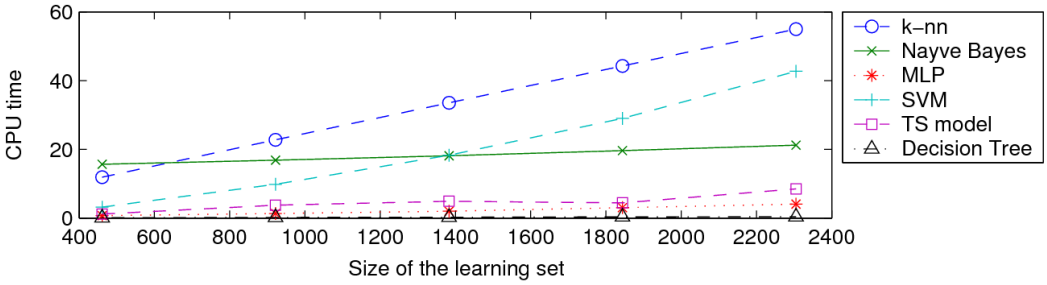

Figure 13: Influence of the size of the learning set over computation times.

are equivalent when compared on the industrial criteria of efficiency. The fuzzy classifier is systematically less performant than the others, whatever the criteria, accuracy or efficiency. This may be partially explained because the library used for the fuzzy model are not optimized. The other elements of these tests are inconclusive: most of the results are not constant depending on whether the data are balanced or not. Thus, it's not possible to establish a total order ranking on the six classifiers and we can not say what is the "better" one.

In this industrial context, the critical events (those of degree 3) absolutely have to be classified without mistake because of the potentially high cost and severe consequences of each of such misclassification. At the opposite, misclassifications of events of classes 0, 1 or 2 are of much less serious. The efficiency measure has been defined by the industrial to manage this specific problem. The six classifiers studied here provide a very good efficiency compared to the initial expert system (about two time better).

5.5 Synthesis of the comparisons and partial conclusion

Regarding the industrial criteria of efficiency, any of the five classifiers among K nearest neighbour, naïve Bayesian network, artificial neural network, support vector machine or decision tree present equivalent performances. Therefore, the choice of a particular machine learning algorithm for the application relies on different questions such as the difficulty to correctly configure the classifier prior to training, the number of training samples available, the time of classification and the evaluation function. The table 9 sums up the advantages of each classifier regarding these criteria, in order to determine the classifier which best fits the application of pollution detection. Decision trees are efficient even for small training sets. Moreover, they do not involve numerous parameters compared to neural networks for instance, for which the structure of the network and the activation functions of each node have to be set prior to model training.

Regarding the robustness of classifiers towards the parameters in our study, Decision tree, knn and MLP are the best choices since their performance are not very sensitive to the variation of their parameters. On the contrary, the SVM parameters may be difficult to set, leading to a decrease of performances. Regarding the robustness of classifiers towards

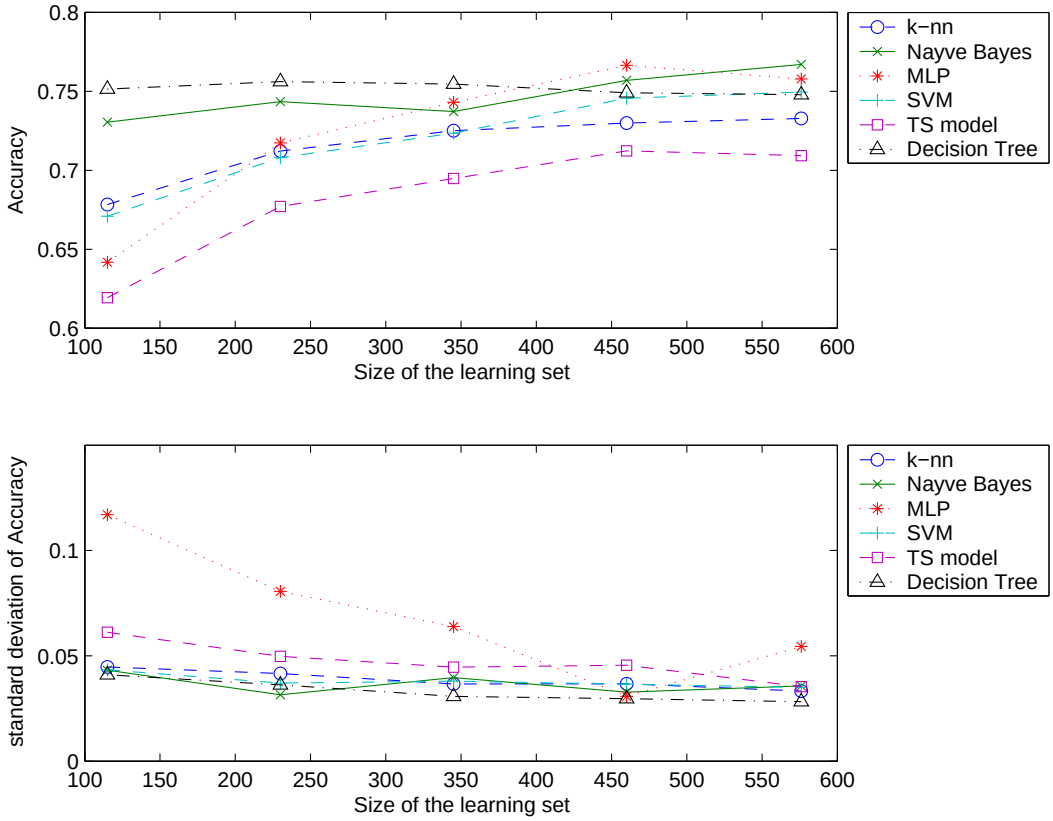

Figure 14: Influence of the size of the learning set over accuracy of six basic classifiers using data with balanced classes.

the size of the learning set, the best classifiers are decision tree and SVM, and to a lesser extent the naïve Bayes classifierRegarding knn, the number of examples used to compute the distance does not influence the performances of the classifier. However, this is time consuming, and may constitue an obstacle to the use in real-time. The other classifiers can be used in real-time, even if they can need a longer time for the learning step.

6 Conclusion

This article presents a comparative study of several algorithms adapted to supervised classification of camera-detected polluting smokes. The main objectif is to provide a monitoring tool that allow to focus on critical misclassified events, such as wrong alarm and non detection regarding the events of highest severity (class 3). In that aim, the industrial has defined the efficiency that is a specific criterion, that is used to evaluate the classifiers, in addition with the usual criteria of accuracy. Various algorithms of classification are used and the benefits offered by each one are compared. All the machine learning algorithms de-

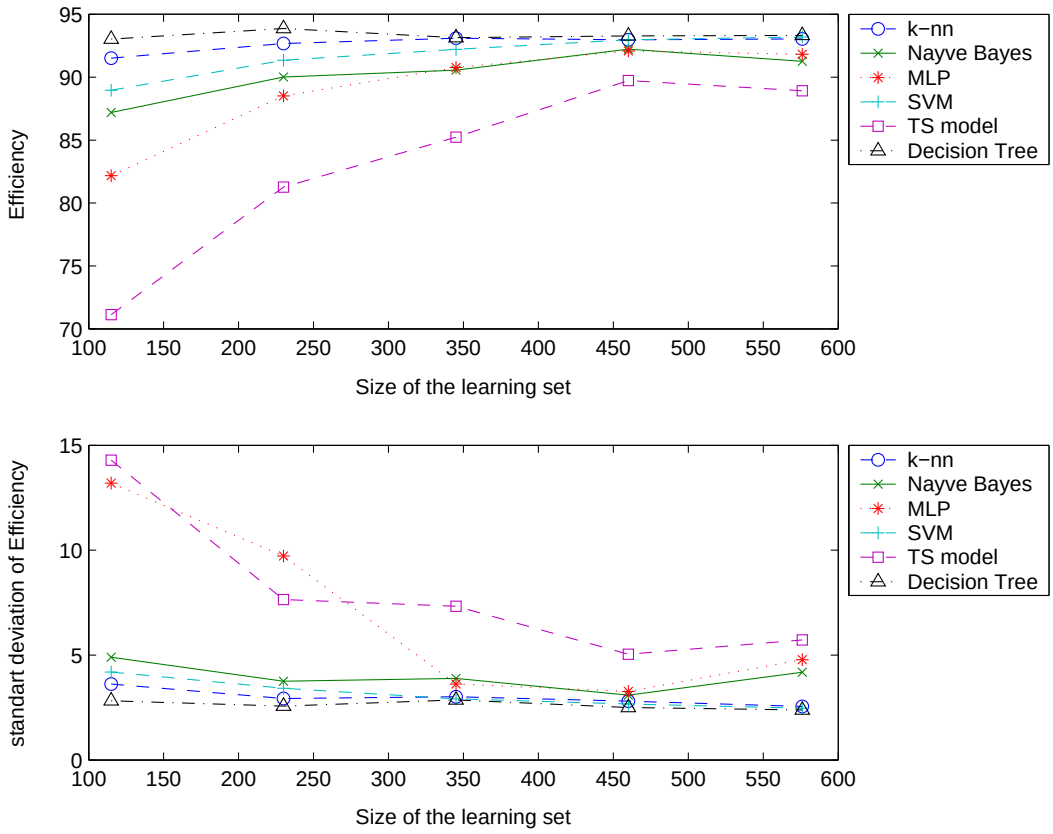

Figure 15: Influence of the size of the learning set over efficiency of six basic classifiers using data with balanced classes.

scribed in this study outperform the original classification device of the DETECT system, both in terms of efficiency and correct classification of the different visual scenes.

The monitoring tool is supposed to be used by human operators who are not necessarily familiar with machine learning ; therefore the complexity of the chosen algorithm can be a critical issue in our case. The supervised learning allows the design of classification procedures automatically from the data, with good overall results. This considerably improves usability of the system and helps reducing the complexity of human intervention to configure the system. This is a very important aspect of our work since one of the objectives is to design a system to be used by operators who do not possess prior knowledge about machine learning.

In this paper we have supposed that the features used for the classification and their numbers are set by the experts of the domain of the study. But it should be interesting in future work to study these aspects.

Regarding future research work, we are also investigating the interest of considering time series to take into account the dynamic aspect of each event to perform classification. This dynamic aspect could be considered as an additional feature for the smoke trail and

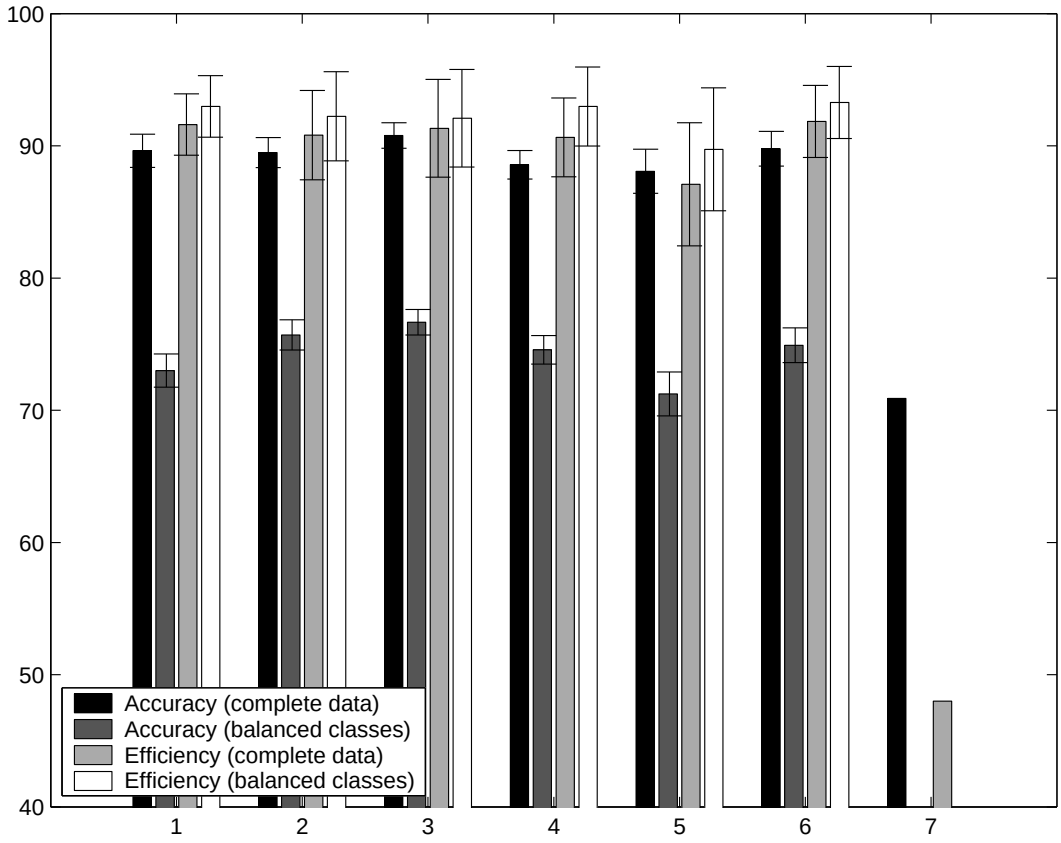

Figure 16: Summary of the performances (average and standard deviation) of the original expert system and the six studied classifiers (1: knn, 2: naïve Bayes classifier, 3: MLP, 4: SVM, 5: Fuzzy classifier, 6: decision Tree, 7: expert system)

could help reducing the number of incorrectly classified event. Another trail concerns the design of multiple classifier systems, such as Bagging (Breiman 1996) or Boosting (Freund et Shapire 1996) and others algorithms (Kuncheva, 2004; Gacquer et al., 2008). These methods combine the decision of several classifiers in order to improve the final performances. The basic classifiers can be either of different types or more commonly all of the same type (such as decision trees). In that case, each classifier can be learnt on a re sampled subset of the learning data set such as in the Bagging method. In the industrial context of atmospheric pollution detection by camera, it would be interesting to evaluate the interest of multiple classifiers by considering their robustness regarding the size of the learning set, the unbalanced data and the computing time. In the one hand one can expect a gain in term of efficiency and accuracy, but in the other hand the model is of higher complexity.

Table 8: *Wilcoxon rank-sum test on pairs of classifiers.*

Balanced Data

Accuracy						Efficiency				
NB	MLP	SVM	TS	Tree		NB	MLP	SVM	TS	Tree
1	1	1	1	1	Knn	0	0	0	1	0
	0	0	1	0	NB		0	0	1	0
		1	1	1	MLP			0	1	1
			1	0	SVM				1	0
				1	Fuzzy					1

Complete Data

Accuracy						Efficiency				
NB	MLP	SVM	TS	Tree		NB	MLP	SVM	TS	Tree
0	1	1	1	0	Knn	0	0	0	1	0
	1	1	1	0	NB		0	0	1	0
		1	1	1	MLP			0	1	0
			0	1	SVM				1	1
				1	Fuzzy					1

Table 9: *Sum up of the advantages of each classifiers according to our study.*

	Robustness to the parameter setting	Robustness to the size of the learning set	Use in real time
Decision tree	+	+	+
knn	+	no learning step	-
Naïve Bayes		±	+
MLP	+	-	+
SVM	-	+	
TS model		-	+

Aknowledgments

The present research work has been supported by the French Agency for Environment and Energy Mastery (ADEME), the International Campus on Safety and Intermodality in Transportation the Nord-Pas-de-Calais Region, the European Community, the Regional Delegation for Research and Technology, the Ministry of Higher Education and Research, and the National Center for Scientific Research. The authors gratefully acknowledge the support of these institutions. The authors would also like to thank the ALOATEC Com-

pany and its director, Philippe Bourrier, for their support concerning the research project exposed in this article, and for having provided the data required for the experiments.

References

Aha, W., Kibler, D., Albert, M. K., 1991. Instance-based learning algorithms. Machine Learning 6 (1), 37–66.

Allwein, E. L., Schapire, R. E., Singer, Y., 2001. Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. 1, 113–141.

Bakos, G., Tsagas, N., 1995. A laser system for pollution detection. Journal of The Franklin Institute 332, 211–218(8).

Bontempi, G., Bitattari, M., 1996. Matlab Software Tool for Neuro-Fuzzy Identification and Data Analysis. ULB, Brussels, Belgium.

Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A., January 1984. Classification and Regression Trees. Chapman & Hall/CRC.

Chang, C.-C., Hsu, C.-W., Lin, C.-J., July 2000. The analysis of decomposition methods for support vector machines. IEEE Trans. Neural Networks 11 (4), 1003–1008.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20 (Issue 3), 273–297.

Dasarathy, B. V., 1990. Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press, Los Alamitos, California.

Dietterich, T. G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923.

Dietterich, T. G., Bakiri, G., 1995. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286.

Duda, R. O., Hart, P. E., Stork, D. G., November 2001. Pattern Classification (2nd Edition). Wiley-Interscience.

Fenger, J., 2009. Air pollution in the last 50 years - from local to global. Atmospheric Environment 43 (1), 13–22.

Frias-Martinez, E., Sanchez, A., Velez, J., 2006. Support vector machines versus multilayer perceptrons for efficient off-line signature recognition. Engineering Applications of Artificial Intelligence 19, 693–704.

Gacquer, D., Delmotte, F., Delcroix, V., Piechowiak, S., 2006. Comparison of bayesian classifiers to detect pollution. In: Proc. 25th Edition of European Annual Conf. on Human Decision-Making and Manual Control. Valenciennes, France, http://www.univ-valenciennes.fr/congres/EAM06/papers_by_author.html.

Gacquer, D., Delmotte, F., Delcroix, V., Piechowiak, S., June 2008. A genetic approach for training diverse classifier ensembles. In: Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008). Malaga, Spain, pp. 798–805.

Hagan, M. T., Menhaj, M. B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE Trans. on Neural Networks 5 (6), 989–993.

Jensen, F. V., July 2001. Bayesian Networks and Decision Graphs. Information Science and Statistics. Springer.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. Morgan Kaufmann, pp. 1137–1145.

Kuncheva, L. I., 2004. Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience.

Leray, P., François, O., 2004. BNT structure learning package: documentation and experiments. Technical Report FRE CNRS 2645, Laboratoire PSI - INSA Rouen, Rouen, France.

Muñoz Expósito, J. E., García-Galán, S., Ruiz-Reyes, N., Vera-Candeas, P., 2007. Adaptive network-based fuzzy inference system vs. other classification algorithms for warped lpc-based speech/music discrimination. Engineering Applications of Artificial Intelligence 20 (6), 783–793.

Nguyen, D., Widrow, B., June 1990. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptative weights. In: Proc. of the International Joint Conference on Neural Networks. Vol. 3. San Diego, Calif., pp. 21–26.

Olivier, C., Colot, O., Courtellemont, P., El Matouat, A., 1994. Information criteria and abrupt changes of probability laws. Signal Processing VII: Theory and Applications 7 (3), 1855–1858.

Pearl, J., March 2000. Causality : Models, Reasoning, and Inference. Cambridge University Press.

Quinlan, J. R., 1986. Induction of decision trees. Machine Learning 1 (1), 81–106.

Quinlan, R. J., January 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

Ripley, B. D., January 1996. Pattern Recognition and Neural Networks. Cambridge University Press.

Sarle, W., 1995. Stopped training and other remedies for overfitting. In: Proceedings of the 27th Symposium on Interface. pp. 352–360.

Shakhnarovich, G., Darrell, T., Indyk, P., 2006. Nearest-Neighbor Methods in Learning and Vision: Theory and Practice (Neural Information Processing). The MIT Press.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 15 (1), 116–132.

Tetko, I. V., Livingstone, D. J., Luik, A. I., 1995. Neural network studies. 1. comparison of overfitting and overtraining. Journal of Chemical Information and Computer Science 35 (5), 826–833.

Vapnik, V., 1998. Statistical Learning Theory. Wiley-Interscience.

Zadeh, L. A., 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Systems, Man, and Cybernetics SMC-3, 28–44.

Zolghadri, A., Monsion, M., Henry, D., Marchionini, C., Petrique, O., 2004. Development of an operational model-based warning system for tropospheric ozone concentrations in bordeaux, france. Environmental Modelling and Software 19 (4), 1003–1008.